# System 80

## System Description

H

SPERRY UNIVAC

This document contains the latest information available at the time of preparation. Therefore, it may contain descriptions of functions not implemented at manual distribution time. To ensure that you have the latest information regarding levels of implementation and functional availability, please consult the appropriate release documentation or contact your local Sperry Univac representative.

Sperry Univac reserves the right to modify or revise the content of this document. No contractual obligation by Sperry Univac regarding level, scope, or timing of functional implementation is either expressed or implied in this document. It is further understood that in consideration of the receipt or purchase of this document, the recipient or purchaser agrees not to reproduce or copy it by any means whatsoever, nor to permit such action by others, for any purpose without prior written permission from Sperry Univac.

Sperry Univac is a division of the Sperry Corporation.

This document was prepared by Systems Publications using the SPERRY UNIVAC UTS 400 Text Editor. It was printed and distributed by the Customer Information Distribution Center (CIDC), 555 Henderson Rd., King of Prussia, Pa., 19406.

PRINTED IN U.S.A.

# Contents

1. INTRODUCTION 1

2. ADVANCED CAPABILITIES OF SYSTEM 80 5

2.1. SIGNIFICANT FEATURES 5

2.2. INTERACTIVE SERVICES 5
2.2.1. Workstation 6
2.2.2. Editors 7
2.2.3. Screen Format Services 9
2.2.4. Dialog Processing Services 11
2.2.5. System Dialogs 12
2.2.5.1. Job Control Dialog 12
2.2.5.2. System Generation Dialog 13
2.2.5.3. Data Utilities Dialog 13

2.3. ENHANCED PROGRAMMING LANGUAGES 14

2.4. INFORMATION MANAGEMENT SYSTEM 15

2.5. DATA BASE MANAGEMENT SYSTEM 16

2.6. JOB PROCESSING 17
2.6.1. Batch Job Processing 18
2.6.2. Interactive Job Processing 19

2.7. MULTIJOBBING 20

2.8. DYNAMIC RESOURCE MANAGEMENT 20

2.9. DATA COMMUNICATIONS 20

2.10. DISTRIBUTED DATA PROCESSING 22

2.11. APPLICATIONS PROGRAMS 23

2.12. CONVERSION AIDS 23

2.13. AVAILABILITY, RELIABILITY, AND MAINTAINABILITY 24

3. SYSTEM SOFTWARE 25

3.1. SOFTWARE DESIGN CONCEPTS AND CAPABILITIES 25

3.2. SYSTEM CONTROL SOFTWARE 27
3.2.1. Supervisor 27
3.2.1.1. Supervisor Interrupt Requests 28
3.2.1.2. Task Switcher 28
3.2.1.3. Physical Input/Output Control System 29
3.2.1.4. Transient Management 30
3.2.1.5. Timer Service Management 31
3.2.1.6. System Console Management 31
3.2.1.7. Workstation Management 31
3.2.1.8. Error Recovery 31
3.2.1.9. Error Logging 31
3.2.1.10. Multitasking 32
3.2.1.11. Diagnostic Debugging Aids 32
3.2.1.12. Main Storage Management 34
3.2.2. Job Control 35
3.2.2.1. Job Coordination 36
3.2.2.2. Device Assignment 37
3.2.2.3. Job Step Operation 37
3.2.2.4. Regulation of Job Environment 38
3.2.2.5. Dialog for Job Control Stream and Jproc Preparation 39
3.2.2.6. File Cataloging 39
3.2.3. Interactive Processors 40
3.2.3.1. Workstation Command Processors 40
3.2.3.2. Dialog Processor 40
3.2.3.3. Screen Format Coordinator 40
3.2.4. Consolidated Data Management 41
3.2.4.1. Logical Input/Output Control System 42
3.2.4.2. Disk Access Method 42
3.2.4.3. Diskette Access Method 42
3.2.4.4. Workstation Access Method 42
3.2.4.5. Magnetic Tape and Unit Record Device Access Methods 43
3.2.4.6. Device Independence 43
3.2.5. System Service Programs 43
3.2.5.1. System Librarians 43
3.2.5.2. Linkage Editor 44
3.2.5.3. Disk, Diskette, and Tape Initialization Routines 46
3.2.5.4. Dump Routines 46
3.2.5.5. Software Maintenance Packages and Changes 47
3.2.5.6. Catalog Manipulation Utility 48
3.2.5.7. Disk Dump/Restore 48
3.2.5.8. System Patch Routine 48
3.2.6. System Installation Facilities 49
3.2.6.1. Software Installation Facilities 49
3.2.6.2. System Generation Facilities 49
3.2.6.3. Installation Verification Programs 50

3.3. EXTENDED SYSTEM SOFTWARE 50
3.3.1. Screen Format Generator 50
3.3.2. Dialog Specification Language Translator 51
3.3.3. Data Utilities 51
3.3.4. SORT/MERGE 52
3.3.5. System/3 Compatible Sort 53
3.3.6. Spooling and Job Accounting/Reporting 53

3.4. LANGUAGE PROCESSORS 55
3.4.1. ESCORT 55
3.4.2. RPG II 56
3.4.3. COBOL 57
3.4.4. BASIC 59
3.4.5. FORTRAN IV 59
3.4.6. Basic Assembly Language (BAL) 59

3.5. THE GENERAL EDITOR 60

3.6. INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM) 60
3.6.1. NTR System Utility 66

3.7. INFORMATION MANAGEMENT SYSTEM 66

3.8. DATA BASE MANAGEMENT SYSTEM 67

3.9. DISTRIBUTED DATA PROCESSING 69
3.9.1. Job Distribution Facility 69
3.9.2. File Transfer Facility 69

3.10. UTS 400 SUPPORT 70
3.10.1. UTS 400 COBOL 71
3.10.2. UTS 400 Edit Processor 71
3.10.3. UTS 400 Load/Dump Facility 71

3.11. APPLICATIONS PROGRAMS 71
3.11.1. UNIVAC Industrial System 80 72
3.11.2. UNIVAC Industrial System 80-Extended 75
3.11.3. Accounting Control System 80 76
3.11.4. Information Collection System 80 77
3.11.5. UNIVAC Distribution Information System - Wholesale 77
3.11.6. Order Entry 80 78
3.11.7. Order Entry 80-Extended 79

3.12. CONVERSION AIDS 80
3.12.1. IBM System/3 80
3.12.1.1. Disk Data File Conversion 81
3.12.1.2. Model 10, 12, and 15 Source and Proc Transcriber 81
3.12.1.3. OCL to DCL Converter (JCLCON801) 81
3.12.2. IBM System 32/34 81
3.12.3. SPERRY UNIVAC 9200/9300 System 81
3.12.4. SPERRY UNIVAC Operating System/4 (OS/4) 82
3.12.5. Honeywell 100 Series 83
3.12.5.1. COBOL Translator 83
3.12.5.2. Data File Transcriber 84
3.12.6. Honeywell 200/2000 Series 84
3.12.6.1. COBOL Translator 84
3.12.6.2. EASYCODER Converter 84
3.12.6.3. Data File Transcriber 84
3.12.7. Honeywell Series 60, Level 62 and Level 64 84
3.12.7.1. Program Library and Data File Transcriber 84
3.12.7.2. COBOL Translator 85

3.13. MAINTENANCE AND DIAGNOSTIC SOFTWARE 85

3.14. SYSTEM ACTIVITY MONITOR 86

**4. HARDWARE** 87

4.1. SYSTEM 80 MODELS 87

4.2. HARDWARE CONFIGURATIONS 87

4.2.1. Basic System 87

4.2.2. Expanded System 87

4.3. CENTRAL PROCESSORS 90

4.3.1. Control Processor 90

4.3.1.1. Basic Instruction Set 90

4.3.1.2. Nonprivileged Instruction Set 91

4.3.1.3. Privileged Instruction Set 91

4.3.2. Main Storage Processor 91

4.3.2.1. Low-Order Main Storage 92

4.3.2.2. Storage Protection 92

4.4. SPERRY UNIVAC PERIPHERAL SUBSYSTEMS 92

4.4.1. Disk Subsystems 92

4.4.1.1. Nonremovable Disk Drive 93

4.4.1.2. Removable Disk Drive 94

4.4.2. Workstation Subsystems 95

4.4.2.1. System 80 Workstation 95

4.4.2.2. Console Workstation 99

4.4.3. Diskette Subsystem 99

4.4.4. Printer Subsystems 102

4.4.4.1. High Performance Line Printer 103

4.4.4.2. Line Printer 103

4.4.5. Card Processor Subsystems 105

4.4.5.1. Card Reader 105

4.4.5.2. Card Punch 105

4.4.6. UNISERVO 10 Magnetic Tape Subsystem 107

4.4.7. Communications Facilities 109

4.4.7.1. UNISCOPE Display Terminals 110

4.4.7.2. UTS 4000 Universal Terminal System 110

4.4.7.3. UTS 400 Universal Terminal System 112

4.4.7.4. BC/7 Business Computer System 112

4.4.7.5. Additional Communications Devices 113

4.5. HARDWARE SUMMARY 113

## FIGURES

1–1. System 80 Installation 1

## TABLES

| | | |
|---|---|---|
| 4–1. | Basic and Expanded System Capacities | 89 |
| 4–2. | Disk Subsystem Characteristics | 94 |
| 4–3. | Workstation Keys | 98 |
| 4–4. | Workstation Characteristics | 98 |
| 4–5. | Diskette Subsystem Characteristics | 101 |
| 4–6. | Printer Subsystem Characteristics | 104 |
| 4–7. | Card Reader Characteristics | 106 |
| 4–8. | Card Punch Characteristics | 106 |
| 4–9. | UNISERVO 10 Characteristics | 108 |
| 4–10. | Communications Functionality | 109 |
| 4–11. | System Characteristics | 113 |

# 1. Introduction

Today, computer performance is being challenged by a variety of applications that have increased in scope and complexity. As a result, a significant number of established data processing systems cannot meet the requirements of these new applications because they lack the inherent processing capability. To meet this need, Sperry Univac has developed System 80, an interactive data processing system for the established users who want to upgrade their data processing capabilities to include interactive capabilities and transaction processing. It is designed for businesses that require a computer resource offering high performance at low cost and extended capabilities that encompass an unmatched range of computational requirements.

System 80 features high-quality, compact, integrated hardware, complemented by an advanced, yet easy-to-use interactive software package: the SPERRY UNIVAC Operating System/3 (OS/3). Together, they extend the computer resource into a maximum usage environment. Figure 1-1 depicts a System 80 installation showing the central processing complex with an integrated nonremovable disk, the system console/workstation, and a variety of devices, including manual and autoload diskettes, a high-speed printer, and an external nonremovable disk unit.

**Interactivity**

**Transaction Processing**

**Multijobbing**

**Advanced Hardware Technology**

**Communications**

**Distributed Data Processing**

**Data Base Management**

**Conversion**

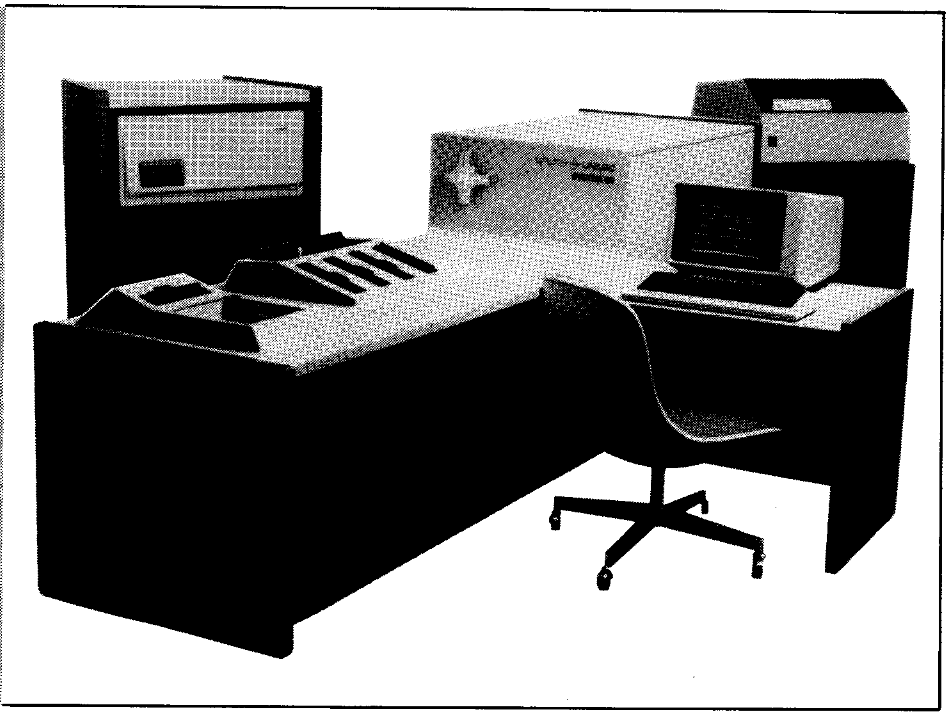


*Figure 1—1. System 80 Installation*

Here are some of the advanced capabilities of System 80 that increase its productivity:

- **Interactivity**

  System 80 offers a number of interactive programming features, all initiated and controlled through the workstation. These features make the system easy to use, make it accessible to the novice or nonprogramming individual, and provide a number of programming tools that simplify the programming effort. A number of users can access the system simultaneously to perform interactive functions. These functions range from interactively generating source programs and job control streams to performing such functions as creating data files and transferring files between storage devices. Facilities are included to allow users to interactively enter variable data into jobs running in the system. In addition, the user can create screen format displays and interactive dialogs to simplify the inputting of data through the workstation and the displaying of output information.

  The interactive services allow for decentralization of system operation, job preparation and initiation through the workstation, interactive program development and execution, and the initiation of system utilities. They assist the novice user in using the system, and they increase the efficiency of experienced programmers.

- **Transaction Processing**

  Sperry Univac places information at the fingertips of the user through the SPERRY UNIVAC Information Management System (IMS). Employing a network of interactive workstations and data communications devices, personnel can instantly access and update user data files with no more knowledge than is needed to operate a keyboard. IMS offers a reliable, easy-to-use cmmunications/data management package that adds a new dimension to computer applications through an inquiry/response capability in a real-time environment. Despite ease of use in the IMS system, security is paramount: all files are protected from unauthorized access, and extensive integrated recovery facilities are available.

  User support for IMS is in the form of applications programs called UNIQUE, which are supplied by Sperry Univac and require no programming effort on the part of the user. In addition, customized application programs can be developed by using COBOL, RPG II, or the basic assembly language (BAL).

- **Multijobbing**

  Multijobbing increases throughput: the actual productivity of a data processing system as measured by the amount of useful computing work done per minute or hour. Multijobbing increases system productivity by interleaving the execution of job steps from more than one job. When any job step is waiting for an external event (I/O request) to occur before processing can continue, another job is given control. Priorities and available resources determine which job step is given control by the data processing system at any point in time. Up to 14 jobs can be processed concurrently by System 80.

- **Advanced Hardware Technology**

  System 80 incorporates a number of advanced hardware features that increase system efficiency and enhance user productivity. Among these features are:

  - Large-scale integrated circuits that increase system speed and reliability and thus increase the processing power of the system.

  - One megabyte direct access diskettes that are well suited as storage media for library and data files.

- An integrated, nonremovable disk subsystem offering a more reliable and faster mass storage device for storing the operating system and critical user files. Additional high-density removable and nonremovable disk devices can be configured to offer a total disk storage capacity in excess of 900 megabytes. Plus, a fixed-head option for nonremovable disk devices is available, offering a significant increase in access speeds and additional storage capacity.

- Autoload diskette drives, allowing the system to cycle up to 20 diskettes without user intervention.

- A high-performance control storage option, which can increase processor speed over 50 percent.

- An input/output microprocessor (IOMP) can be configured to increase the number of peripheral I/O devices attached to the system.

- High-speed printers offering flexibility in character sets and line length selections, coupled with ease of operation, provide a maximum system printing capacity of over 7500 lines per minute.

- A variety of card processing subsystems (read only, read/punch, and punch only) is available, offering maximum choice to the user selecting a punched card configuration.

- High-density, extremely reliable tape drives make the SPERRY UNIVAC tape system a viable mass storage device for all users.

■ **Communications**

Data communications use commercially available communications facilities to link a central processing site with remote sites to accomplish a variety of data processing applications. In meeting today's rising demand for this type of service, the System 80 data processing system includes a sophisticated, yet easy-to-implement communications package – the integrated communications access method (ICAM) terminal support facility. All users, whether they are investigating communications for the first time or are upgrading their present capabilities, will find ICAM an indispensable tool. It is an integrated system capable of supporting several levels of communications processing. ICAM can be tailored to fit the user's needs, the type of service desired, and the installation configuration.

■ **Distributed Data Processing**

System 80 can be easily incorporated into a distributed data processing network. Software is supplied that provides the interface required to match the systems to one another so that each system can be used to initiate a job at another location. Additionally, data files can be transferred between systems. The establishment of a distributed data network allows all systems included in the network to share the processing load of the entire organization.

■ **Data Base Management**

The SPERRY UNIVAC data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs provide for the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

- **Conversion**

  In the purchase of any new system, the conversion of existing programs and data files is always uppermost in the customer's mind. Sperry Univac provides utility routines to convert data files generated for the SPERRY UNIVAC 9200/9300 and 9400/9480 systems, the IBM System/3 and System 32/34, and Honeywell Series 100, Series 200/2000, and Series 60, Level 62 and 64 into data files that are suitable for use on the System 80 data processing system. Also, information is provided on how to convert your present programs to System 80 formats.

The System 80 interactive data processing system is an advanced and powerful computing system – a system that is both easy to use and specifically designed to meet the needs of today's business world. The system combines the latest in hardware technology with the proven OS/3 operating system, now enhanced by the inclusion of integrated, interactive software. The system offers a variety of programming languages, communications and data base facilities, interactive communications with the system, and interactive programming aids. The interactive features, coupled with the simplified operating system, make this system easy to use, simplifying information entry and retrieval to such an extent that even inexperienced and noncomputing professionals can access the system and achieve useful results.

Although System 80 sets a new standard in ease-of-use computing, it provides a powerful, versatile computing environment to meet even the most complex programming needs. The system incorporates the latest developments in both hardware and software technology and offers a number of advanced design features that make it unique among today's data processing systems.

# 2. Advanced Capabilities of System 80

## 2.1. SIGNIFICANT FEATURES

This section highlights the features of System 80 that make it unique as a total business data processing system: a powerful and flexible system that meets the needs of all users. Among these features are:

- The full range of interactive services
- Enhanced programming languages
- Communications and data management facilities, including distributed data processing
- Application packages for users with specific software requirements
- Conversion aids for those migrating from other systems

## 2.2. INTERACTIVE SERVICES

To meet the growing need to obtain immediate information and results from a data processing system, System 80 offers interactive services. These services:

- Provide greater system usage and productivity through faster response times and enhanced data entry and retrieval methods
- Allow every member of an organization to access the system through easy-to-use facilities while maintaining a high degree of security through protective passwords and user identification features
- Simplify the entire programming effort by providing interactive communications to the system control software and the use of dialog and screen formats for control information entry

The interactive services include:

- Workstations
- General and language editors
- Screen format services
- Dialog processing services
- System dialogs

### 2.2.1. Workstation

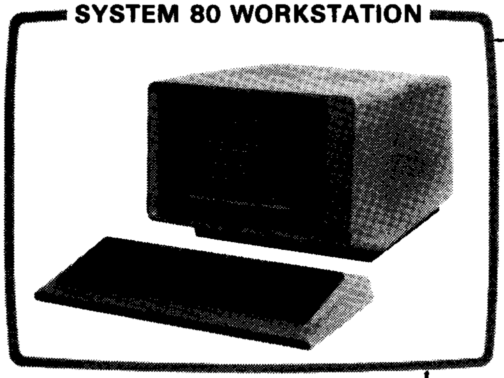
In the interactive OS/3 programming environment, the workstation is the user's primary means of communicating with the system. A set of interactive commands can be issued directly from the workstation to:

- direct the operation of the system or a system component;
- query the system for specific information;
- direct the execution of a job; and
- interactively create and modify source programs, data files, job control streams, etc.

The workstation can also be used as an input/output device dedicated to a job or system component.

The workstation consists of a keyboard used for inputting entries and a video screen used to display the input entries and system responses.

SYSTEM 80 WORKSTATION



VIDEO SCREEN

Displays:

- Input entries
- System responses

KEYBOARD

Inputs entries through:

- Standard alphabetic and numeric keys
- Cursor control
- Function keys

The cursor control pad is used to control the positioning of the screen cursor. The function keys, when pressed, cause the system to perform a specific function. Some keys perform the same function at all times, some functions vary according to the system component that the workstation is in communication with, and others can be programmed by the user to perform a particular function.

To initiate workstation activities, the user must enter a LOGON message with accompanying user identifier and password. Once logged on, the user can begin any activity he chooses. The type of activities performed fall into two categories: control activities and data input/output activities. The user has commands available to control the initiation and execution of jobs and to initiate the various interactive services, such as the job control dialog or the editor.

The workstation is also used for input and output to system and user programs. It can be assigned to an IMS network to initiate and respond to IMS transactions. Or it can be assigned to a user job, in which case it is used as an input/output device for the job.

The workstation can also be used as a program development tool using the various interactive program development facilities, such as Editor, RPG II Editor, ESCORT, and BASIC.

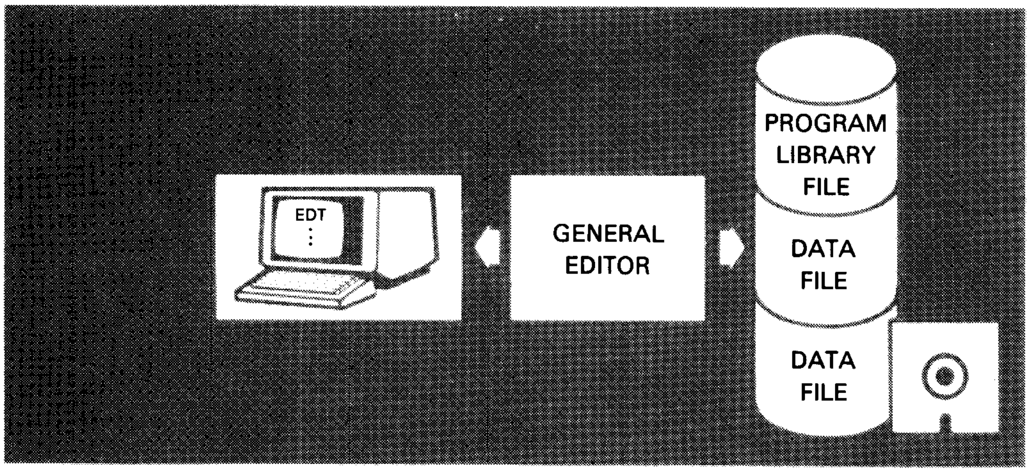
## 2.2.2. Editors

The OS/3 editors are easy-to-use, interactive, integrated components of OS/3 that help the user create and edit job control streams, data, and program files. The general editor features a comprehensive command set for creating and maintaining computer-based files on disks or diskettes. The language editors are specialized subsystems of the general editor and are tailored to handle the unique problems of creating and updating source language programs. The language editors can be called at any time during a general editing session.

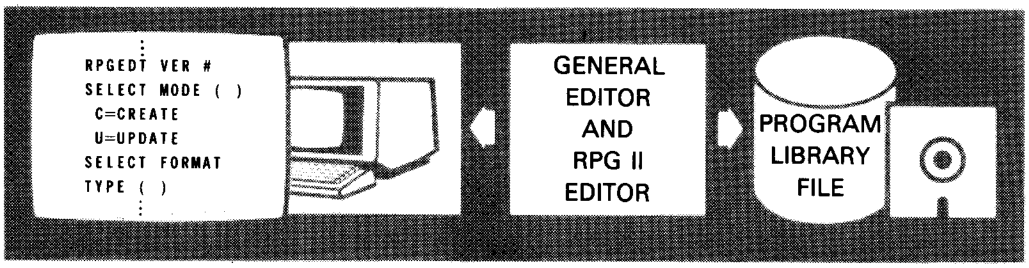
Because they are an integral part of OS/3, the editors do not occupy a job slot when they are being used. Thus, any number of editing sessions may be in progress at the same time that jobs (up to 14) are being processed by the system.

The general editor and the language editors are activated with a simple command from the workstation.

**GENERAL EDITOR**
**IS USED TO CREATE AND EDIT DATA AND PROGRAM FILES**



**LANGUAGE EDITORS**
**ARE SPECIALIZED SUBSYSTEMS OF THE GENERAL EDITOR DESIGNED TO HELP CREATE AND EDIT SOURCE LANGUAGE PROGRAMS**

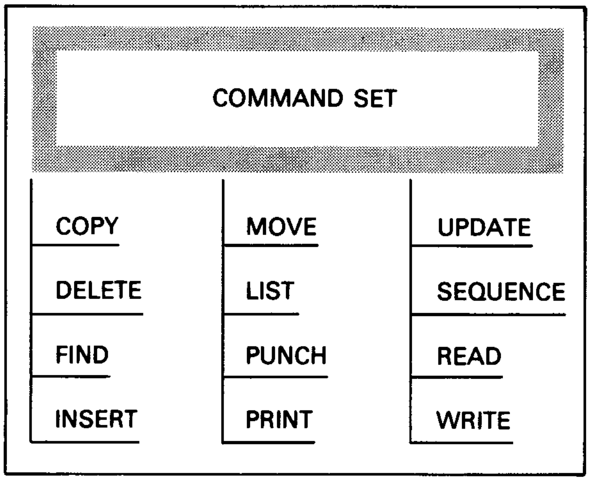


```
     :
RPGEDT VER #
SELECT MODE ( )
 C=CREATE
 U=UPDATE
SELECT FORMAT
TYPE ( )
     :
```

The files created with the general editor and the language editors are displayed at the workstation screen as they are being created – making it easy to spot and correct errors as they occur. The same is true during an editing session – the files being edited are displayed at the workstation screen. When an editing command is keyed in, the user can see its effect on the file immediately.

The versatility of the general editor can best be described by examining the features it offers: commands, command modifiers, and procedures.

The general editor provides an easy-to-learn interactive command language. The command set calls these functions:

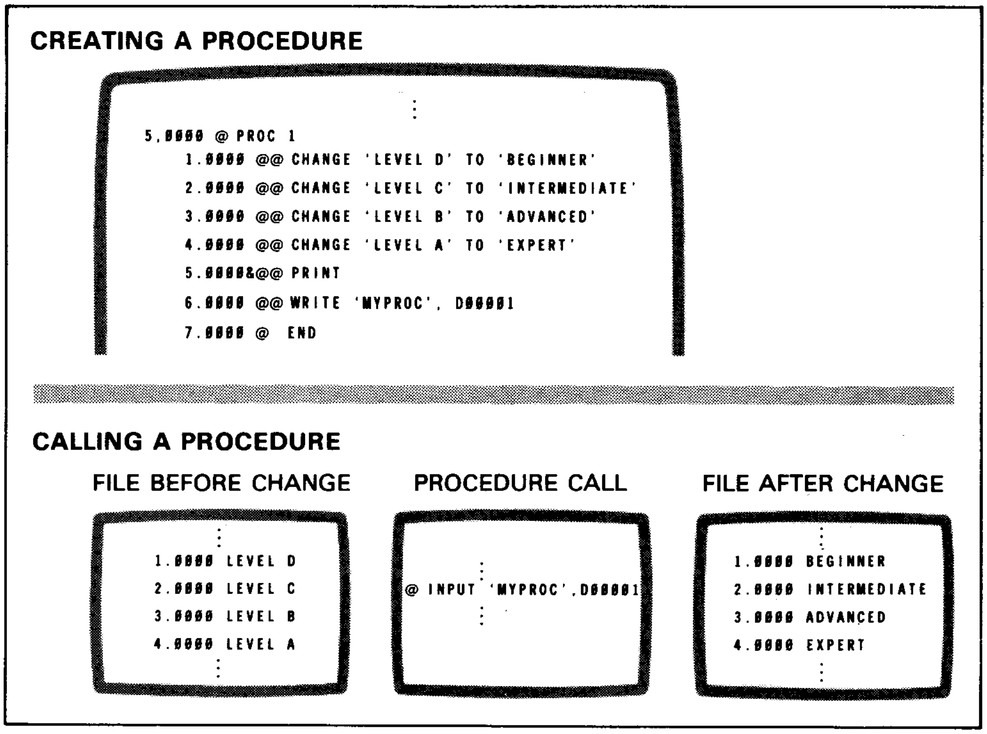

COMMAND SET

COPY
DELETE
FIND
INSERT
MOVE
LIST
PUNCH
PRINT
UPDATE
SEQUENCE
READ
WRITE

This is only a partial list of the functions available through simple, one-word commands.

The power of the general editor commands is enhanced even further by command modifiers that can be appended to each command. The general editor can be told to change all ABC character strings in a file to XYZ character strings, for example. There are many command modifiers available, but they all serve the same function – they tailor the general editor commands to the user's specific needs.

The general editor can also be used to create and call procedures. A procedure is simply a series of commands that perform a specific function. Once created, a procedure can be called with a simple command at the workstation keyboard. The procedures created with the general editor are stored in the editor work space file.


**CREATING A PROCEDURE**

```
                          :
5,0000 @ PROC 1
    1.0000 @@ CHANGE 'LEVEL D' TO 'BEGINNER'
    2.0000 @@ CHANGE 'LEVEL C' TO 'INTERMEDIATE'
    3.0000 @@ CHANGE 'LEVEL B' TO 'ADVANCED'
    4.0000 @@ CHANGE 'LEVEL A' TO 'EXPERT'
    5.0000&@@ PRINT
    6.0000 @@ WRITE 'MYPROC', D00001
    7.0000 @  END
```

**CALLING A PROCEDURE**

FILE BEFORE CHANGE

```
      :
1.0000 LEVEL D
2.0000 LEVEL C
3.0000 LEVEL B
4.0000 LEVEL A
      :
```

PROCEDURE CALL

```
         :
@ INPUT 'MYPROC',D00001
         :
```

FILE AFTER CHANGE

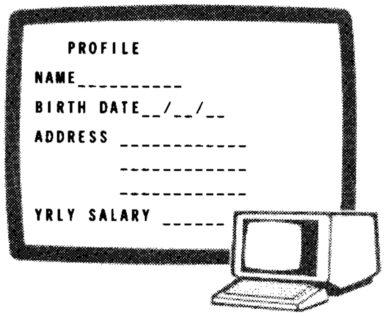
```
      :
1.0000 BEGINNER
2.0000 INTERMEDIATE
3.0000 ADVANCED
4.0000 EXPERT
      :
```

The general editor also provides file protection. When editing a program or data file, a copy of that file is moved to the general editor work space file. The original file is left unchanged as a backup file and is not deleted until requested to do so.

To summarize, the general editor and language editors of OS/3 are powerful interactive tools that help the user create and maintain all his computer-based files.

## 2.2.3. Screen Format Services

OS/3 screen format services simplify and standardize the process of entering variable data into a computer system.



```
   PROFILE
NAME__________
BIRTH DATE__/__/__
ADDRESS ____________
        ____________
        ____________
YRLY SALARY ______
```

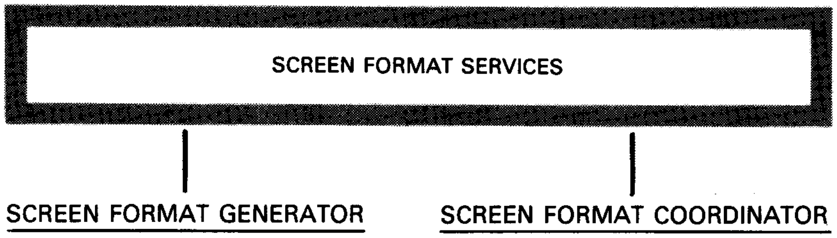
**SCREEN FORMAT SERVICES LET THE USER:**

Design his own computerized fill-in-the-blank forms (screen formats)

Route data entered in response to a screen format back to the application program that requested it

Display *filled in* screen formats to a workstation for informational purposes

Screen format services are comprised of two software components – the screen format generator, which helps the user create screen formats, and the screen format coordinator, which manages screen formats once they are created.



SCREEN FORMAT SERVICES

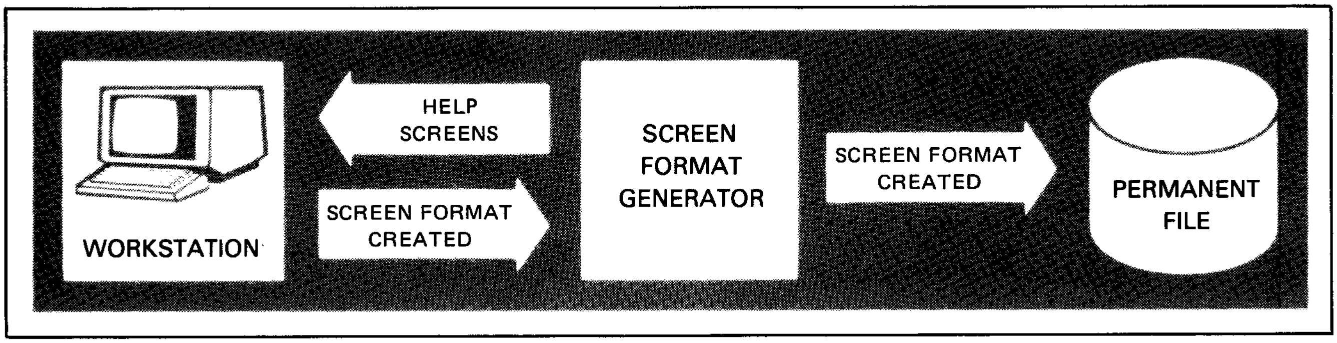
SCREEN FORMAT GENERATOR

Creates, stores, and maintains screen formats at the user's direction

SCREEN FORMAT COORDINATOR

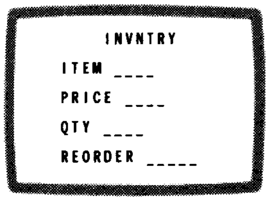
Displays screen formats in response to a program request; routes data entered in response to screen format back to the program that requested it; displays filled-in screen formats at workstation for informational purposes; rejects invalid data as defined at screen generation time.

The screen format generator is activated by a simple command at the workstation. If the user needs assistance to create a screen format, it prompts him with HELP screens that explain the creation process. In addition, the screen format being created is displayed at the same time, making it easy to lay out a screen format exactly as desired. The screen format generator solicits information from the user about the variable data fields of the screen format, including whether they are to be used for input, output, or both. Once the screen format is created, the screen format generator automatically stores it in a permanent library file.



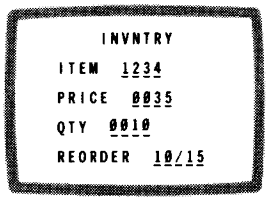
The screen format coordinator is activated in response to a program request. It fetches the appropriate screen formats from the permanent file and displays them - either as blank screen formats or with the variable data filled in. The variable data is stored with the program. If the screen format called by a program was created as both an input and output screen format, the screen format coordinator handles the display of the screen format and variable data, accepts new variable data from the workstation user, and routes that data back to the application program that called the screen format coordinator.

### SCREEN FORMAT COORDINATOR DISPLAYS INPUT SCREEN FORMATS AND THE WORKSTATION USER SUPPLIES THE VARIABLE DATA



```
      INVNTRY
ITEM ____
PRICE ____
QTY ____
REORDER _____
```

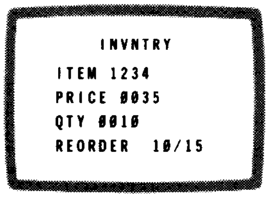
Variable data fields are underscored.



```
      INVNTRY
ITEM  1234
PRICE  ØØ35
QTY  ØØ1Ø
REORDER  1Ø/15
```

Workstation user supplies variable data that is routed by the screen format coordinator to an application program.

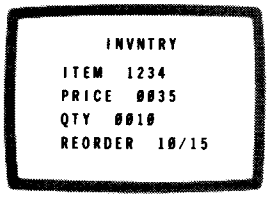
### SCREEN FORMAT COORDINATOR DISPLAYS OUTPUT SCREEN FORMATS FOR INFORMATION PURPOSES



```
      INVNTRY
ITEM 1234
PRICE ØØ35
QTY ØØ1Ø
REORDER  1Ø/15
```

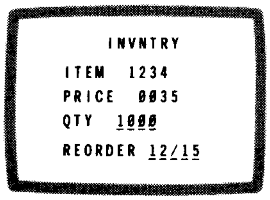
The screen format coordinator fetches the screen format from the permanent file and adds the variable data from the application program.

### SCREEN FORMAT COORDINATOR DISPLAYS INPUT/OUTPUT SCREEN FORMATS



```
      INVNTRY
ITEM  1234
PRICE  ØØ35
QTY  ØØ1Ø
REORDER  1Ø/15
```

The screen format coordinator displays an output screen format and...

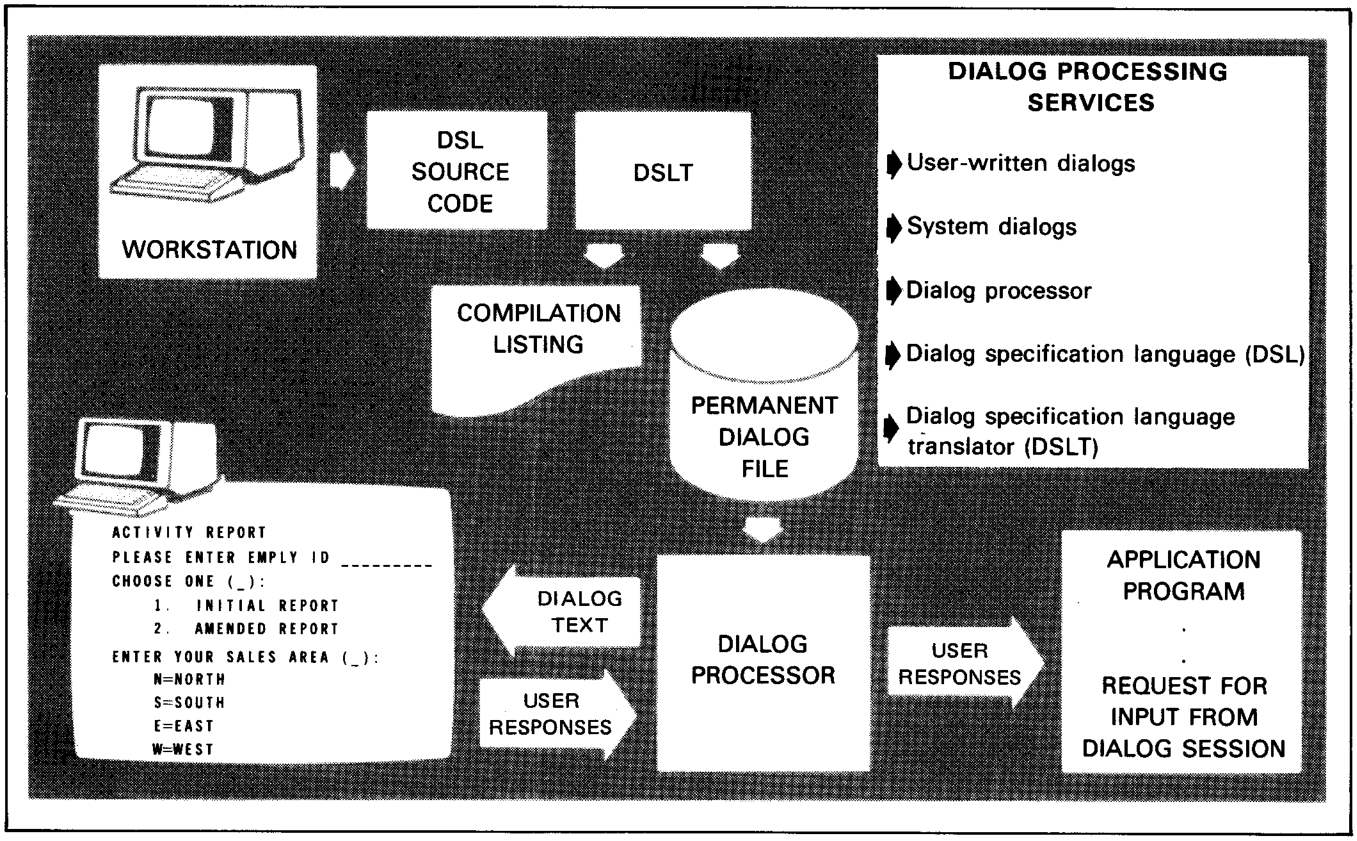


```
      INVNTRY
ITEM  1234
PRICE  ØØ35
QTY  1ØØØ
REORDER 12/15
```

...the workstation user supplies new variable data.

## 2.2.4. Dialog Processing Services

OS/3 dialog processing services make it easier to communicate with the programs and procedures in operation by providing an interactive, conversational link to the computer system.



```
ACTIVITY REPORT
PLEASE ENTER EMPLY ID _________
CHOOSE ONE (_):
    1.  INITIAL REPORT
    2.  AMENDED REPORT
ENTER YOUR SALES AREA (_):
    N=NORTH
    S=SOUTH
    E=EAST
    W=WEST
```

System dialogs are interactive dialogs available from Sperry Univac that guide users through the processes of system generation, building job control streams, and initializing data utilities routines.

DSL is a programming language designed by Sperry Univac specifically for the creation of interactive dialogs. The user can use DSL to write interactive dialogs. DSL source code is submitted to the DSLT, which compiles it to produce the desired dialog and stores the dialog in a permanent file. The DSLT also produces a compilation listing.

Both user-written and system dialogs are managed by the dialog processor, which displays dialog text at a workstation screen, accepts user responses to the dialog, and routes that input to the application program that the dialog is designed to complement. In addition, the dialog processor produces a printed summary of each dialog session and (if requested) an audit file that contains a complete record of dialog responses. Besides providing a record of the dialog sessions, the printed summary and audit file can be used as guides to changing dialog responses in a subsequent session.

The application programs that call for input from a dialog session can be written in any language.

## 2.2.5. System Dialogs

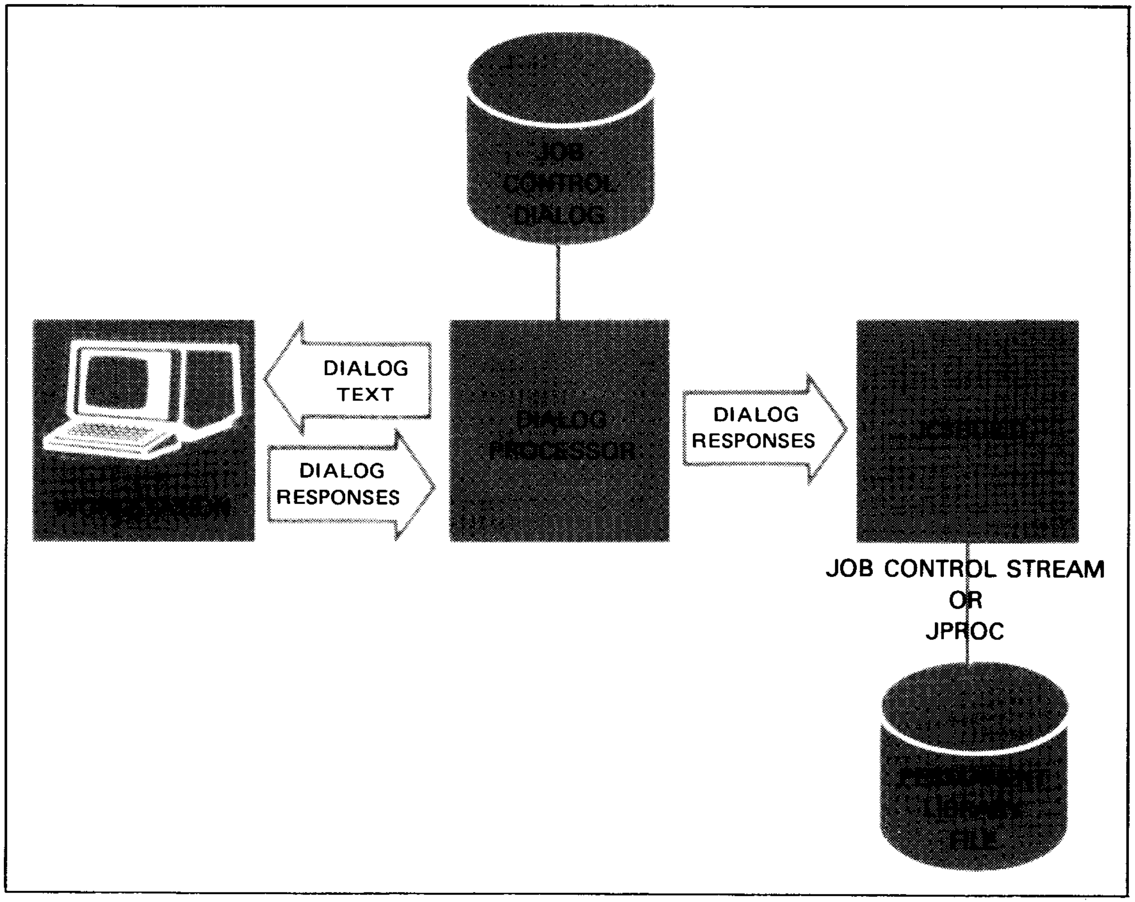
System dialogs, as previously noted, provide an interactive, conversational interface to the processes of system generation, building job control streams and job control procedures (jprocs), and initializing data utilities routines. In addition, they are designed to provide the user with tutorial assistance, when needed, in the form of HELP screens. HELP screens describe system concepts and specific dialog choices that need explaining in order to make a valid choice. System dialogs that use the dialog processor also offer an audit facility that allows the changing of dialog responses in a subsequent session.

## 2.2.5.1. Job Control Dialog

The job control dialog guides the user step by step through the process of building a valid job control stream or jproc. The statements and system jprocs that make up a job control stream are presented in the form of menu items, and the user is asked to choose those that are desired.

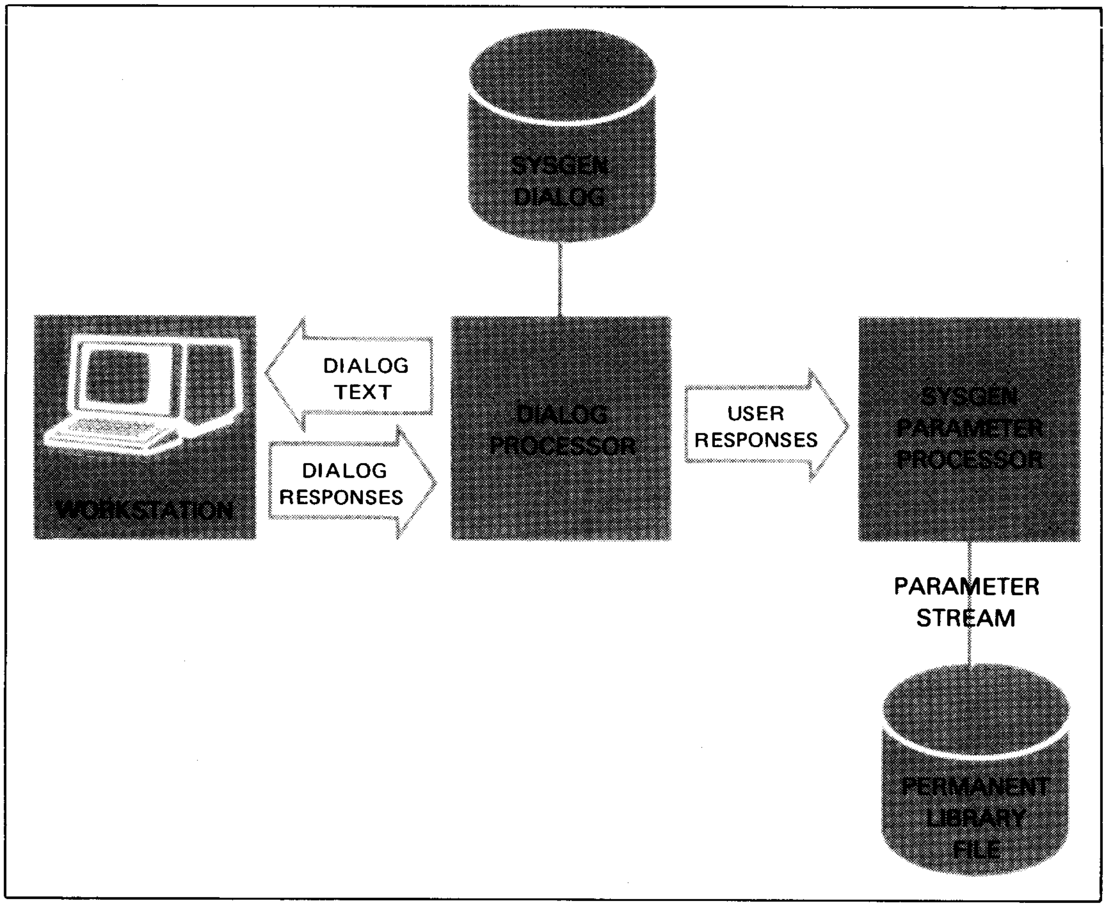
HELP screens can be displayed upon request whenever a dialog choice or a particular job control statement or system jproc explanation is needed. A more experienced job control user can use the dialog to build a control stream without using HELP screens – thus building the control stream quickly while still being constrained by the dialog to build a syntactically correct control stream. Novice users, on the other hand, can learn about job control as they are actually building a valid job control stream.

The job control dialog is initiated with a simple workstation command that activates the system program JC$BUILD. JC$BUILD activates the dialog processor, which then manages the dialog session with the workstation user. The workstation user's responses to the job control dialog are routed back to JC$BUILD, which uses them to create a control stream or jproc and then stores the control stream or jproc in a permanent library file for future use. The user control stream can then be initiated with a run command from the workstation. Jprocs, of course, are initiated when the control stream that references them is run.



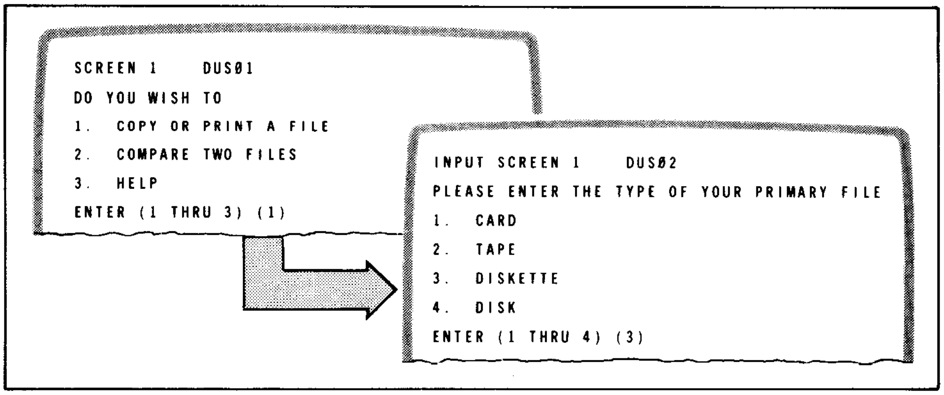
### 2.2.5.2. System Generation Dialog

The system generation (SYSGEN) dialog operates in the same manner as the job control dialog. It solicits SYSGEN parametric values that configure the operating system best suited for the user's processing needs. The SYSGEN dialog, like the job control dialog, is initiated with a workstation command. Responses to the dialog are routed to the SYSGEN parameter processor and then stored in a permanent library file.



### 2.2.5.3. Data Utilities Dialog

The data utilities dialog operates in a manner similar to the job control and SYSGEN dialogs, except that it does not use the dialog processor. It offers the same advantages, however. A simple workstation command activates the data utilities dialog, which then solicits information about the data utilities functions needed for use. HELP screens are available that explain the dialog choices. When the dialog is completed, the functions requested are automatically initialized.



```
SCREEN 1     DUSØ1
DO YOU WISH TO
1.  COPY OR PRINT A FILE
2.  COMPARE TWO FILES
3.  HELP
ENTER (1 THRU 3) (1)
```

```
INPUT SCREEN 1     DUSØ2
PLEASE ENTER THE TYPE OF YOUR PRIMARY FILE
1.  CARD
2.  TAPE
3.  DISKETTE
4.  DISK
ENTER (1 THRU 4) (3)
```

## 2.3. ENHANCED PROGRAMMING LANGUAGES

System 80 features a full range of versatile problem-oriented programming languages. These languages meet established standards and offer extended features unique to System 80. The languages provide the capability for programs to access workstations for dynamic data entry and display during execution. Language editors are available to support the interactive development of programs at the workstation. The implementation of a consolidated data management system offering a uniform file type specification for all devices offers a significant degree of device independence to user programs. These are but a few of the enhanced features available to the System 80 programmer.

Two of the programming languages supported, BASIC and ESCORT, are fully interactive. Source statements for these languages can be entered directly through the workstation, with error messages provided for the isolation of syntax errors. Users can quickly and easily build programs with these languages.

### ENHANCED PROGRAMMING LANGUAGES

#### ESCORT

ESCORT is an interactive language especially suited for developing file processing programs. This language is among the easiest languages to learn and use, and even the most inexperienced programmers can quickly and easily begin generating useful and productive programs. ESCORT offers extensive prompting at the workstation; users can get quick assistance by asking the system for help.

#### COBOL

The Common Business Oriented Language (COBOL) is a general purpose, yet powerful language designed with the business community in mind. COBOL offers a full range of programming services to meet common commercial needs, such as payroll, accounting, inventory, and personnel management. The COBOL supplied by Sperry Univac also offers a powerful sorting facility allowing users to sort data files on multiple keys. In addition, COBOL users can readily interface with the IMS and the DMS.

#### FORTRAN IV

FORmula TRANslation (FORTRAN IV) is a powerful computing language that meets the problem-solving needs of the scientific community. Businesses that require a great deal of mathematical computing power should find FORTRAN IV an invaluable programming tool.

#### RPG II

The Report Program Generator II (RPG II) is a high-level language suitable for producing reports and maintaining files in a business environment. The basic logical flow of a program is provided to the user by the RPG II cycle. The user specifies the details of processing by using the input format, output format, calculations, and other specifications. Entering RPG II programs through the workstation has been simplified by the RPG II Editor; source program statements may be entered on the workstation by using formatted displays.

#### BASIC

The Beginner's All-purpose Symbolic Instruction Code (BASIC) is a simple, straightforward programming language designed for general business applications. The salient feature of BASIC is its ease of use. BASIC is an interactive language that uses simple, English-language statements.

#### BASIC ASSEMBLY LANGUAGE

The Basic Assembly Language (BAL) is a flexible, machine-level language that offers a variety of features that allow the user to specify the most complex algorithms as a series of mnemonic symbols and command directives. Each mnemonic represents a single machine action. BAL is a complicated programming language that can be cumbersome to the inexperienced user, but can be of great value to the user with particular programming requirements.

## 2.4. INFORMATION MANAGEMENT SYSTEM

With the increased use of computers to store day-to-day business information came the need for users to be able to access and update that information quickly and easily. Sperry Univac meets that need with the information management system (IMS). IMS is a transaction processing system; all activities are initiated by an inquiry, and each inquiry results in a system response. IMS allows nonprogramming personnel to interactively access and manipulate large-scale data files through a network of workstations and terminal devices. Through IMS, any authorized member of an organization can use a terminal in the IMS network to instantly obtain information in the data files and to make changes to the data file. IMS is easy to install and specifically designed for use by computer personnel at all levels. It is modular for simplicity of use and maintenance, and configurable to fit an installation's requirements.

### IMS FEATURES

#### INQUIRY/UPDATE LANGUAGE

IMS supplies an easy-to-use inquiry/update language called UNIQUE for general purpose file processing. Each UNIQUE command initiates a particular file processing activity. The range of commands includes record deleting, modification, and listing the contents of a file. Users with more specific file processing demands can generate and implement their own file processing programs. These programs, referred to as action programs, interpret terminal operator requests, perform the requested function, and issue the appropriate response to the initiating terminal. User action programs can be written in COBOL, RPG II, or the basic assembly language (BAL). IMS handles the scheduling of, and resource allocation for, action programs initiated by terminal operator requests.

#### RELIABILITY

IMS is an extremely reliable system providing extensive features to prevent accidental alteration or destruction of data files. A record lock facility prevents a user from accessing a file that is currently being updated by another user. Files that are accidently or incorrectly modified can be recovered either as they existed before being altered by a transaction (backward recovery) or after modification (forward recovery). A quick-recovery facility provides online backward recovery of all files adversely affected by a general system failure.

#### DATA FILES

Any type of data file can be accessed by IMS, including files generated by the data base management system (DMS). In addition, IMS provides the facilities for logically redefining existing data files to meet changing needs. This is called defined record management. Using defined record management, a user can logically redefine a file without actually altering the physical file structure. Defined record management can also be used to combine selected items from records in a variety of source files into a single defined file to be used by an IMS action program. DMS data bases can be interfaced by generating IMS action programs that include DMS data manipulation language commands or by using defined record management to redefine the data base.

#### INTERACTIVE COMMANDS

In addition, to the terminal transaction processing provided by UNIQUE and action programs, a powerful set of interactive commands is available to assist with the administrative, operational, and educational aspects of IMS. Included is a set of general commands that can be used by individuals to control the processing of their own terminal and a master terminal command set issued from a designated master terminal to control the processing activity of the entire network. A terminal-to-terminal message command permits communication between terminals.

IMS offers a fast, reliable, and interactive method of accessing data files. It is an easy-to-use system capable of making any individual within a user organization into a potential IMS user, yet providing the security and file protection so important when dealing with vital data files.

## 2.5. DATA BASE MANAGEMENT SYSTEM

The data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs handle the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of DMS data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

### DMS FEATURES

PHYSICAL BASE BASE CHARACTERISTICS

The device media control language (DMCL) defines physical characteristics of a user data base and the data dictionary.

LOGICAL DATA BASE CHARACTERISTICS

The description of data in a DMS data base is entirely separate from the manipulation of that data in application programs. This results in a higher level of data independence for application programs. All descriptions are done in a high-level language comparable to the data declaration language of COBOL.

STORAGE STRUCTURES

Data can be organized into three storage structures: sequential, tree, and network. The same record can belong to several different structures simultaneously and is stored directly by data base key, sequentially within area, by calculated key value, or according to its set relationship to other records in the storage structure. This reduces data redundancy and promotes processing efficiency.

MANIPULATION OF DATA

The data base is accessed by using a combination of data manipulation language (DML) statements and conventional COBOL statements in the procedure division of a COBOL application program.

SYSTEM SUPPORT FUNCTIONS

DMS supplies processors and utilities, collectively known as system support functions, that create, establish, and maintain a data base.

IMS/DMS INTERFACE

IMS user-written action programs or UNIQUE action programs can access DMS data bases.

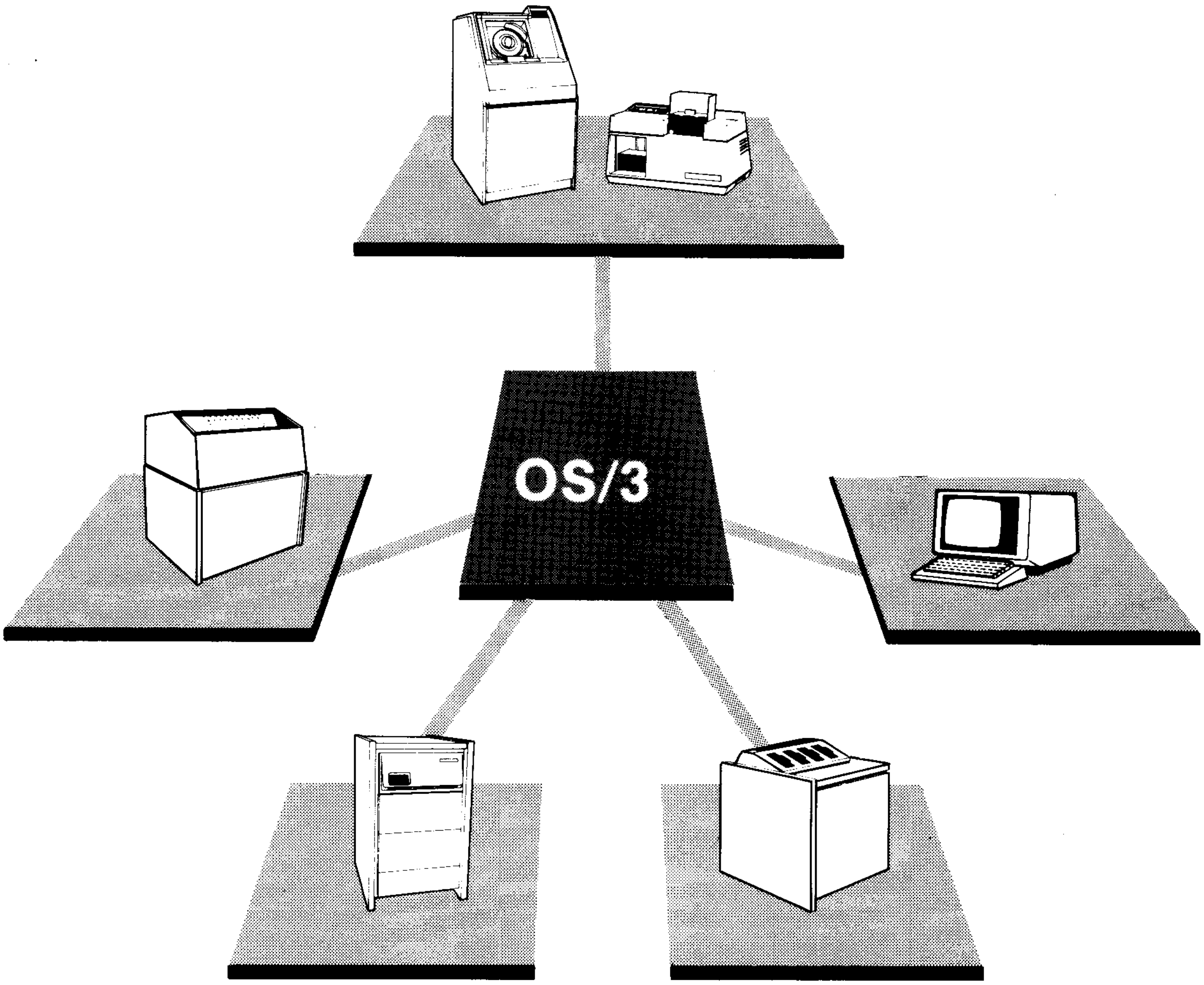
RECOVERY

Offline recovery utilities use a journal file generated by the data base management system for forward and backward data base recovery. Online recovery automatically restores a data base after system crash or abnormal program termination.

## 2.6. JOB PROCESSING

OS/3 is structured to handle multijob processing demands in both batch and interactive processing environments. Up to 14 jobs can be active concurrently in OS/3. A primary user interface to the system is job control, which is the OS/3 component that tells the system (at the user's direction) what hardware and software resources are needed to process a job as directed. The user must prepare a job control stream that specifies these requirements for every job in OS/3. The method of preparing the control stream varies, depending on whether the job is running in a batch or interactive environment. After the job's processing requirements are defined, the method of storing and running jobs also varies, depending on the environment.

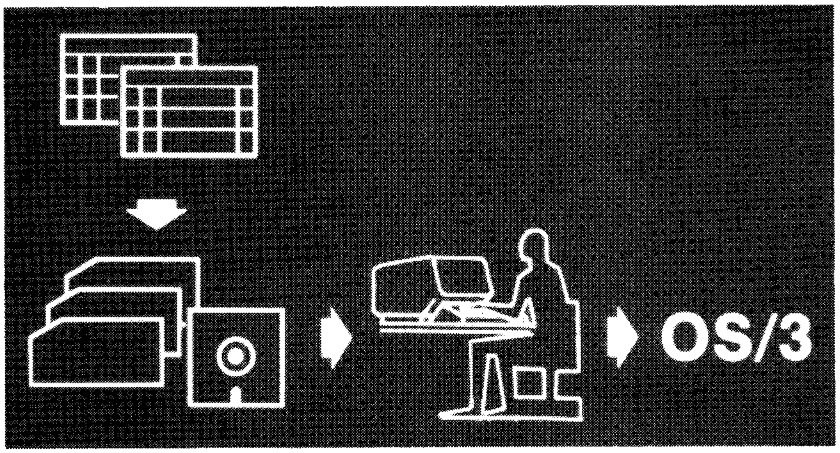
OS/3 HANDLES THE MULTIJOB PROCESSING DEMANDS OF BATCH AND INTERACTIVE ENVIRONMENTS



### 2.6.1. Batch Job Processing

In a typical batch environment, the user builds a control stream for a job by coding and keypunching the stream on cards or a diskette and submitting the job to the system operator to be run. The user can communicate with the system operator through certain job control statements in the control stream that, when processed, are displayed at the system console. The primary responsibility for job processing rests with the system operator once the job is submitted. The user has considerable control over the way the job is processed, however, through the job control specifications. The user can also initiate the running of a job from the control stream of another job through a special job control statement that simulates many system operator functions. Job control streams and programs are stored through system console commands. In a batch environment, the results of the job are seen after it has been run, sometimes hours or days later. This contrasts with an interactive environment, where the user can see the job results displayed at the workstation screen almost immediately.

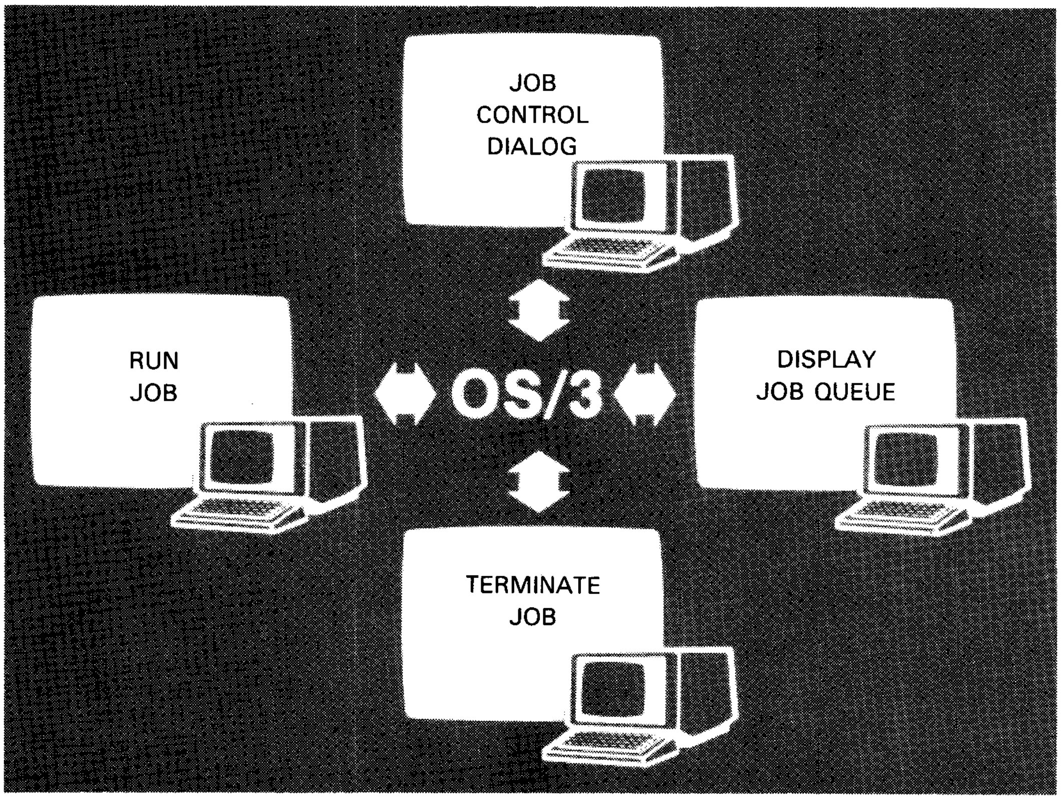
IN A BATCH ENVIRONMENT,
JOBS ARE RUN BY THE SYSTEM OPERATOR



### 2.6.2. Interactive Job Processing

In a typical interactive processing environment, the user communicates directly with the system from a workstation. The user can create program source code, build a control stream for a job, store job streams and programs, and run jobs from a workstation. Certain facilities of interactive job control allow variable values to be submitted for a job at run time, from the workstation, and to actually change the way a job is executed by performing dynamic skip functions from the workstation. The programs and control streams can be stored in permanent files from the workstation by using the general editor and the job control dialog. Other job processing functions, like connecting a workstation to a job, terminating a job, and requesting the status of a job, can be performed through simple workstation commands. In an interactive environment, in short, the user can control job processing from the time program source code is created to the time a job is run and terminated.

**INTERACTIVE JOB PROCESSING PERMITS CONTROL OVER A JOB FROM BEGINNING TO END**



## 2.7. MULTIJOBBING

System 80 can concurrently process from 1 to 14 jobs, with each job consisting of one or more job steps, or tasks, executed serially. Each task can be a program compilation or the execution of a user- or system-supplied program. This capability allows the user greater flexibility in attaining maximum use of the system resources and in scheduling tasks.

OS/3 multijobbing consists of scheduling multiple jobs for concurrent execution. The allocation of processor time is based on a system switch list that contains information regarding program priorities, task synchronization, and input/output utilization. While one task is awaiting the completion of an external event (such as completion of an input/output request), OS/3 activates another task that is ready to run to ensure optimum utilization of the processor's capabilities. Because the majority of programs require support other than processing instructions, OS/3 multijobbing provides an effective method for the user to reduce processor idle time and increase system productivity (throughput).

OS/3 was designed around the multijobbing concept. All the software that Sperry Univac provides in the OS/3 package is designed to take full advantage of this multijobbing environment. The controlling software automatically, concurrently processes any user jobs submitted to the system. The interactive facilities do not occupy a job slot; therefore, they are not included in the multijobbing algorithm. The interactive facilities are handled by the system in such a way that each individual workstation operator receives optimum response times to entered requests.

## 2.8. DYNAMIC RESOURCE MANAGEMENT

The OS/3 software has the capability of dynamically managing the resources available for use in the system. This dynamic management is provided for both interactive programming and batch activities. The system can dynamically manage such things as the main storage assigned to a job, unassigned main storage, workstations, and the allocation of peripheral devices, such as disks and diskettes. The dynamic management of system resources frees the interactive user from constraints of the physical devices and software components planned for use. In addition, the batch user with a program interfacing with a workstation user cannot always predict the extent of the resources the job requires. Thus, dynamic resource management frees all users from such dependencies and increases the efficiency and throughput of the entire operating system.

## 2.9. DATA COMMUNICATIONS

OS/3 provides an integrated, modular communications package designed for ease of operation while offering the communications user a system that provides a broad range of capabilities. This package, the integrated communications access method (ICAM) terminal support facility, can be configured to meet the needs of the individual system. ICAM offers several levels of support, each of which balances the services provided with the amount of system resources required. The small user can configure a simple, yet effective system without causing a severe strain on resources. The communications-oriented user can build a complex communications system that provides a full range of message handling functions.

ICAM provides the communications capability required to support a number of other Sperry Univac components. Among these are.

### ICAM COMMUNICATIONS SUPPORT

- Information Management System
- Distributed Data Processing
- RPG II Telecommunications
- COBOL Communications
- Remote Batch Processing
- Nine Thousand Remote (NTR) Processing

These functions are performed independently from the operation of ICAM because the comunications system has been designed to offer total device independence to both the user programs that interface with the system and to the system programs that interface with ICAM.

ICAM allows the user to configure a variety of terminal device types and communications lines into a single network usable by a number of user and system programs concurrently. ICAM has the responsibility for preventing conflicting resource assignments and for releasing facilities when jobs terminate.

### ICAM PROVIDES

- The queueing of messages in main storage or disk (a user option) while waiting for servicing
- Specification of multiple destinations for a single message
- Scheduling of activities and support of user-defined priority levels
- Timer services to time the occurrence of events
- Reconstruction of messages and queues from user-specified points in the event of system failures
- Accumulation of relevant statistical information

An ICAM configuration includes: the software modules included in the system during the system generation procedure; the communications lines; the terminal devices; the system utilities that the ICAM network requires; plus the programs that are to interface with the network.

OS/3 communications supports two basic types of communication: narrowband and wideband transmissions. Narrowband transmission permits voice-grade communications (referred to as dial-up), switched DDD (direct distance dialing), or privately leased lines. Wideband transmission permits data transmission over high-speed privately leased lines. Sperry Univac provides full- and half-duplex interfaces for its own terminals, as well as commercially available data sets.

## 2.10. DISTRIBUTED DATA PROCESSING

The SPERRY UNIVAC distributed data processing (DDP) system allows users having a number of separate processing systems to tie those systems together in a network so that all systems can share the processing load of the entire organization. In such a distributed data processing network, a user at one site can control the operations of another site and can perform such functions as:

- site-to-site file transfers;
- initiation of jobs at the remote site; and
- inputting of data through the local site to the remote site for processing.

The software required to support a DDP nework includes a configured communications network to physically link the included sites, and distributed data processing processors at each site. The DDP processors interpret all DDP related commands and perform the requested functions. The DDP processors are designed to operate in an interactive environment and respond to commands issued from a workstation.

The DDP software provides these capabilities:

- Site-to-site data file and program library transfers
- Remote job initiation and control
- Operator console control over remote site and routing of message to remote operator console

The DDP software can be used to copy data files and program libraries from one system to another or to add data to an existing data file in a remote system. Data files can be deleted from a remote system. Data files and program libraries can be transferred between OS/3-based systems.

Users can utilize the DDP commands to submit and initiate a job to a remote system through the local system. Output from the job can remain at the processing site or can be directed back to the initiating site. The DDP software can take advantage of spooling services if they are included in the system. Jobs running at another site can also be cancelled from the initiating system.

Messages can be sent to the system console of a remote system and the answer routed back to the initiating device. A device on a system can, in a limited way, operate as the system console for a remote system.

The capability to distribute the processing work load among systems in a DDP network can provide improved business operations and management control. Jobs can be decentralized and given to the location responsible for gathering and using the data. Distributed data processing also provides greater control over work priority, improved response times, and a recovery system in the event of local system failures.

## 2.11. APPLICATIONS PROGRAMS

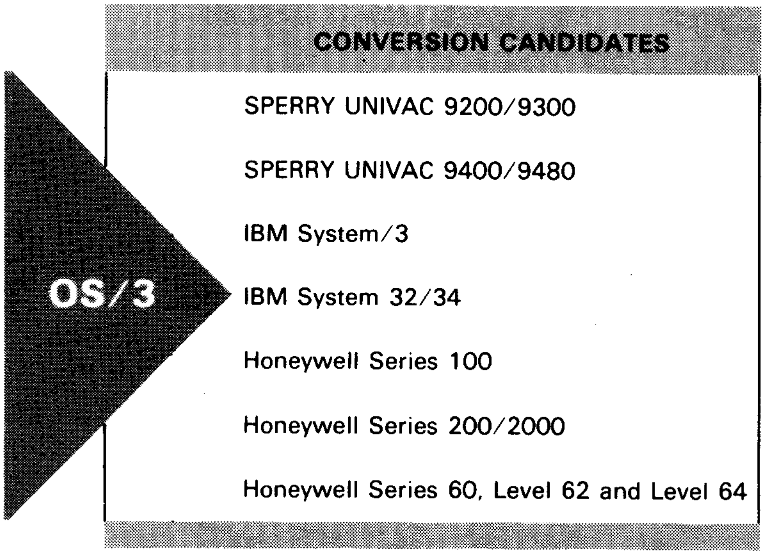
OS/3 readily lends itself to the support of diverse needs of the user organization. In addition to the major standalone applications systems, such as the UNIVAC Industrial System, the OS/3 applications programs library contains over 700 programs and subroutines for interactive and batch use in the solution of a wide variety of business and technical problems. The interactive nature of OS/3 makes it especially suitable for the support of these operations without any unusual scheduling or data entry impact on the primary system users.

The list of available applications programs constantly fluctuates as Sperry Univac adds new programs, and modifies existing software, to meet the ever-changing needs of today's business community. Users wishing to know if Sperry Univac has an applications program that suits their needs can do so by contacting their local Sperry Univac representative.

## 2.12. CONVERSION AIDS

To convert an existing system to an OS/3 system, data files must be transferred to the peripherals in the new system, and those program elements not accepted by the new system must be changed. If the user is satisfied with the structure of the existing data base, file conversion becomes a relatively simple question of dumping existing files to a transportable medium and restoring them on the new system. If the instruction repertoire of the new system is substantially identical to that in the existing system, the conversion task is reduced to modifying the macros, file definitions, and incompatible instructions in the program.

To meet the needs of a converting user, Sperry Univac supplies a full range of conversion tools for a number of systems.



CONVERSION CANDIDATES

- SPERRY UNIVAC 9200/9300
- SPERRY UNIVAC 9400/9480
- IBM System/3
- IBM System 32/34
- Honeywell Series 100
- Honeywell Series 200/2000
- Honeywell Series 60, Level 62 and Level 64

## 2.13. AVAILABILITY, RELIABILITY, AND MAINTAINABILITY

A prime consideration when purchasing a new computer system is the amount of productive computing time the user can expect from the system. The best computing system is of little value if it is unavailable for use because of a system failure. Sperry Univac recognized this and designed the hardware and software for System 80 to be as available as possible by using reliable components and including extensive automatic recovery and maintenance features.

The hardware represents the state of the art not only in technology, but also in reliability. System 80 is designed to perform effectively under various conditions, and extensive testing ensures that any installed hardware is in perfect operating condition and was not damaged in transit.

The OS/3 software has proven itself through years of reliable service. It has enjoyed the long-term success associated with an effective and reliable product. However, even the most reliable systems available can suffer an occasional failure. To ignore the possibility would be unwise; therefore, Sperry Univac included the features needed to ease the impact of a system breakdown and to get the system up and running as quickly as possible. Sperry Univac supplies its customers with an impressive array of diangnostic, backup, and recovery facilities. These features cover all possible software and hardware failures. The system even monitors its own operation to help the user predict the possibility of a failure.

To make System 80 more reliable, Sperry Univac regularly issues software maintenance packages. These packages contain changes that can enhance system stability or forestall potential difficulties. Additionally, Sperry Univac responds to maintenance requirements by providing individually tailored software maintenance changes.

# 3. System Software

## 3.1. SOFTWARE DESIGN CONCEPTS AND CAPABILITIES

Through centralized control of all activities of System 80, the combined hardware and software capabilities are fully established and maintained to satisfy the requirements of all applications. The responsibility for efficient and flexible centralized control is borne by OS/3, which allows the programmer to use the system with relative ease, while relieving him of concern for the internal interaction between his program and other coexistent programs.

OS/3 is a comprehensive library of programs consisting of an executive, a collection of language processors, utility routines, and application programs. The programs that make up the executive are the OS/3 Supervisor and Job Control. Through its versatile job control language, OS/3 organizes and directs operations and system activities to achieve maximum use of computer facilities. The OS/3 supervisor provides the central control, interface, coordination, and allocation of the hardware to achieve optimum system utilization. The supervisor also controls the initiation, loading, execution, and termination of user jobs.

OS/3 has been designed and implemented to offer an efficient multijobbing environment that is needed to utilize the full capabilities of System 80.

The speed and hardware capabilities of System 80 are used to maximum advantage, and a given hardware configuration is used most effectively in the complex internal operating environment created by OS/3. This environment allows for the concurrent operation of programs; for immediate reaction to the inquiries, requests, and needs of many different users at remote and local workstations and terminals; for storage, filing, retrieval, and protection of large blocks of data; and for optimum use of all available hardware facilities while minimizing job turnaround time.

Ease of use is emphasized in the system. Interactive facilities simplify the user-to-system interface, allowing nonprogramming personnel to use the system. Work to be performed by the system is described through the OS/3 job control language, which was designed to minimize job turnaround time and operator intervention. The user may construct any logical combination of programs for a particular job by inserting the proper control statements in his job stream.

Jobs can be collected and entered into the system from many sources - central or remote. The executive controls the loading, allocation, and execution of the programs once they have been entered. Job steps that cannot be completed because of program error are automatically deleted from the system with appropriate diagnostic information. The console operator is, in effect, responsible only for participation in the data processing activities as directed by the executive.

## FUNCTIONAL CAPABILITIES OF OS/3

### Interactive Processing

The executive provides the services required to meet the unique needs of the interactive user. It interfaces with the user through the workstation and with the interactive services to ensure that all activities proceed efficiently and that all interactive workstation users can perform their operations free of interference from other users. It coordinates the interactive activities with all other activities to provide a balanced processing environment so that the needs of one type of user do not interfere with those of another.

### Communications Processing

The executive efficiently responds to the demands of communications processing and gives preference to the operational needs of a communications system, these being the most critical requirements of the system. The executive functions dedicated to this type of processing receive the highest priority for scheduling within the system. The contingencies of message control are supported by procedures to roll out conflicting user programs, restart the system, and perform other necessary functions.

### Batch Processing

Facility of job preparation and submission, with minimization of job turnaround time, is a design feature of the executive. User-assigned priority by job can provide preferential service when batch jobs are submitted from workstations or remote terminals or where turnaround time is critical.

The executive provides an operational environment for a high volume of jobs. The user may specify preferred priorities for certain jobs within a group of jobs; this fulfills the user's responsibility related to multijobbing scheduling to achieve machine optimization. Automatic job-to-job transition, communication within jobs, and associated services are automatically provided by the system.

### User Program Development

The executive interfaces with a specific set of source code language processors and editors that enable the user to write programs in COBOL, ESCORT, BASIC, FORTRAN IV, basic assembly language (BAL), and report program generator (RPG II). The output from any one of these language processors (object module), when processed by the linkage editor, will produce a program (load module) that can be executed on System 80.

### Disk-Oriented Processing

The executive uses disk storage as buffers for accommodating any job backlog and for storing output data resulting from executed jobs. The buffering allows the system to operate independently of the essentially low-speed peripheral devices. All executable programs are obtained from disk storage through one of the system libraries maintained by job control. Temporary files required during program execution are generally assigned to disk storage rather than magnetic tape to facilitate disk-oriented processing. Operator participation is explicitly defined and is minimized as much as possible.

The programs of the OS/3 Executive – the supervisor and job control – provide control for coordinating and executing system-supplied and user programs and for furnishing a flexible processing environment. In concert with a collection of programming languages, utility routines, and application programs, Sperry Univac provides the user with a viable program library to take full advantage of the extended capabilities of System 80.

## 3.2. SYSTEM CONTROL SOFTWARE

### 3.2.1. Supervisor

The supervisor is a collection of sophisticated routines that provide the central control needed for the system's hardware and software, user programs, and interactive facilities to work together efficiently. Without the services provided by the supervisor, such features as multiple program processing, simultaneous interactive access, error control and recovery, and automatic resource management would not be possible. The supervisor manages and coordinates all system activities, handles randomly occurring events, initiates and coordinates the execution of batch and interactive programs, and provides advanced programming facilities. It is one of the most complex components in the system, yet the supervisor is what makes the System 80 an easy-to-use and efficient data processing system.

The supervisor is built around executable modules, or routines, each of which has a specialized function. Those routines commonly used by the supervisor always reside in main storage. Other less often used routines, called transients, are stored on the system resident volume and are loaded in main storage only when the supervisor needs them. This arrangement promotes supervisor efficiency: it minimizes the amount of main storage the supervisor uses by overlaying unneeded transients with newly loaded transients, and it eliminates the input/output time needed to load the most commonly used routines by keeping them resident.

**SUPERVISOR FUNCTIONS AND SERVICES**

- Supervisor interrupt requests – service the various interrupts generated by the system.
- Task switcher – determines the order in which the various tasks are given CPU time.
- Physical input/output control – controls the dispatching, queueing, and interrupt processing for all I/O devices directly connected to the system.
- Transient management – schedules, locates, and loads transient (nonresident) supervisor routines.
- Timer and day clock services – provide a system clock for timed, batch, or interactive activities.
- System console and workstation management – controls information to and from the system console and various workstations configured.
- Error recovery and logging – handles any program or machine error that causes an interrupt.
- Main storage management – allocates and deallocates main storage as required by other system control software.
- Diagnostic services – support a variety of diagnostic services to isolate and correct diverse error conditions.

The supervisor is configured during the system generation procedure (SYSGEN). During SYSGEN, users enter supervisor-related parameters to indicate those supervisor features they want to include as resident and as transient (nonresident). Additional features increase the main storage requirements of the supervisor. (This may be a consideration for smaller systems.) A number of supervisor generation parameters provide default values the supervisor used to handle certain conditions. Users can generate a number of separate supervisors during system generation, but only one supervisor at a time can be operating.

### 3.2.1.1. Supervisor Interrupt Requests

A supervisor interrupt is a request made to the supervisor to perform a function. Requests are generated by the various hardware and software components. They are called interrupts because they interrupt normal processor flow and must be handled in some way before processing can continue. OS/3 recognizes eight types of interrupts:

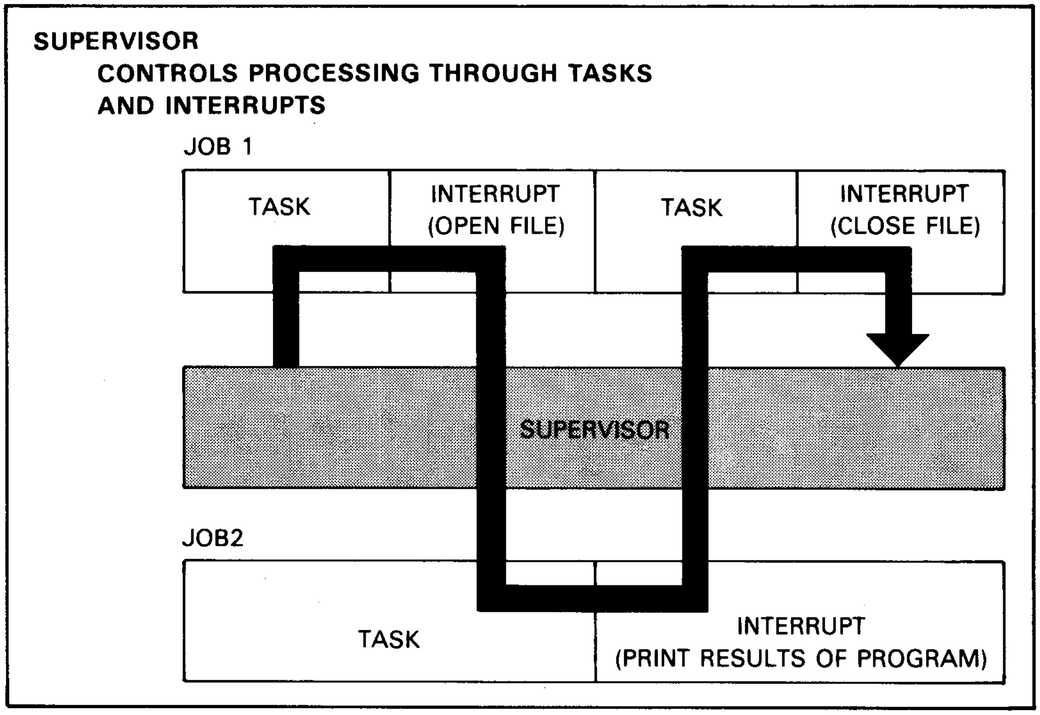
**INTERRUPTS**

- Supervisor call - occurs in response to the SUPERVISOR CALL (SVC) machine instruction. Although it is handled as an interrupt, programs routinely use the supervisor call to request supervisor services.
- Exigent machine check - indicates a malfunction in or around the processor from which the supervisor cannot recover.
- Repressible machine check - indicates a malfunction in or around the processor from which recovery is possible.
- External interrupt - generated either by the processor interval time or the system console interrupt key.
- Program check - occurs when the processor attempts to execute a nonexistent instruction or to execute an existing instruction in an illegal manner.
- Program event recording (PER) - provides dynamic monitoring of executing programs by storing information about the current instruction when a specified event occurs.
- Input/output - occurs in response to signals from I/O channels.
- Restart - occurs when the restart key on the system console is pressed and can be used to put a stopped processor in the operating state.

Some interrupts, like supervisor call or input/output, are routinely encountered; others, like program or machine checks, represent serious errors that the supervisor must handle with minimal system interruption.

### 3.2.1.2. Task Switcher

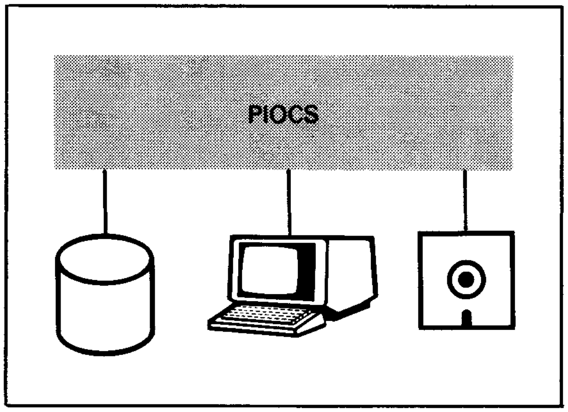
Any activity that uses the central processing unit (CPU) is called a task. The system and users can initiate tasks. User-initiated tasks are interactive activities or functions requested through job control. The system initiates tasks to support user requests or as a part of normal system operation. The mechanism that coordinates the processing of all current tasks is the task switcher. The task switcher decides, based on established priority, which one of the tasks awaiting execution should be processed next.



SUPERVISOR
CONTROLS PROCESSING THROUGH TASKS AND INTERRUPTS
JOB 1
TASK
INTERRUPT (OPEN FILE)
TASK
INTERRUPT (CLOSE FILE)
SUPERVISOR
JOB2
TASK
INTERRUPT (PRINT RESULTS OF PROGRAM)

### 3.2.1.3. Physical Input/Output Control System

OS/3 performs all input/output (I/O) operations with peripheral devices through the physical input/output control system (PIOCS). PIOCS handles the queueing and initiation of all I/O commands and the processing of I/O interrupts. OS/3 PIOCS is composed of general purpose software routines designed to provide maximum throughput on all peripheral devices and to allow for ease of expansion to support new devices.



PIOCS receives control when data management issues a request for an I/O operation. Control is not returned to data management until the I/O request has been completed. However, other tasks in the system may be activated if their status indicates a ready-to-run condition. Requests for I/O operations are initially queued, by priority, in device and channel queues. Dispatching follows the queueing of an I/O order if the channel and device are free or, if not, upon completion of a previous order on the same channel. The I/O dispatch routines perform service functions, as required, such as disk address verification and parameter checking. Interrupts from I/O channels are serviced as a high priority function of the supervisor to free the channel for dispatching other I/O orders that may have been queued. Upon completion of an I/O order, a general I/O status analysis is performed to determine whether any abnormal conditions occurred. For normal I/O terminations, control returns to the I/O dispatcher. If no additional requests remain in the queue, control passes to the task switcher for return of control to another task.

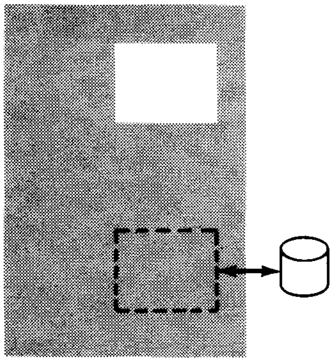
If an error condition occurs, the appropriate device is flagged as unavailable for all tasks. For an error on the system resident device, the resident error recovery routine receives control.

Other error conditions require more detailed analysis. A device error recovery overlay routine is called in to complete processing of the error condition.

OS/3 also gives the user the option to perform his own error processing. If this option is employed, system error analysis and recovery is bypassed, and the hardware status information is returned to the task that issued the I/O request. The responsibility for processing the error remains with this task. OS/3 also supports the capability for the user to interface directly to PIOCS, if desired, to provide special handler interfaces for nonsupported devices, or to use supported devices at the physical level.

### 3.2.1.4. Transient Management

The supervisor takes full advantage of auxiliary storage to provide maximum services while keeping main storage requirements at a minimum. This is accomplished by having two types of supervisor routines:

| | |
|---|---|
|  | RESIDENT ROUTINES<br>Frequently used software modules are made a part of the supervisor program that resides in main storage at all times. |
| | TRANSIENT ROUTINES<br>Routines used only occasionally by the supervisor or by user programs are kept in the form of overlays on disk storage. These routines are referred to as transient routines. When needed, the transient routines are transferred from disk storage to main storage and activated. |

The number of areas in main storage set aside to contain transient routines is specified by each user. An important function of the supervisor is managing these transient areas by monitoring their use, controlling the loading of a requested transient routine into a selected area, and transferring control to the transient. Only one disk access is required to load a transient. Open files, close files, and terminate jobs are examples of transient routines. Some frequently used transients can be made resident at the option of the user.

### 3.2.1.5. Timer Service Management

The central processor complex contains a high-resolution timer that can provide an interrupt after any time period greater than 1 millisecond. The calling task may specify a wait interval in milliseconds or seconds, or may specify a time of day at which an interrupt is to occur.

A simulated day clock provides the time of day to tasks upon request and is used by the supervisor for time stamping messages and job accounting entries.

### 3.2.1.6. System Console Management

System console management provides for displaying messages on the system console, with responses and commands coming from the operator. The screen images are rolled upward, with new display lines or operator input appearing on the bottom of the screen. Console management routines selectively delete messages from the top of the screen that do not require responses. Console management is nonresident and is loaded as a series of transients when needed.

### 3.2.1.7. Workstation Management

Workstation management provides physical level support for the interactive services and data transfers for workstations dedicated to a job. Workstation management coordinates all the activities of the workstation with those of the requested functions to ensure efficient operation.

### 3.2.1.8. Error Recovery

Any error that causes a program interrupt is examined to determine the type of error, such as program check or protection violation, and the type of job involved.

The processing of error information by means of user-supplied subroutines is optional. Standard error control actions are initiated in the absence of user code. If an error is detected, the system is brought to an orderly halt, and a restart from this point is attempted. If recovery fails, information is collected for an orderly abnormal termination.

Invalid or inconsistent requests for supervisor functions are reported to the requesting program.

Machine check interrupts are examined by the system to determine whether the error is recoverable. If the error is not recoverable or recovery fails, the system is brought to an orderly abnormal termination. The pertinent information is collected and logged.

### 3.2.1.9. Error Logging

The error logging facility provides the capability to record hardware and software errors in the system error log file. The records placed in this file can be subsequently retrieved and statistical reports prepared.

The error logging facility is included in the system at system generation time and it is then that the type of errors to be recorded and the devices to be monitored are decided. Error logs can be collected for every device configured into the system. The type of error log records that can be collected are:

- Peripheral device errors
- Machine check errors
- Communications errors
- User specified errors

During the initial program load (IPL) of a new session, the operator is given the option of retaining the error log file from the previous session or resetting the file with either the same set of collection parameters or a new set. During the operation of the system, the operator can alter the collection of record types.

### 3.2.1.10. Multitasking

Multitasking is a programming technique that can significantly reduce the time required to process a job. Each job entered into the system consists of one or more functions, or tasks, for the system to perform. These tasks are executed one at a time in the order they appear in the job. Each task represents a step in the job's processing. Normally, each job step has only one task actively vieing for CPU usage. Through multitasking, the user can establish a hierarchy of independent tasks such that several tasks from a single job step can be active at the same time. The system takes control of the CPU away from one task when it is awaiting the completion of an external event and passes it to another task, so if that second task is from the same job, then that job gets processed faster. The separate tasks defined for each job step are called subtasks and vie for CPU usage as independent tasks along with the tasks from other jobs.

To control the tasks and subtasks vieing for the CPU concurrently, the system provides a task priority mechanism that permits the user to establish up to 60 separate task priority levels. Tasks and subtasks can be executed at separate priority levels to ensure that in a multitasking environment the proper tasks are executed in the proper order. The number of task priority levels the system supports is determined at system generation. If a task is executed without a specified priority, it is automatically assigned the lowest priority configured into the system.

Multitasking can significantly improve the processing time of an individual job, and improve the throughput of the system on a whole, by reducing the amount of idle CPU time. The system software supplied by Sperry Univac takes advantage of the multitasking facility to increase the system's efficiency.

### 3.2.1.11. Diagnostic Debugging Aids

Diagnostic debugging aids provided by the supervisor include:

**DIAGNOSTIC DEBUGGING AIDS**

**Monitor Routine**

A hardware monitor interrupt enables the monitor routine to trace the execution of a program so that errors can be located and corrected. The routine interrupts each instruction before it is executed and tests for the conditions specified in the monitor input. When a specified condition is satisfied (a specified storage location, instruction, or instruction sequence is reached), the monitor can print out current program information (PSW and register contents, next instruction to execute, etc) and can suspend program execution or continue with or without monitor intervention.

Trace conditions and information to be printed can be specified via the job control stream or entered at the system console. An entire program or a portion of a program can be monitored.

▷▷▷

# DIAGNOSTIC DEBUGGING AIDS

## Snapshot Display of Main Storage

A partial main storage printout can be obtained to aid in the resolution of a problem. The area of main storage to be displayed and the time the display is to occur are identified by parameters at run time, or specified in the body of the program.

## Main Storage Dumps

A main storage dump may be provided for diagnostic purposes under the following conditions:

- Abnormal Termination Dump for User Program

  This provides a main storage dump of a program region in hexadecimal, alphanumeric, or both, plus a formatted display of error codes, job-oriented tables, and supervisor information to assist the user in debugging.

- Program or Operator Request Dump

  This enables the operator or any program to request a main storage dump in the same format as the abnormal termination dump.

- System Failure Dump

  This routine is intended to be used when an abnormal condition occurs and other dump programs cannot be used.

## Standard System Error Message Interface

An error message service routine provides complete and specific error messages. This routine locates the message in a disk file and transfers control to the system console handler for message display.

## Error Response to User Jobs

Error codes returned by the supervisor to the calling program are standardized to provide a uniform interface for all system services.

If a user requires control returned after the detection of hardware failure or software exception, a user-supplied subroutine must be provided; otherwise, an orderly abnormal termination, which optionally may include a main storage dump, is called for the user job.

## Program Checkpoint Restart

A checkpoint request facility provides for restarting a job with synchronization of all disk and tape sequential files. If unit record files are buffered to disk or tape, the position of these files may also be restored on restart.

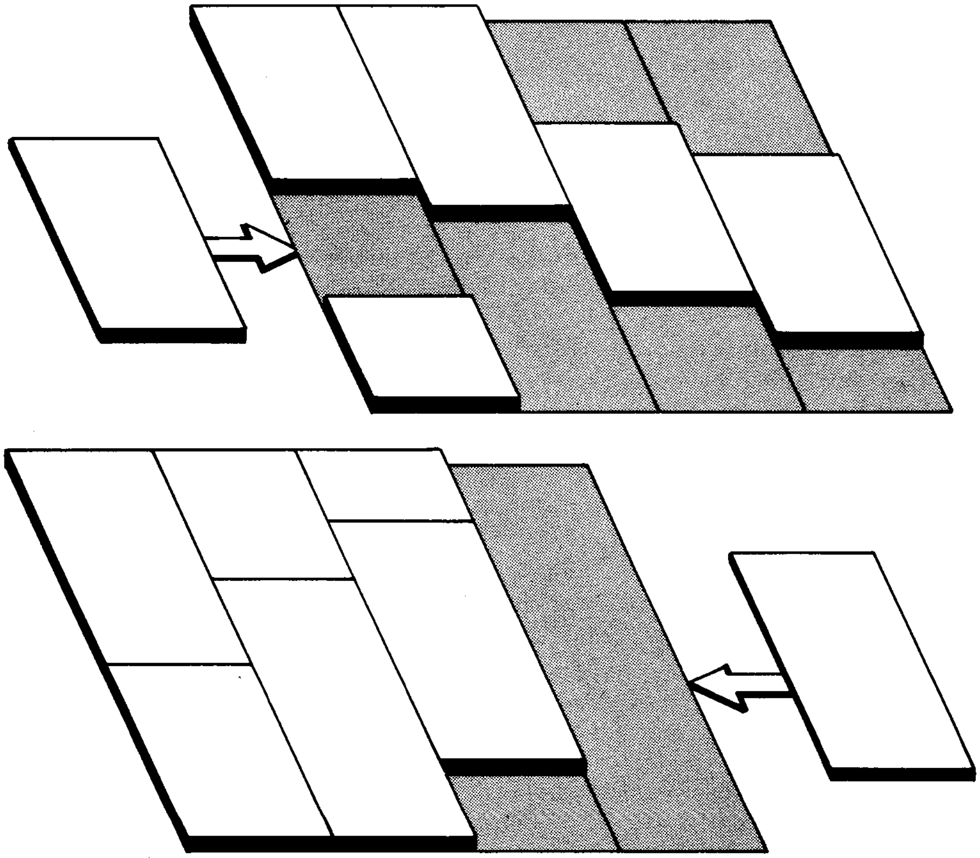
### 3.2.1.12. Main Storage Management

The supervisor acts to ensure the efficient operation and proper use of the system's main storage. Primarily, the supervisor reserves the main storage space required by jobs, programs, and system routines and loads them into their reserved areas. The supervisor makes certain that they are loaded into the proper areas and at the proper time. In addition, the supervisor can:

- dynamically expand the main storage regions of certain programs, jobs, and routines when necessary;
- rearrange main storage to provide as much contiguous space as possible; and
- temporarily suspend and remove jobs from main storage to make room for preemptive priority jobs.

The main storage management functions the supervisor performs are designed to promote system efficiency and ease of use by streamlining main storage operations and providing as much automatic storage management as possible.

THE SUPERVISOR CONSOLIDATES MAIN STORAGE SO THAT MORE JOBS CAN FIT



### 3.2.2. Job Control

Job control manages the system resources and prepares the jobs submitted for execution. A job represents a unit of work to be performed by OS/3. Each job consists of one or more job steps, each requesting the execution of a system or user program. Job control services are performed prior to the execution of the initial job step, during the transition between job steps, and at the conclusion of the job.

The services of job control are directed by the user through statements provided by the job control language (JCL). These control statements define the system resources required for proper execution of a job and facilitate the efficient management of these resources. OS/3 JCL is a flexible language that enables the user to specify the requirements for a variety of essential resources and affords a high degree of independence from limitations imposed by system configurations. Through the use of cataloged procedures, OS/3 effectively reduces the usual effort required when running frequently executed jobs.

**JOB CONTROL SERVICES**

- Automatic job scheduling and initiation
- Automatic main storage allocation
- Device assignment
- Initiation of interactive services for user input from a dedicated workstation
- Volume and file label processing
- Retrieval of cataloged control streams for subsequent modification and execution
- File cataloging
- Program restart from a checkpoint
- Query of dedicated workstation for run-time variables
- Scheduling of additional jobs from within a job
- Control stream and data storage

A control stream is a group of sequenced statements, written in the OS/3 job control language, that defines a job and directs its execution. These statements are divided into these functional groups:

- Job Coordination
- Device Assignment
- Job Step Operation
- Regulation of Job Environment

### 3.2.2.1. Job Coordination

This group of control statements provides an interface that coordinates overall job execution.

**STATEMENTS FOR JOB COORDINATION SPECIFY INFORMATION ON:**

SCHEDULING PRIORITY

Indicates one of the following priorities is to be used when scheduling a job:

- Normal priority

  Used for regular scheduling considerations within the system

- High priority

  Used for rush scheduling

- Preemptive priority

  Used for urgent jobs that require immediate scheduling and execution

Within priorities, all jobs are scheduled for execution on a first-in, first-fit basis, with main storage allocated for preemptive jobs via the rollout/rollin capability.

MAIN STORAGE REQUIREMENTS

The main storage requirements for a job can be calculated automatically by OS/3 if all programs to be executed by the job either currently reside in a load library or are specified by the user through the job control language. The user has the option of specifying a minimum and maximum value.

The minimum value is the basic storage requirement needed to properly execute all programs within the job. The maximum value specifies an additional storage requirement that, if available, could be dynamically utilized by the programs within the job to improve and speed up job execution. To exploit this additional main storage, the programs within the job must be specifically designed to take advantage of the additional main storage allocation if it becomes available to the job.

JOB IDENTIFICATION AND DELIMITATION

Uniquely identifies the job and indicates its starting and ending points.

SCHEDULING A JOB FROM WITHIN A JOB

OS/3 JCL enables the user to serially execute jobs by allowing a currently active job to request another job, which resides in the permanent library, to be executed. The requested job is processed and entered on the job queue as soon as the control stream processor is loaded. Also, a new priority level may be specified for the called job, which does not have to be the priority specified in the original job.

RESTART OF AN INTERRUPTED JOB

Job restart from a specified program checkpoint is specified through JCL by giving the required checkpoint information and resource requirements for the recurring portions of the job. Checkpoint data must be established by the user prior to requesting job control to restart a job.

MODIFICATION OF CATALOGED CONTROL STREAMS

Permanently cataloged control streams may be dynamically modified at execution time. Job control incorporates the submitted changes, as required, into the control stream retrieved from the permanent control stream library and subsequently processes the modified control stream for execution, leaving the content of the original control stream intact.

DATA MANAGEMENT SHARED MODULES

This feature defines, for the system, the specific data management shared modules required for job execution. This facility allows job control to ensure the availability of required common code modules when the user has specified a load library other than system residence.

### 3.2.2.2. Device Assignment

This group of control statements is used to declare the devices required for the proper execution of a job.

**STATEMENTS FOR DEVICE ASSIGNMENT
SPECIFY INFORMATION ON:**

DEVICE ASSIGNMENT SETS

OS/3 JCL offers a set of control statements that provide the information required to assign devices and establish the relationship between files or volumes and devices, such as the number of files per volume and the number of volumes that contain a file. The peripheral devices that may be assigned to satisfy the requirements of a job include card readers, printers, punches, disks, diskettes, and tapes.

DELETION OF FILES

This function deletes specified files from the system prior to the execution of a job step.

LOGICAL DEVICE ASSIGNMENT

This function provides the ability to temporarily change the logical unit number associated with a given device type from that specified at system generation time to one that will be used for a specific job, thus permitting the user to run existing control streams among different System 80 installations.

RELEASE OF PERIPHERAL DEVICES

This function is used to release peripheral devices that are presently assigned to a job and that are not required in subsequent job steps. Upon release, they are no longer available to the original job, and they are made available for subsequent assignment to other jobs being considered for initiation.

### 3.2.2.3. Job Step Operation

This group of control statements is used to specify job step operation. A job always consists of one or more job steps. One job step could compile a source program; another job step would link the object program, followed by a job step to execute it.

**STATEMENTS FOR JOB STEP OPERATION
SPECIFY INFORMATION ON:**

PROGRAM EXECUTION

OS/3 JCL provides the facility to specify that a system or user program be loaded for execution from a system library or an alternate user library. A job step priority can also be indicated, with the lowest value indicating highest priority.

ALTERING PRESTORED PROGRAMS

This feature of job control allows programs to be altered at execution time. The control statement gives the name of the module to be altered. After being loaded into main storage, the module is altered according to the specified changes. However, the master copy of the named module, contained in the appropriate load library, remains intact.

DATA AND PARAMETER SPECIFICATION

OS/3 JCL allows the user to include information in the control stream as embedded data or program parameters. This information is stored in the temporary control stream library for subsequent retrieval by a system or user program.

### 3.2.2.4. Regulation of Job Environment

This group of control statements is used to regulate the environment of the job by (1) making designated job steps conditional on the outcome of a previous job step or on any errors that might previously have occurred during execution, or (2) modifying the environment of the job.

**STATEMENTS FOR REGULATION OF JOB ENVIRONMENT SPECIFY INFORMATION ON:**

JOB STEP OPTIONS

OS/3 JCL allows the user to specify certain optional software functions to be performed in a job step. The specified functions are effective only in the job step in which they are included. The user can request any of the following options:

▶ Alter

Indicates that a loaded program is to be altered prior to being given control for execution.

▶ Binary Overflow

Indicates that the user program to be loaded is given control for execution with binary overflow enabled.

▶ Decimal Overflow

Indicates that the user program to be loaded is to be given control for execution with decimal overflow enabled.

▶ Load Module Construction

Indicates that the source program being compiled is to be linked and executed using default linker parameters.

▶ No Abnormal Dump

Indicates that a dump is not desired in the event of abnormal job step termination.

▶ System Dump

Indicates that a system dump is desired in place of a job dump in the event of abnormal job termination.

▶ No Volume Label Check

Indicates that header labels are not to be read and verified on disk, diskette, or tape volumes.

SKIP CONTROL STATEMENTS

This feature allows the user to bypass any number of control statements, including all statements in an executing control stream. The user can indicate a forward skip from any control statement to any other control statement in the stream. If no destination is specified, job control advances to the first statement in the next job step.

JOB-TO-WORKSTATION COMMUNICATIONS

User jobs can communicate with the workstation by including statements that instruct the job to display queries on the workstation screen during the running of the job. The user has the option of altering the processing of the job at the point where the message was displayed.

JOB-TO-OPERATOR COMMUNICATIONS

This feature enables the user to communicate with the operator. The user can place a message of up to 60 characters anywhere in the control stream, and the message will be displayed on the screen of the system console during control stream processing.

DATE CHANGE

This allows the user to alter or modify the calendar date for a specific job by submitting a 6-character date consisting of the month, day, and year in any order, or a 5-character date consisting of two characters for the year and three characters for the day.

▷▷▷

**STATEMENTS FOR REGULATION OF JOB ENVIRONMENT SPECIFY INFORMATION ON:**

TRACE MODE

OS/3 JCL provides a trace option for use in program debugging. When the trace mode is requested, every instruction in the job step is examined before execution without interfering with the normal operation of the program, and the pertinent operational information for every traced instruction is printed by the system.

MAGNETIC TAPE POSITIONING

This feature allows the positioning of tape volumes prior to the execution of a job step. It can be used to position a data file or pre-position a multifile tape volume. It will space the tape volume forward or backward a specified number of tape marks or blocks, rewind the tape volume, or write a tape mark.

### 3.2.2.5. Dialog for Job Control Stream and Jproc Preparation

The job control dialog, supplied by Sperry Univac, is a product that leads the user step by step through the process of building a job control stream or a jproc from a workstation. It allows the user to choose the statements and system jprocs required from lists of menu items and requests parametric values where necessary. It also prompts with HELP screens, which when requested by the user explain the choices available. The dialog guides even inexperienced personnel in building valid job control streams. The job control dialog automatically stores the control streams and jprocs in either the system job control library file ($Y$JCS) or an alternate library file identified by the user.

### 3.2.2.6. File Cataloging

The file cataloging facility provides a method to control file usage and to limit the use of selected files to only authorized individuals. File cataloging also provides the capability to build and maintain generation files. In addition, file cataloging can be used as a convenience facility because it reduces the job control required to access a file.

To use the cataloging facility, the device assignment sets of the files to be cataloged are placed in the system catalog file. A device assignment set is the job control statements that define a file and its location. To place a device assignment set into the catalog, a cataloging statement is included in the device assignment set.

If the use of a cataloged file is to be restricted, protective passwords can be assigned to the file. A read password can be assigned to the file, and only those knowing the password can access the file. A write password can also be assigned, and only those knowing the password can make changes to the file.

Often it is desirable to alter a file yet retain a backup copy of the file as it appeared before alteration. Files maintained in this manner are called generation files. The catalog facility provides the capability to build and maintain generation files.

A catalog manipulation utility is provided for use by the system administrator to perform certain maintenance functions on the catalog. This utility can be used to:

- assign a protective password to the catalog file to prevent its unauthorized access;
- make a backup copy of the file and restore the copy if necessary; and
- obtain a listing of the contents of the file.

### 3.2.3. Interactive Processors

Three interactive processors support the workstation-to-job command interface:

- Workstation command processors
- Dialog processor
- Screen format coordinator

### 3.2.3.1. Workstation Command Processors

The workstation command processors provide the interface between the workstation and the various functional components of the OS/3 software. The command processors handle:

- commands requesting supervisor control system functions, such as logging on, job execution, and initiation of the interactive facilities;
- commands issued to control the processing of jobs; and
- commands issued to the various interactive facilities.

### 3.2.3.2. Dialog Processor

The dialog processor supports the dialog processing services provided by System 80. Dialogs are used to enter variable data directly to a processing program through the workstation. The dialog processor coordinates the displaying of prefiled dialogs, extracts the data entered in response to the dialog, and routes the data to the appropriate user program for processing. The dialog processor functions in response to commands issued from within the processing program that is to make use of the information. When a program issues a request to the dialog processing services, the dialog processor locates and displays the dialog specified by the program. Several system functions use dialogs to make the system easier to use. The job control and system generation dialogs are two examples of such system functions.

The dialog processor maintains an audit file to save entries made to a dialog. The next time that dialog is encountered, the dialog processor can be instructed to read the entries from the audit file. The audit file can be used to make entries to all, or selected portions, of the dialog. The user supplies the information in the normal fashion for those parts of the dialog not filled in by the audit file. The audit file can be amended by the user, and a printed listing of the file, called an audit file summary, is presented each time it is created or altered.

### 3.2.3.3. Screen Format Coordinator

The screen format coordinator is responsible for interfacing the screen formats with the program and workstation users that use them. Screen formats are simply forms displayed on the workstation screen. Screen formats can be used by a program to output data to a workstation user or to input data from a workstation user. When a screen request is encountered, the screen format coordinator locates the requested format and then either displays it with the appropriate output data or displays it and waits for data to be input by the user. These functions are performed automatically in response to the screen format request commands embedded in the programs using the formats.

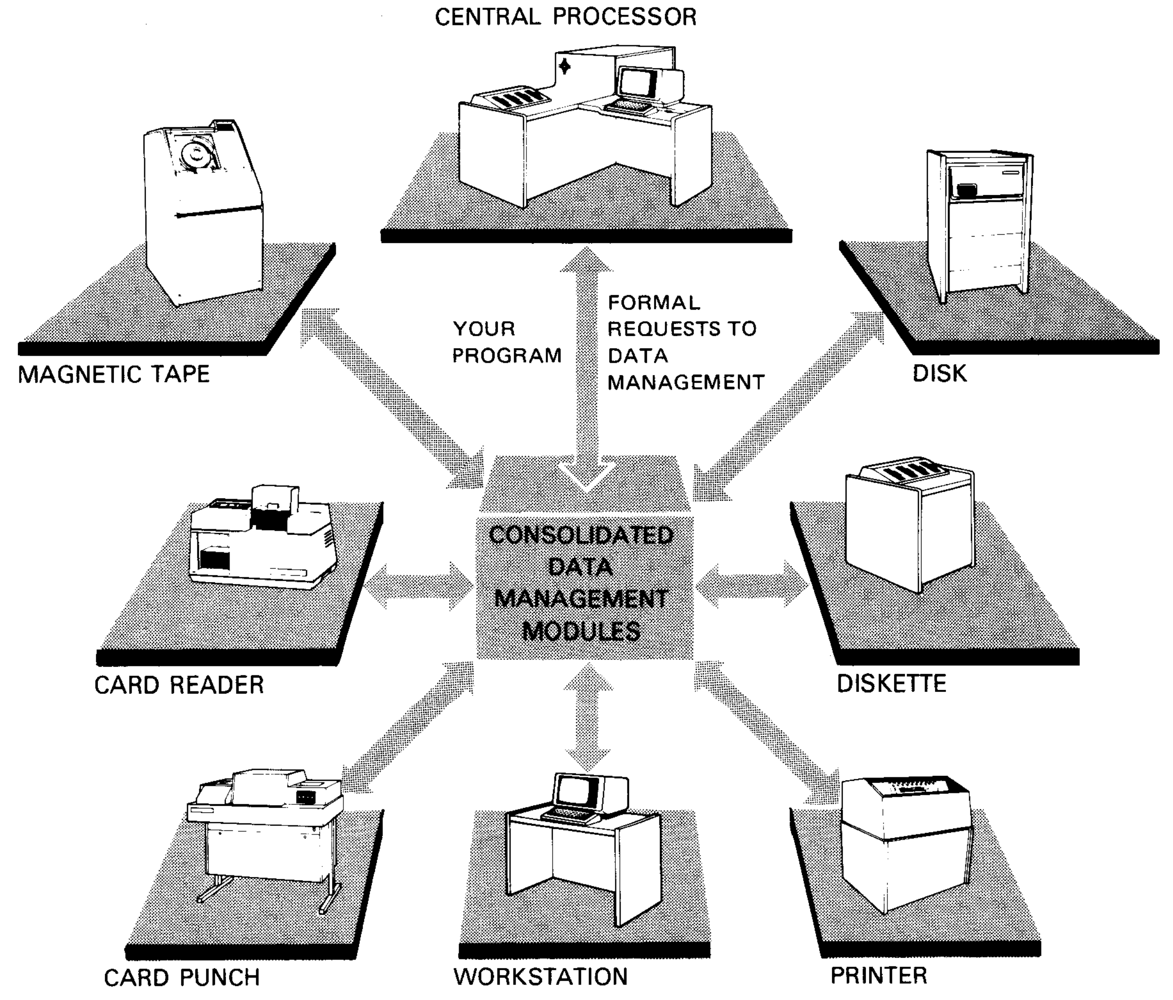
### 3.2.4. Consolidated Data Management

The consolidated data management system acts as a convenient intermediary between user programs and the input/output facilities of the supervisor and assists in the task of accessing data files on various peripheral devices. Data management relieves the user of the necessity of coding the routines for block/deblocking, buffering, and communication with the physical input/output control system. Consolidated data management provides an object code level interface for all higher level languages and a macroinstruction level interface for BAL users. The higher level language interface is transparent to the user.

The consolidated data management system offers the following features:

- A single access method for disk files
- Shared data management modules
- Device independence
- Support of the interactive features and workstation data transfers

DATA MANAGEMENT HANDLES
THE MOVEMENT OF DATA BETWEEN DEVICES



### 3.2.4.1. Logical Input/Output Control System

The logical input/output control system (IOCS) modules that control each access method are sharable subroutines and are capable of supporting a single or multiple program environment. When referenced within a user program, these subroutines are dynamically loaded into main storage and can then be shared by many user programs.

### 3.2.4.2. Disk Access Method

The OS/3 consolidated data management access method for disk files is the multiple indexed random access method (MIRAM). MIRAM simplifies the data management requirements for disk files by offering a single access method that provides several ways of accessing a disk file. The records of a MIRAM file can be accessed:

- sequentially in order of placement;
- sequentially by ascending key;
- randomly by multiple keys; or
- randomly by relative record number.

The use of MIRAM conserves main storage space by providing a single data management module for disk access that can be shared by every user program, rather than having several data management modules, one for each access method.

### 3.2.4.3. Diskette Access Method

Consolidated data management provides for accessing the records of a diskette file:

- sequentially in order of placement;
- randomly by relative record number; or
- by data set labels.

### 3.2.4.4. Workstation Access Method

The workstation access method provides the logical level interface between the workstation and the system and user software. The workstation access method provides:

- Automatic support for the workstation as an interactive programming device, including the support of the interactive features that cause screen displays and the management of the screen displays
- Transferring application data and application messages between the workstation and user job
- Console-like capability between the user program, system console operator, and the workstation operators
- Support of workstation function key capabilities
- Device independence

These capabilities are provided automatically or may be controlled through a set of declarative and imperative macroinstructions that provide for accessing workstations at a logical level through the common data interface.

### 3.2.4.5. Magnetic Tape and Unit Record Device Access Methods

Magnetic tape subsystems, card readers, card punches, and printers are devices on which I/O operations are performed in sequential order; that is, records are handled from the first to last according to physical placement.

For these four device types, OS/3 consolidated data management provides sequential access method (SAM) modules to handle the access requests of user programs.

### 3.2.4.6. Device Independence

The consolidated data management system provides a significant level of device independence by providing a single system control table format to be used to control and define all file types (disk, diskette, printer, etc) instead of a unique format for each. This allows the files in a program to be changed from one type to another (e.g., disk to diskette) without requiring the user to alter the source code of the program that accesses that file. The only changes required would be to the job control statements that associate the file with a device. Of course, device-unique commands occurring in source programs, such as tape position commands, would need to be removed or altered accordingly. Thus, the OS/3 consolidated data management provides device independence by allowing logical data transfers among devices having dissimilar physical characteristics.

## 3.2.5. System Service Programs

### 3.2.5.1. System Librarians

System and user program libraries can be generated as either system access technique (SAT) files or as multiple indexed random access (MIRAM) files. To provide maintenance capabilities to the user for these program libraries, OS/3 includes two program librarians: the SAT librarian, and the MIRAM librarian.

The OS/3 SAT librarian is a set of integrated subroutines existing as a separate system entity and functioning as a housekeeping and maintenance tool for the system and user libraries generated as system access technique files. Programs and elements such as language processor source modules, language processor output (relocatable object modules), or system-executable load modules reside within a library (which is either a system or private file in the OS/3 environment). The storage, collection, creation, modification, correction, deletion, addition, duplication, and transposition of library modules are the primary tasks of the librarian. These housekeeping functions may be performed on entire program libraries, groups of program elements within a specified library, or individual program elements.

The librarian also transfers program library modules from one medium to another. While library modules reside primarily within disk files, the librarian is capable of transposing program libraries to or from a card, diskette, or tape medium.

Control statements supplied to the librarian via the control stream are the means by which the user directs the performance of the operations desired. Through the use of control statements, the librarian initiates such functions as copy, delete, add, compare, merge, compress, correct, group, rename, sequence, reproduce, list, and punch.

The librarian may modify existing libraries, create new libraries via copy or merge operations, or duplicate libraries in their entirety. The librarian also ensures that a given library does not contain any modules with identical names and types.

The listing and punching capabilities of the librarian allow a user to examine the contents of an individual program module or groups of program modules. A module may be listed by the printer as well as punched into cards. The user also may obtain a table-of-contents-type listing of all the program modules in a file. A correction facility allows modification and updating within program modules existing in a library. Source modules may be corrected or updated, as well as sequenced. When any program module is processed by the librarian, its format is verified.

At the option of the user, the librarian provides a map of the functions requested and the results of the action taken, along with diagnostic error indications and warnings, as appropriate. The extent and amount of information provided on the librarian control statements coded are dependent upon the options specified by the user. At the discretion of the user, the control statements themselves may be a part of the library map.

The MIRAM librarian is used to perform maintenance functions on program libraries generated as MIRAM files. System libraries generated as MIRAM files include the system screen format library and the saved, expanded run library. The MIRAM librarian can be used to:

- Copy all or selected modules from one library to another
- Delete selected modules from a library
- Print all or selected modules in a library
- Print a library directory consisting of all active header records
- Insert comments into a module header record
- Change the name or type of a module

The MIRAM librarian responds to a set of user control statements inserted into the control stream that executed the librarian. The librarian will produce a printed listing of all operations performed during a given librarian session. This listing is called a librarian map.

### 3.2.5.2. Linkage Editor

All OS/3 language processors produce object modules as the output of their compilation processes. These elements may subsequently be structured into a user-tailored executable program by the OS/3 linkage editor. Object modules from various language processor compilations can be structured by the linkage editor into a single loadable program embodying the segment and overlay characteristics described by linkage editor control statements. Additionally, the object modules thus collected may contain cross-references to each other for specific purposes of program execution and communication. Such cross-references between separately compiled modules are resolved by the linkage editor when these elements are collected and a loadable program is constructed.

The loadable program the linkage editor builds can consist of program segments fashioned into multiple regions. Each of these regions is structured as a hierarchical tree with specific boundaries inherently defined as branches of a tree trunk, or root segment. This structure permits a user to produce a program that is larger than the main storage area assigned for its execution. The various segments of a multiphase load module are loaded and executed as required by the logic of the program.

Cross-references made by the user (within compiled object modules) and resolved by the linkage editor in forming the loadable program may exist within or across segments. Segments are composed of one or more object modules, and the loadable program is composed of one or more segments, the initial segment always being designated as the root phase segment. The user's tools for establishing cross-references between processor object modules are:

- Language processor external reference declarations (EXTRN) used to create indirect references in requesting object modules.

- External definition declarations (ENTRY) used to create indirect definitions in satisfying object modules.

These declarations are used by the linkage editor to resolve cross-references between two or more object modules that are linked together at link-edit time. Executable program generation is, therefore, essentially a two-fold process:

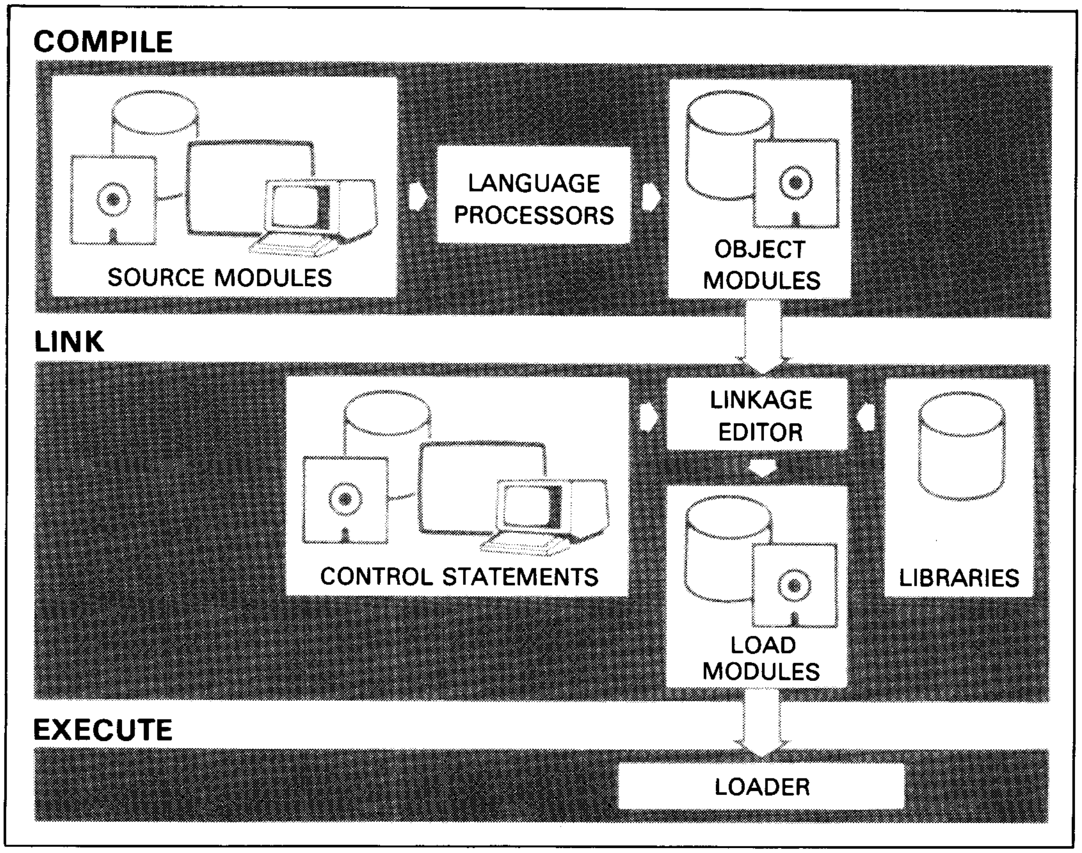
- The source program code must be compiled by the various language processors to produce one or more object modules.

- The various object modules must be combined by the linkage editor to create a single, executable load module.

At link-edit time, the linkage editor assigns, to the object modules included in the load module, a new relative address based upon their new relative position within the load module. Any cross-references that exist between object modules are satisfied by replacing them with the relative address of their respective definitions or references, as the case may be.

The output of the linkage editor is, therefore, a loadable executable program acceptable to the OS/3 loader.

The role of the linkage editor in program preparation is as follows:



The ability of the linkage editor to construct a single executable load module from several object modules has the following major advantages:

- If a change is required in one of the included object modules, only that object module must be compiled or assembled again.

- The various source modules may be written in the appropriate language and combined into a single executable program.

- Routines common to two or more object modules need be assembled or compiled only once and the resulting object code linked as needed; the result is reduction in the total time required to generate an executable program.

In addition to the basic linking function, the linkage editor performs the following:

- Searches an appropriate library and, either on request or automatically, incorporates object elements other than those in its primary input

- Performs program modification by deleting and rearranging control sections of an object module as directed

- Produces an optional overlay structure to be used by the supervisor during loading

- Reserves space automatically for common storage requests generated by the language processor

### 3.2.5.3. Disk, Diskette, and Tape Initialization Routines

The various tape, disk, and diskette initialization, or prep, routines check the condition of the magnetic storage media and prepare them for use by the system. The disk and diskette prep routines respond to a set of keyword parameters inserted into the job control stream while the tape prep routine is executed with no special parameters required. Associated with the disk prep routine is a utility to automatically assign alternate tracks to replace each defective track identified by the disk prep routine.

### 3.2.5.4. Dump Routines

The system has a variety of dump routines to aid in the diagnosis of subtle hardware and software errors. The routines include a system dump routine and user job and program dump routines.

### DUMP ROUTINES

**SYSTEM DUMP ROUTINE**

The system dump routine provides a printout of all, or selected portions, of the contents of main storage. The routine can be initiated unconditionally by the user or can be preset to automatically begin if a system failure occurs. The dump obtained is the same in either case.

The system dump listing is divided into several parts, each part corresponding to a system component. The sections are clearly labeled with the appropriate heading for ease of use. A typical system dump listing contains:

- The contents of low order storage
- The physical unit, system information, and channel control blocks
- The system switch list
- Translated job region
- The supervisor
- Hexadecimal job region
- Free region

The dump listing can be in hexadecimal, decimal, or EBCDIC format.

**USER DUMP ROUTINE**

It is possible for a job to terminate abnormally, or crash, with no apparent reason. Often the only way to determine what caused the job to crash is to list all the information in the system relating to the job. Such a listing is called a job dump, and two routines are available for obtaining a job dump. They are: the job dump routine, and the user end-of-job (EOJ) dump. Both routines produce a listing containing the same type of information, but the formats are different.

## 3.2.5.5. Software Maintenance Packages and Changes

Software maintenance packages and corrections provide maintenance changes to System 80 software that enhance system stability or forestall potential difficulties. Software maintenance packages are issued on a regular basis, while software maintenance changes are provided to meet individual requirements.

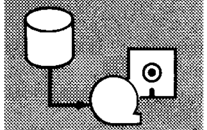
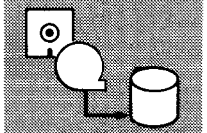
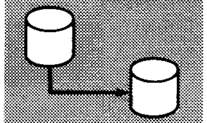
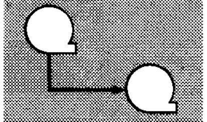
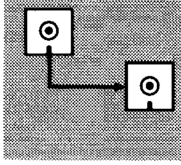
Software maintenance packages and changes are issued on diskettes. They are installed interactively and perform all functions necessary to apply and test the changes. Instructions on application and a list of contents of each package or change are included on the diskette. The user must execute the interactive software maintenance change list routine to obtain a copy of application instructions and contents.

### 3.2.5.6. Catalog Manipulation Utility

The catalog manipulation utility is used to maintain the system catalog file, which holds the device assignment sets for often-used user files. A file whose device assignment set has been placed in the system catalog file can be accessed thereafter through one job control statement. The catalog manipulation utility enables the system administrator to assign a protective password to the catalog file, make backup copies of the catalog file, and restore these copies to disk when required. The utility operates in response to a set of command statements embedded in the job control stream that executes the utility.

### 3.2.5.7. Disk Dump/Restore

The disk dump/restore utility allows the user to make backup copies of the contents of a disk volume. Using the dump/restore utility, the user can perform these operations:

| | |
|---|---|
| Dump<br> | Copy all of, or any part of, a disk to a magnetic tape or diskette |
| Restore<br> | Copy a magnetic tape or diskette to a disk |
| Disk Copy<br> | Copy all of, or any part of, a disk to another disk |
| Tape Copy<br> | Copy a tape created by a previous dump/restore operation to another tape |
| Diskette Copy<br> | Copy a diskette created by a previous dump/restore operation to another diskette |

### 3.2.5.8. System Patch Routine

The system patch routine is provided to allow users and Sperry Univac personnel to correct transient routines and overlays.

### 3.2.6. System Installation Facilities

System installation is the essential process of installing the SPERRY UNIVAC System 80 hardware, integrating into it the Operating System/3 (OS/3) software, and generating this software so that it fits the user's particular needs. Once hardware installation is complete, the user can install and generate the OS/3 software.

#### 3.2.6.1. Software Installation Facilities

Sperry Univac delivers OS/3 software on a set of release diskettes, on tapes, or on disk. Software installation involves the transfer of this delivered software to the nonremovable disk pack integrated in the System 80 processor complex. Once the delivered software has been copied onto the integrated disk, the disk is referred to as the system resident volume, or SYSRES. The SYSRES disk holds all system software and must remain on line when the system is in operation.

To accomplish software installation, Sperry Univac provides installation routines as part of the standard OS/3 release. These routines provide the user with the capability to install:

- The initial release of OS/3 software
- Any new software received between major releases
- Updated software as Sperry Univac releases major improvements to OS/3
- Software maintenance packages and changes

#### 3.2.6.2. System Generation Facilities

System generation, or SYSGEN, is the process whereby the user defines the system's hardware configuration to OS/3 and generates, or creates, the control elements required to meet particular processing requirements. This responsibility is placed in the user's hands because it is the user who is most familiar with his or her own requirements.

To help the user accomplish system generation and to simplify the procedure, Sperry Univac provides these SYSGEN facilities:

- Dialog
- Parameter processor
- Job control streams

The SYSGEN dialog is an easy-to-use facility that helps the user prepare and process required SYSGEN parameters directly through the workstation. The user makes SYSGEN parameter selections in response to queries displayed on the workstation screen. For omitted or incorrectly specified parameters, OS/3 provides default values that ensure that a usable system is generated. The dialog accepts user choices and passes them to the SYSGEN parameter processor. The parameter processor, in turn, validates the submitted parameter selections and generates a series of job control streams based upon the selections and lists these streams for the user. The user then executes the actual system generation by interactively running the generated control streams through simple workstation keyins.

### 3.2.6.3. Installation Verification Programs

Sperry Univac supplies a set of installation verification programs that can be run after SYSGEN is complete to test the functional capability and operation of the various components included on the resident disk unit.

## 3.3. EXTENDED SYSTEM SOFTWARE

The extended system software provides a number of optional features and improves the utilization and operation of the system. The extended system software is available as a package, including all these components:

**OS/3 EXTENDED SOFTWARE**

- Screen format generator
- Dialog specification language translator
- Data utilities
- SORT/MERGE*
- SORT3*
- Spooling and job accounting/reporting facility

**Available as separate items*

### 3.3.1. Screen Format Generator

The screen format generator provides the user with the capability to interactively create template-like formats for display on workstation screens. These formats can be used to simplify entering variable data and parameters and to present output data on the workstation screen in a formatted manner.

Users can generate screen formats to meet particular requirements and applications. The screen format generator permits the user to exercise great control over the appearance of the screen format. The user can include as many fields within the screen as he chooses and specify the characteristics of each field. Users can specify the length of the field; editing attributes; whether the field is input, output, or both; and the disposition of the screen after use.

The screen format generator includes extensive prompting features to assist the user in the generation or modification of formats. Formats generated are stored in a screen format file where they may be accessed by the programs that need to use them or by workstation users for further modification. Formats are accessed through user-defined format names. The screen format generator also provides error detection capabilities during the creation and modification of screen formats.

Screen formats often prove to be the most efficient and accurate way of entering variable data into the system. If properly constructed, they can provide a way for inexperienced users to access the system.

### 3.3.2. Dialog Specification Language Translator

The dialog specification language translator allows the user to create his own job-oriented dialogs. It accepts dialog specification language source code as input and uses that input to generate a dialog display on a workstation screen. Dialogs consist of a series of queries to which the user responds with the appropriate information. Users can use the dialog specification language to produce dialogs to meet their requirements. The user assigns a name to each dialog generated through the dialog specification language translator, and this name is used to access the dialog. Each language has commands to access a dialog. Once accessed, the dialog is displayed on a specified workstation screen, and the information entered in response to the dialog is passed to the requesting program for processing.

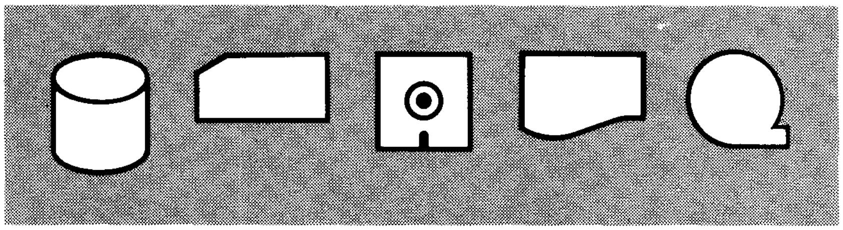
### 3.3.3. Data Utilities

The data utilities (DATA) program performs a number of essential data file maintenance and manipulation functions. Through the data utilities program, the user can:

- Make a copy of a data file on the same type of storage device or a different type
- Reformat the records of a data file so that it can be used in another program
- Select or delete specific areas of a file for testing or report preparation
- Produce a printed listing of a file in a variety of formats
- Compare duplicated files to their originals to check for discrepancies

The DATA program can process files stored on any of the supported storage media: disk, diskette, tape, and cards. Files can be transferred from one medium to any other medium, and printed listings can be made from files on any medium. Users can also perform such device-related functions as label checking, tape rewinds, sequence checking, and write protected files.

**DATA UTILITIES TRANSFER DATA FROM ONE MEDIUM TO ANOTHER**



The DATA program can be executed as an interactive program or run as a batch job. If executed as an interactive program, the user initiates the DATA program through a simple keyin command. Control specifications for the program are entered in response to a dialog diaplayed on the workstation screen. Once the user has traversed the dialog, providing all appropriate information, the DATA program is executed and performs the specified function.

If executing the DATA program through job control, the user must prepare a job control stream. The appropriate DATA program control statements are included in the control stream. The DATA program has the same capabilities regardless of how it is executed.

### 3.3.4. SORT/MERGE

The OS/3 SORT/MERGE can operate in the following manner.

**SORT/MERGE**

INDEPENDENT

As an independent sort/merge program defined and initiated by job control, SORT/MERGE is constructed as a processor that consists of interrelated modules operating within the framework of a system-driver program. Each module is designed to perform a specific function. As SORT/MERGE progresses through the execution phases, the modules required are called by the driver program, loaded into main storage, and executed.

SUBROUTINE

As a modular subroutine integrated into a user program, SORT/MERGE can be integrated into a user program to allow the user flexibility and freedom, with respect to the external format and source of input records, and to the external format and disposition of the output records. SORT/MERGE can be called from a language processor program that includes the verb specifying this operation. The modular structure gives the user efficient operation despite variations in hardware configurations and data requirements. The modular structure allows a module to be changed, replaced, deleted, or added.

**CAPABILITIES**

- An interface that permits disk or magnetic tape to be used as work areas
- Input and output that may be associated independently with disk, diskette, or magnetic tape
- Sorting of blocked or unblocked records
- Sorting of fixed-length or variable-length records
- Handling of seven types of key field formats:
  - Character
  - Binary (signed or unsigned)
  - EBCDIC data in ASCII collating sequence
  - Decimal (signed zoned and unsigned zoned)
  - Leading and trailing sign numeric
  - Overpunched leading and trailing sign numeric
  - Floating point (single and double precision)
- Specification of up to 255 key fields
- Sorting of noncontiguous key fields in ascending or descending sequence
- User specification of alternate collation sequence
- Execution of input and output own code
- Sorting of two or more different characters with the same collating value (multiple character sort)
- Shared input and reserved output devices
- A merge-only capability
- Performing of data validity and data integrity checks during sorting
- Convenient restart procedure

### 3.3.5. System/3 Compatible Sort

Sperry Univac also supplies an IBM System/3 compatible sort (SORT3). SORT3 is an OS/3 program initiated by means of job control. This sort can process disk files, diskette files, tape files, and card input and:

- Rearrange the records in a file
- Select specific records from a file
- Reformat the records in a file
- Summarize fields in the records

SORT3 is capable of performing three different types of sorts: full record sort, tag sort, and summary sort.

The output from the full record sort is 10-byte (binary) relative record numbers of the records in the input file. The tag sort output is a file of sorted records that can contain control fields and data, control fields only, or data only. The output of the summary sort can be any of the following:

- Control fields, data fields, and summary data
- Control fields only
- Data fields only
- Data fields and summary data
- Summary data fields only
- Control fields and summary data fields

### 3.3.6. Spooling and Job Accounting/Reporting

Spooling (Simultaneous Peripheral Operations On Line) is a technique that increases the throughput of System 80. Data from low-speed peripheral devices is transferred to disk storage independently of the program that will use the data. When a user program is logically retrieving data from a low-speed peripheral, it is physically retrieving the data from a higher speed device. On output, the user program logically specifies a low-speed device, but the images are physically recorded on disk storage and later, under system control, transferred to a low-speed device. Spooling of output allows concurrent use of a specified device by multiple programs. OS/3 spooling consists of these routines:

- Input Readers

  Input readers accept data files from local or remote subsystems. When these data files are used, no distinction is made as to the method of submission.

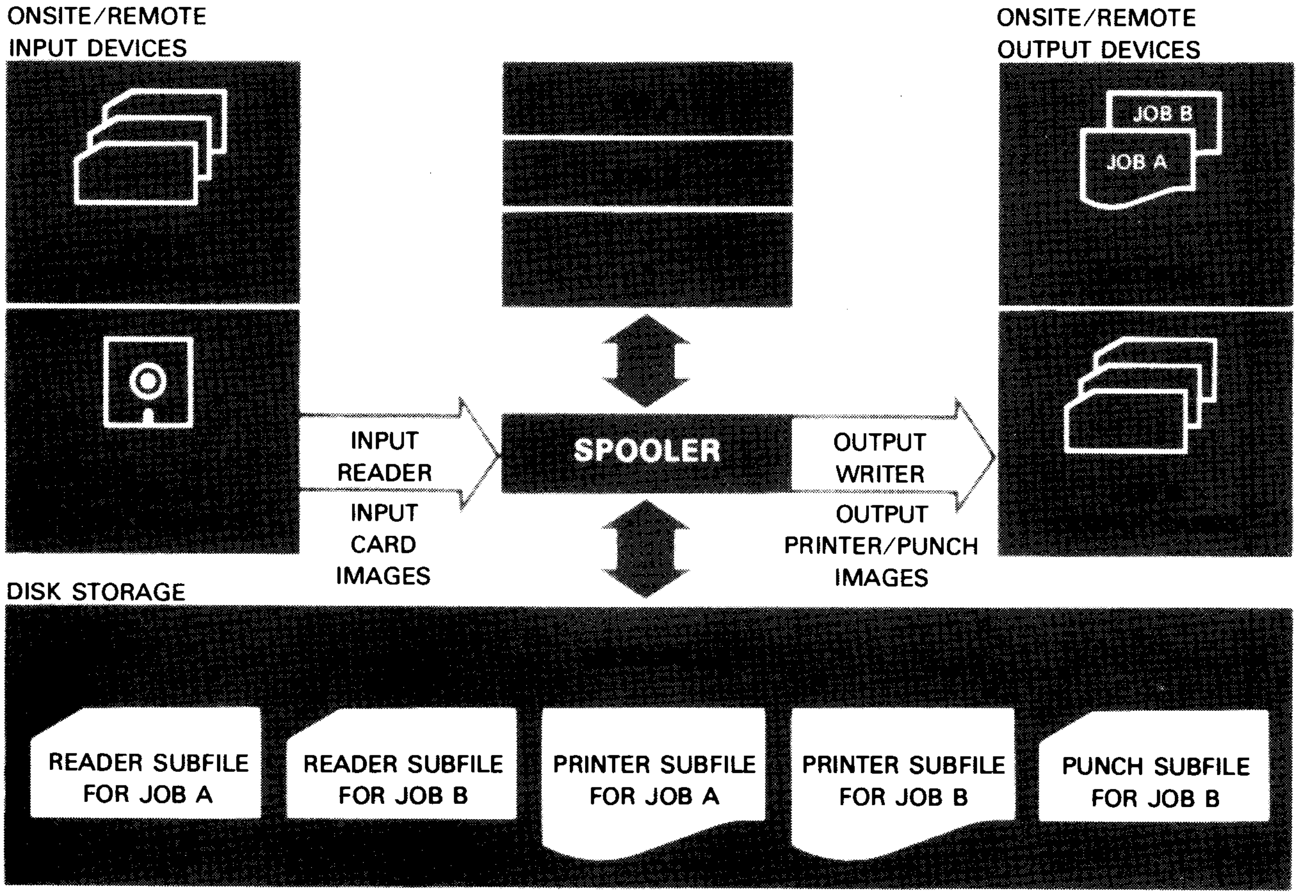
- Spooler Cooperative

  The spooler routine stores card data in the spool file until it is required by a user program. Data output from a user program is stored by the spooler in the spool file until a printer or punch is available.

- Output Writers

  Output writers provide local or remote users with output from user programs that receive identical service regardless of final destination. In addition, the operator of the printer can control forms recovery at page increments, control the number of copies, and control starting and stopping.

Here is how several active jobs use the concurrent spooling capability of OS/3:



As a normal part of spooling operation, job accounting information is generated for each job run on the system. This information is included in the normal message output log, but the information is not saved. However, the user can specify that all the accounting data generated for each job be saved in a spooling LOG file.

The system console and job log accounting utilities gather selected information from the system spool LOG file, sort the information according to user specifications, and produce a printed report. The system console and job log accounting utilities are:

- System log accumulation utility

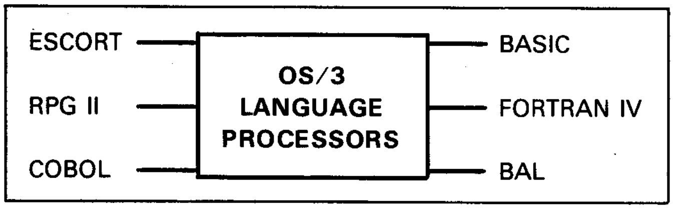
- Joblog report program

The system log accumulation utility transfers all of, or selected portions of, the spooling LOG file onto a disk or tape file for use as input to user accounting routines. The system log accumulation routine is initiated by running a canned job control stream.

Once created, the file can be used by the joblog report program to create a printed report containing selected job accounting information. This program gathers the information and sorts it according to user specifications. The joblog report program is initiated by running a canned job control stream. Parameters allow the user to select the type of accounting records to be included in the report and the manner in which they are to be sorted.

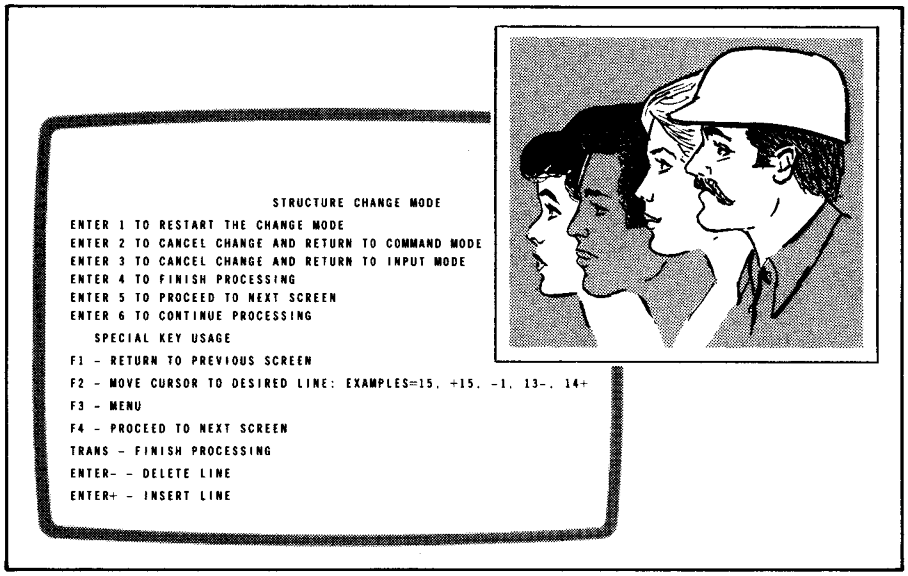
## 3.4. LANGUAGE PROCESSORS

Six language processors allow flexibility in preparing programs for use with OS/3:



### 3.4.1. ESCORT

ESCORT is a high-level, interactive programming language used to generate file processing and data handling programs. It is an easy-to-use language requiring a minimum of training yet having the power of more complex data handling languages or routines. The ESCORT language employs English-like verbs and clauses to express programming operations, thus permitting any programming novice to quickly learn how to express programming solutions. Here is a typical ESCORT screen display.



```
                 STRUCTURE CHANGE MODE
ENTER 1 TO RESTART THE CHANGE MODE
ENTER 2 TO CANCEL CHANGE AND RETURN TO COMMAND MODE
ENTER 3 TO CANCEL CHANGE AND RETURN TO INPUT MODE
ENTER 4 TO FINISH PROCESSING
ENTER 5 TO PROCEED TO NEXT SCREEN
ENTER 6 TO CONTINUE PROCESSING
   SPECIAL KEY USAGE
F1 - RETURN TO PREVIOUS SCREEN
F2 - MOVE CURSOR TO DESIRED LINE: EXAMPLES=15, +15, -1, 13-, 14+
F3 - MENU
F4 - PROCEED TO NEXT SCREEN
TRANS - FINISH PROCESSING
ENTER- - DELETE LINE
ENTER+ - INSERT LINE
```

ESCORT can be used to develop programs for report generation, data entry and retrieval, and transaction processing. An ESCORT program can: create, sort, or merge files; add, delete, sort, or replace records within a file; issue prompting messages to a workstation for input data; and validate the input data. Computational operations available include addition, subtraction, multiplication, and division.

ESCORT programming sessions are conducted directly at the workstation, and the ESCORT compiler employs prompting messages for data entry. The amount of prompting used depends on the level of programming expertise of the user. The user can conduct a session in the program mode, which uses a minimum of prompting, or when necessary, the session can be conducted in the tutorial mode with extensive prompting and explanatory screen displays. The compiler offers immediate syntax checking of ESCORT statements and sends messages when more extensive errors are detected. These errors can be corrected immediately through the workstation.

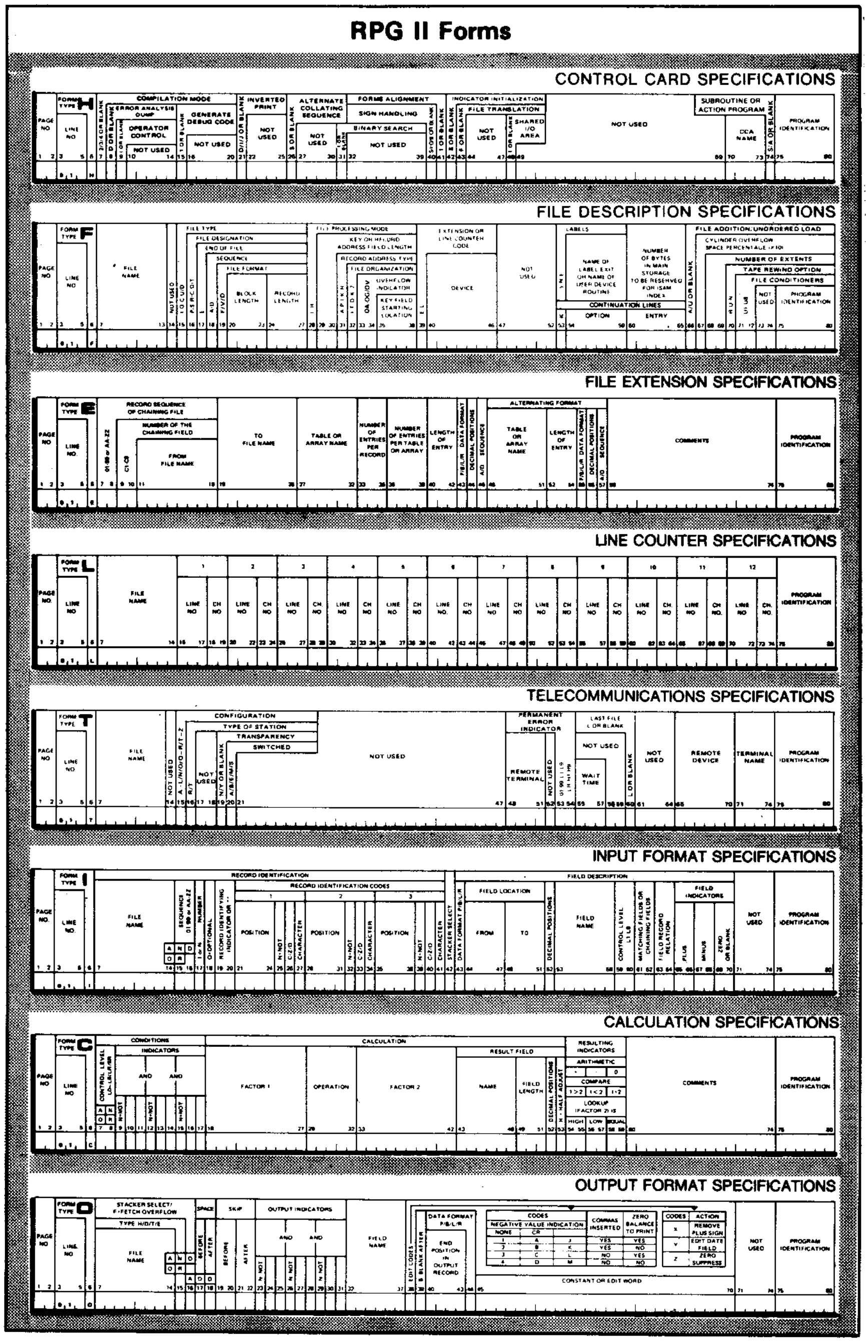
ESCORT employs the common data interface for data transfers, and all data files are MIRAM files. In addition, field names and attributes for the records within a file are contained in control data for the file. Thus, the programmer is freed from knowing detailed data descriptions for existing files and is able to concentrate more fully on how to express the actions to be taken to solve his problem.

## 3.4.2. RPG II

The OS/3 report program generator (RPG II) is a processing language that generates an object program from a series of interrelated specifications. RPG II provides the basic logic of the program with the RPG II cycle. Through the various RPG II specifications forms, the user can define records and indicate the operations to be performed on these records. Each record is then processed by the same RPG II program cycle, and the user-specified operations are performed, provided the conditions specified for the operation are satisfied.

To define a report program for generation, the requirements are listed by the programmer on specifications forms.

RPG II Forms


CONTROL CARD SPECIFICATIONS
FILE DESCRIPTION SPECIFICATIONS
FILE EXTENSION SPECIFICATIONS
LINE COUNTER SPECIFICATIONS
TELECOMMUNICATIONS SPECIFICATIONS
INPUT FORMAT SPECIFICATIONS
CALCULATION SPECIFICATIONS
OUTPUT FORMAT SPECIFICATIONS

The information coded on these forms becomes the actual input to the system. The compiler generates an object module for input into the linkage editor; the subsequent output is a loadable program module. The system can be instructed to display the RPG II formats on the workstation screen and the user can file the proper entries. Users may also enter unformatted RPG II source statements. Enhancements have been included in RPG II that enable the user to develop more sophisticated report programs within the existing framework of the language. These enhancements are:

- Telecommunications interfaces are supplied via an RPG II specifications form offering extensive communications capabilities to the RPG II user.
- Eight control-stream-time user switches (indicators) that may be used to condition calculations, input files, output files, or specific output records.
- IMS action programs can be written in RPG II.
- Workstations are accessible through programmed operations.

Sperry Univac offers two facilities to simplify RPG II source program entry:

- RPG II editor
- Auto report facility

The RPG II editor is an extension of the general editor and is used specifically to create and maintain RPG II source programs. The RPG II editor is initiated at the workstation and provides a number of ways for the user to enter and update the source programs. The experienced RPG II user can indicate at the start of the session that he will be entering the source statements in a free-form format. The less experienced programmer may find it more advantageous to use the positional format. In this format, the columns that must be filled in are displayed. The novice user can take advantage of the specification format display. If this display is used, the standard RPG II specifications format is displayed along with detailed explanations of their uses and all possible entries. The source programs generated can be stored in the program library that the user chooses and can be redisplayed in either of the formats for updating.

The RPG II auto report facility accepts simplified specifications and standard RPG II source statements and generates a complete RPG II source program. By simplifying programming, auto report increases programmer productivity.

### 3.4.3. COBOL

Common Business Oriented Language (COBOL) is a programming language oriented toward problems in business applications. The language is similar to the English language, rather than a notation that considers the technical aspects of a particular data processing system. The source programs are easily transferable among systems that accept *American National Standard COBOL X3.23-1974.* Each of these systems provides a COBOL compiler to translate the COBOL source program into a machine-oriented object program. The ability to advance from one generation of equipment to another in a logical, orderly, and rapid manner is assured through this limited machine dependence. Source programs written in COBOL consist of four major divisions.

## COBOL DIVISIONS

### IDENTIFICATION

This division contains information identifying the source program and the output of a compilation; the author, installation, and so forth also may be identified.

### ENVIRONMENT

This division specifies a standard method of expressing those aspects of a data processing problem that are dependent on the physical characteristics of a specific system; also, it permits specification of the compiling system hardware characteristics, input/output control techniques, and so forth.

### DATA

This division describes the data that the object program is to:

- accept as input;
- manipulate;
- create; or
- produce as output.

The division is further divided into sections to facilitate the description of data contained in input or output files or developed during the program, or present as constant information to be used in the object program.

### PROCEDURE

This division describes the logical steps that must be taken in the solution of the data processing problem.

The levels of *American National Standard COBOL X3.23—1974,* modules are:

## LEVELS

COBOL

- Nucleus - level 2
- Table handling - level 2
- Sequential I/O - level 2
- Relative I/O - level 2
- Indexed I/O - level 2
- Sort - level 2
- Segmentation - level 2
- Library - level 2
- Debug - level 2
- Interprogram communications - level 2
- Communications - level 2

### 3.4.4. BASIC

The Beginner's All-purpose Symbolic Instruction Code (BASIC) language is an interactive programming language designed to be easy to use yet meet the requirements of both business and scientific programming. The BASIC language available on the OS/3 operating system complies with the *American National Standard Minimal BASIC, X3.60-1978* and includes Dartmouth features and compatibility. It provides a powerful, but simple set of commands that allow the novice to learn the language quickly, and yet gives the experienced programmer an extensive list of features for various applications.

BASIC is an interactive language, and all source statements can be entered directly at the workstation with the results and error messages displayed on the screen. All source statements are checked for syntax errors as they are entered, and a message appears on the screen if a line is in error. The BASIC source program can be compiled directly at the workstation, and compilation errors can be corrected immediately. During an interactive BASIC session, users can input, modify, execute, and save programs.

The OS/3 BASIC compiler has facilities for arithmetic operations, data file processing, matrix generation and processing, and logical operations. Subroutines and string operations may be used in a BASIC program.

### 3.4.5. FORTRAN IV

FORTRAN IV is a programming language designed for performing mathematical computations required for solving engineering and scientific problems. FORTRAN IV is also useful for many nonscientific data processing applications.

FORTRAN IV is designed so that the user can express an algorithm in a way natural to the problem. The user requires minimal considerations of the particular characteristics of the system in which the program is executed. Procedures defined outside the FORTRAN IV program, and possibly written in a language other than FORTRAN IV, can be referenced by name and thereby be made an implicit part of the program.

FORTRAN IV is a proper superset of the *American National Standard FORTRAN (X3.10—1966).* It is also a compatible superset of the IBM/DOS 360 FORTRAN IV. This system features code optimization, high-performance I/O, and extended functional capability.

### 3.4.6. Basic Assembly Language (BAL)

OS/3 assembly is a versatile and detailed symbolic language. The combination of a macro facility and the ability to handle procedural directives gives the assembly language unique capabilities. Each instruction within the language is assigned a mnemonic to denote the particular hardware function performed.

The symbolic format for writing an assembly language instruction, macroinstruction, or procedural call consists of three basic symbolic fields; use of the fields requires conformity to simple rules so that efficient translation of symbolic code to object code can be performed. The fields are:

- the label field, which may contain a symbolic name used to provide an entry point or a label for a block of data or a block of instructions;

- the operation field, which must contain a mnemonic instruction code or the name of a macroinstruction or procedural call; and

- the operand field, which provides for a variety of uses ranging from simple to complex specifications.

Combining names, parentheses, arithmetic operators, and self-defining terms into operand expressions makes possible the use of highly sophisticated coding sequences. Operand expressions gain power by being able to include location counter references and literals.

A wide range of data types is provided for constant generation and storage definition. Binary, hexadecimal, decimal, fixed-point, floating-point, and character formats can be used to specify absolute values in the source code.

Output from the assembler run consists of a complete listing of symbolic coding, generated object coding, diagnostic messages, and a cross-reference listing. A relocatable object module is produced that is suitable for linking to other modules prior to loading for subsequent execution.

The assembler recognizes a set of directives that can be used to direct its operation. These directives allow the user to control program sectioning, base register assignment, the format of the output listing, sequence checking, and other auxiliary functions.

The assembler includes a macro facility that can reduce the effort required to write patterns of coding, either repeated in one program or common among several programs. The flexibility of the macro facility allows a macro to be written so that the pattern of coding generated can vary widely, depending upon the parameters supplied with the call. Macro definitions may be specified in two formats: macro and proc. The elements of each format type may not be mixed within a definition; however, definitions of both types are permitted within a program. The OS/3 provides the user with a comprehensive selection of system macros that interface with data management, the supervisor, and other elements of the software.

## 3.5. THE GENERAL EDITOR

The general editor provides the capability to interactively generate source programs, data files, and job control streams. It can also be used to create and edit textual information. The editor responds to a set of commands designed specifically to aid in the generation and formatting of text. The general editor can also store the created material in a program library or a data file. The editor command repertoire provides the user with the ability to:

- Create, add to, delete from, and modify text
- Create, modify, and merge files
- Copy files and texts

The editor offers file protection facilities that prevent a file being modified by the editor from being inadvertently destroyed or incorrectly altered either by direct action of the user or through some system failure.

## 3.6. INTEGRATED COMMUNICATIONS ACCESS METHOD (ICAM)

As with all the OS/3 software, the design emphasis for the integrated communications access method is on ease of use for the casual as well as the major real-time communications user. A wide range of system options provides support for users committed to a real-time multijobbing environment without penalizing those whose needs are more modest. ICAM is available for workstation support as well as for terminals supported over a communication line through the ICAM Terminal Support Facility.

ICAM offers the communications user these features:

## ICAM FEATURES

### MESSAGE QUEUEING

Message queueing stacks complete messages in main or disk storage while they wait to be serviced by a remote device handler or a message processing program. A single message queue consists of one or more messages with their header segments linked together. The message text that overflows the header segments is contained in additional segments, which have secondary links out of the header segment. A network may contain one or more message queues associated with lines, terminals, or process queues.

A user, when defining the network configuration, may select line or terminal message queueing. Terminal queueing results in a single queue being defined for each terminal on a communication line. Line queueing results in a single queue being defined for all terminals associated with a line. When line queueing is specified, the message header prefix contains the terminal name to which a message is directed.

### MULTIPLE DESTINATION ROUTING

The disk queueing facility provides multiple routing of messages. A single message may be routed to up to 255 destinations. Only a single byte is required in main storage to determine when a message has been delivered to all of its destinations. The message is released from its queue only when all deliveries have been made. Any mix of nonbatch devices may be designated in a single multirouted message.

### ACTIVITY SCHEDULING AND PRIORITY CONTROL

An activity scheduling routine is provided with an optional priority suspension and scheduling capability. In small system configurations, only a single level of scheduling is provided; a suspension capability is not required. Multiple priority levels and a suspension capability are provided for large systems with critical timing requirements.

### TIMER SERVICE

A centralized timing service for control of active data buffers and for scheduling an activity is provided for use by all ICAM software elements. The user can specify a value, in 0.7-second increments, to the timer for determining the maximum time period an active buffer can wait for data transmission. In addition, the user can specify a value to the timer routine, in 0.1-second intervals, for determining elapsed time prior to the scheduling of an activity.

### CHECKPOINT/RESTART

Restart procedures can reconstruct message queues to their status at the point of system hardware or software failure. The reconstruction involves only complete messages; incomplete messages must be reprocessed after the recovery operation. Status messages are transmitted to remote terminals to effect an orderly restart and provide assurance that messages are not lost in the system. Restart is supported only in conjunction with a checkpoint and journaling feature. Checkpoint results in the complete network definition being written on a tape or disk file. When checkpointing has been specified, it is performed automatically by the ICAM on an adjustable periodic basis.

▷▷▷

## ICAM FEATURES

### JOURNAL CONTROL

Journal files are required when a user of ICAM requires a recovery/restart capability, or when a statistical accounting of the network operation is required. The journal control routine provides for the recording of message processing events and message data on a disk or magnetic tape file. To reduce the number of I/O accesses required, each journal entry is transferred into a larger buffer area for staging prior to being written on the output device. A method of queueing journal requests by priority permits preferential treatment for critical entry processing.

### STATISTICS ACCUMULATION

ICAM maintains statistics that reflect the operating status of the communication environment. These statistics are maintained by line and by terminal and are available to a user program upon request. The accumulation contains totals for types of information, such as:

- Number of messages received
- Number of input retransmission requests
- Number of messages transmitted
- Number of output transmission requests
- Number of poll messages
- Number of no-traffic responses

These statistics are maintained in the network definition and provide the user with information not generally available from journal files.

A diagnostic log used to accumulate error statistics associated with lines and terminals is also included. These statistics are available to online diagnostic routines that are loaded when an operational failure is detected by ICAM.

At system generation time, the individual user can choose the level of support needed to meet the site requirements. Levels of support are arranged by combining the appropriate ICAM library elements. The levels are named for the interface between the user's message processing program and the ICAM modules combined into a loadable software module. Each interface contains its own unique set of macroinstructions.

## ICAM INTERFACES

### STANDARD INTERFACE

The standard interface is a conventional GET/PUT interface for communications that automatically queues input and output messages in network buffers. In this interface, the processing program requests data from a process file or input queue and not directly from the line. Conversely, output is placed in a destination queue or output queue for transmission by ICAM. An extended set of macroinstructions known as message processing procedure specifications is optionally available to this interface for additional processing of messages.

▷▷▷

## ICAM INTERFACES

### TRANSACTION CONTROL INTERFACE

The transaction control interface provides the communications facilities needed by the information management system (IMS). After defining the communications network, the operation of the transaction control interface is completely transparent to the IMS user.

### DIRECT DATA INTERFACE

This interface permits programs to interface directly with the ICAM remote device handlers. This level of support is intended to provide a communications capability in the smallest systems while maintaining a degree of device independence for the programs.

### COMMUNICATIONS PHYSICAL INTERFACE

Here the interface between ICAM and the user's program occurs at the physical input/output level. This level utilizes less main storage with the greater part of the programming effort shifted to the user program. This interface permits the user to write message control programs without modifying the operating system interfaces.

Each level represents a different tradeoff between the amount of main storage required when ICAM is attached to the system and the variety of services provided. While choosing the higher level of ICAM support seems expensive in terms of main storage usage, writing a message processing program at the communications physical interface level can be both difficult and costly if many services are required.

ICAM is a modular software package capable of supporting either simple or complex communications environments. A single ICAM configuration can provide concurrent support for multiple user programs that handle a variety of terminals and line types.

ICAM prevents conflicting facility assignments and releases facilities when jobs terminate. User programs are provided with macroinstructions that perform the following services:

- control table generation;
- data transfers to and from user-specified buffer areas;
- communication facility initialization and control; and
- dynamic terminal and poll table entry alterations in the communications control areas.

The OS/3 components that make up ICAM are:

## ICAM COMPONENTS

### CHANNEL CONTROL ROUTINE (CCR)

The CCR provides the physical input/output interface to the multiple line or single line communications adapters (MLCA/SLCA) and the specific type of communications subsystems.

### REMOTE DEVICE HANDLERS (RDH)

The remote device handlers provide the software logic and control required to interface the unique characteristics of specific remote devices to the other ICAM components.

RDHs are available to accommodate a wide range of terminal equipment, including the following SPERRY UNIVAC equipment: UNISCOPE 100 and 200 Display Terminals; UTS 400 Universal Terminal System; UTS 4000 Universal Terminal System; and TELETYPE* teletypewriter models 33, 35, 37 and 38. Binary synchronous communications procedures are also supported.

Device handlers provide all device-dependent functions that are required to permit other ICAM components to function independently of the terminal device. These device-dependent functions include:

- Station and device polling where applicable
- Code translation
- Compression and decompression of data where necessary
- Error detection and correction
- Accumulation of operation statistics

### COMMUNICATIONS NETWORK CONTROLLER (CNC)

The CNC coordinates message flow between the remote device handlers and either main storage or a disk-based message queue. The CNC is the ICAM component that places incoming messages on their appropriate processing queues or submits them to special system functions (the user program interfaces are described later) for disposal. It provides the interface to user-specified message processing procedures specification (MPPS) routines and user-generated routines. The CNC also detects the presence of a message on an outgoing (destination) queue and provides the orderly transmission of that message.

Additional functions performed, either directly or indirectly, by the CNC include:

- providing the control to dynamically modify a network in accordance with a changing operating environment;
- monitoring the orderly shutdown of ICAM during end-of-job processing (this service is extended to both ICAM and user programs); and
- scheduling the message user service transcriber (MUST) routines when new activity is detected and there is an outstanding request waiting.

▷▷▷

*Trademark of the Teletype Corporation.*

# ICAM COMPONENTS

## COMMUNICATIONS CONTROL AREA (CCA)

A CCA contains all the tables required to define and control a specific network configuration and CCAs can be tailored to specific needs. A user program may also specify a network CCA by using a control statement with the network name in the run stream. Job control then automatically locates the CCA in the system library and loads it at job execution time.

Network control is exercised by setting indicators and flags in the line, terminal, and queue tables within the CCA. These flags control polling, indicate the operational status of communication hardware, and reflect the current disposition of message queues.

Each CCA contains a pool of network buffers that are under the control of the CNC. Incoming and outgoing messages are temporarily staged in these buffers during their active transition through the system. Buffers from this pool provide the base when main storage queueing is specified. When disk storage queueing is specified, these buffers provide intermediate storage during the active input or transmit phase of a message.

## MESSAGE USER SERVICE TRANSCRIBER (MUST)

A MUST routine provides a message staging service that isolates a user program from the device dependence that is usual in data communications programs. MUST is responsible for copying data into user-designated work areas from the network buffer pool or copying data into the network buffers from user-designated work areas.

Variations of a MUST routine provide support for specific user program interfaces, such as remote batch processor (RBP), RPG II, and IMS. The MUST routine isolates these functions from the CNC and device handlers, obtaining the maximum commonality of ICAM components.

User programs are capable of receiving or sending message data by exercising one or more options when calling the MUST routine. Both batch data and nonbatch modes of operation are provided.

## DEFERRED USER SERVICE TASKS (DUST)

The deferred user service tasks are a series of ICAM overlays that perform those functions that are not time-dependent or that are used infrequently. An overlay control routine within the CNC provides queueing for multiple requests for DUST functions because only one overlay may be active at one time.

The overlays provided by DUST perform functions such as:

- ICAM initialization
- Network initialization
- Subsystem parameter loading
- Line connect and autodialing
- Program termination
- System console message capability

## GLOBAL USER SERVICE TASK (GUST)

GUST allows the formation of a global, or nondedicated, communications network permitting several user programs to access the network concurrently. The GUST routine controls network and line requests and releases in a global environment.

In addition to providing a communications interface between a terminal network and System 80, ICAM can provide the following interfaces.

### ADDITIONAL ICAM INTERFACES

#### RPG II TELECOMMUNICATIONS

RPG II users are provided with a simplified remote I/O capability through RPG II telecommunications and the ICAM direct data interface. Support for RPG II telecommunications is extended to all the devices already listed under remote device handlers, as well as the IBM System/3, the IBM System 360/20 (BSCA equipped), and the IBM System 360 (with OS or DOS BTAM binary synchronous communications support).

#### COBOL MESSAGE CONTROL SYSTEM (CMCS)

Users can write COBOL programs that access and use the resources of their ICAM system. COBOL programs interface with ICAM through the COBOL message control system (CMCS), which handles the flow of control information and data between COBOL programs and ICAM. CMCS translates the COBOL communications verbs into ICAM instructions and translates ICAM error and status conditions into COBOL format. Its access to ICAM is through the standard interface. Message processing procedure specification features can be included in COBOL communications programs.

#### REMOTE BATCH PROCESSING (RBP)

ICAM provides an RBP capability that permits jobs to be entered into the computer system from a variety of remote terminal devices. The software components that perform the RBP functions operate as system tasks and are referred to as symbionts.

These symbionts may be configured as input only, output only, or input and output. In small computer installations, the execution of the input symbiont, the user's job, and the output symbiont occurs serially. In larger systems, the entry and execution of a specific job are serial, but more than one job may be processed concurrently.

Remote symbionts may be loaded and initialized manually via the system console, or these remote jobs may be processed automatically in response to a call from a remote station. An instruction set is available to send job requests, activate workstations, defer processing, obtain status, and deactivate workstations.

### 3.6.1. NTR System Utility

The nine thousand remote (NTR) system utility gives a user the ability to have System 80 act as a remote terminal to a SPERRY UNIVAC 1100 Series system. The NTR is a combination system utility and job task that may run concurrently with other job tasks. It makes use of the direct data interface, with some modification, as the connection between the two systems. The user is afforded data transmission control, operator-to-operator communications, and control features for status and command functions between both systems. The system utility controls local input and output tasks through use of a combination of data management macroinstructions and a set of special NTR macroinstructions for user own code control of device buffers.

## 3.7. INFORMATION MANAGEMENT SYSTEM

The information management system (IMS) is an interactive transaction processing system. IMS is ideal for users who have a need for rapid access to up-to-the-minute information. Information retrieval and updating can be performed by personnel with no formal data processing training.

IMS is transaction oriented. This means that processing is triggered by an input message from a remote terminal (either a display terminal or a teletypewriter). Applications programs, called action programs, process the input message, access data files, and display an output response at the terminal.

The user creates his own online IMS system in a configuration process in which he defines the communications network, user files, action programs, and optional IMS features. Any number of different IMS load modules can be configured. This allows the user to tailor each online IMS system to a specific application and conserves main storage. The configuration process is simplified by an IMS configuration job control procedure and by the use of default parameters.

IMS provides a collection of action programs called UNIQUE, which retrieves, updates, and displays data via a simple inquiry/update terminal language. To use UNIQUE, the user first creates *defined files* from elements of existing files using an IMS utility called the *data definition processor*. A password capability is provided with UNIQUE, which allows the user to limit access to data files or to certain elements within those files.

For specialized file processing or message formatting needs, IMS supports user-written action programs in COBOL, RPG II, and basic assembly language (BAL). Action programs operate under the control of IMS and access files via function calls to IMS. Programming is simplified because IMS handles all communications and file I/O functions. IMS software allocates resources, and schedules action programs, to process each transaction. IMS provides file protection via a record locking facility and online and offline recovery.

IMS can run in a multijobbing environment because IMS is executed as merely another program under control of OS/3.

## 3.8. DATA BASE MANAGEMENT SYSTEM

The data base management system (DMS) is a collection of system programs that support the development of integrated data bases. These programs handle the description, initialization, creation, accessing, maintenance, backup, and recovery of data bases. The languages used in the description and manipulation of DMS data bases are derived from the CODASYL data base specifications. A data base may be accessed by batch application programs and communications application programs.

### DMS FEATURES

PHYSICAL DATA BASE CHARACTERISTICS

The Device Media Control Language (DMCL) uses a COBOL format to describe physical characteristics of a data base and data dictionary by supplying page and area sizes within the data base, the number of data base buffers, recovery options, and file identification.

LOGICAL DATA BASE CHARACTERISTICS

The Data Description Language (DDL) uses a COBOL format to describe the logical characteristics of a data base. The schema DDL describes the entire data base, and the subschema DDL describes the portion of the data base used by one or more application programs. Thus, many different subschemas may exist for the same data base depending on the specific collection of data required by different application programs using the data base.

▷▷▷

# DMS FEATURES

## STORAGE STRUCTURES

Data items are organized into record types, and record types are organized into logical relationships called sets. Records are grouped into three basic logical data structures: sequential, tree, and network, according to record set linkages. In addition, data is organized physically by record types within pages, and pages are organized within areas that are subdivisions of storage. Users can control physical placement of records in a data base and control access to the records by assigning a location mode and set order. This capability increases access efficiency.

## MANIPULATION OF DATA

The data base can be accessed by writing data manipulation language (DML) statements in the procedure division of a COBOL application program. COBOL programs containing DML are processed by a DMS preprocessor to produce American National Standard COBOL source programs that can be subsequently compiled by the COBOL compiler.

The DML contains statements that call a specified subschema, establish contact with the DBMS, open and close areas of the data base, find data base records and get them into working storage, store new records, modify and delete existing records, insert and remove records from set occurrences, save current information, and test for set membership.

## SYSTEM SUPPORT FUNCTIONS

System support functions include three types of utility processors: generation, report and statistics, and recovery utilities. Users control these utilities via the utility processor language (DUPL) commands. Some DMS utility activities include data base initialization, dump and restore, and data base recovery.

## IMS/DMS INTERFACE

IMS action programs can access DMS data bases via data manipulation statements embedded in COBOL action programs. Also, DMS data bases can be used to build a defined file accessible via COBOL, RPG II, or BAL action programs, or UNIQUE.

## RECOVERY

A journal file is generated by the DMS to support the operation of offline recovery utilities. The offline recovery utilities can perform forward recovery of a data base to its final state or backward recovery to any selected run-unit checkpoint. Online recovery automatically restores a data base after a system crash or abnormal program termination.

## 3.9. DISTRIBUTED DATA PROCESSING

The distributed data processing software permits the distribution and cooperative processing of user jobs and files among multiple computer systems. This interchange is based on a standard SPERRY UNIVAC DDP command language implemented within each of the supporting operating systems. The OS/3 DDP consists of:

- a job distribution facility; and

- a file transfer facility.

The DDP job distribution and file transfer facilities allow the user to view each system in the DDP network as an available resource for scheduling and executing his work. Using simple commands, the user can initiate job distribution and file transfer within the system without concern for the requirements of the hardware and software of each system, or the communications protocols needed to initiate and monitor the distribution of a job.

### 3.9.1. Job Distribution Facility

The job distribution facility allows the user to initiate execution of a job on any system within the DDP network and to monitor the execution from the initiating site. Any printed or punched output generated by the job is normally routed to the initiating system. However, the initiator can request routing of this output to any available system in the network.

The job distribution facility provides commands to:

- Submit jobs to the DDP network

- Monitor the execution of submitted jobs

- Cancel a submitted job

- Communicate with a remote operator's console

- Issue instructions to a remote operating system

- Respond to messages issued by a job executing on a remote system

### 3.9.2. File Transfer Facility

The file transfer facility provides the capability to transfer sequential files from one system to another or to duplicate file structures between systems.

Files can be transferred between systems that contain the same file handling facilities, in which case no data transformation is required. Files with serially-accessible records may be transferred between systems of dissimilar architecture with or without character conversion. That is, a file can be transferred as a bit stream to be reformatted by the user, or it can be converted (character translation only) during the transfer process to the internal code recognized by the receiving system.

The file transfer facility does not support automatic reformatting of records within a file, or converting records to their equivalent form in the destination system, when transfer is between systems of dissimilar architectures. In this case, the contents of the file are treated as all character data or bit string data. Neither record sensitivity, record sequence, nor numeric field characteristics are recognized or adjusted during the transfer operation.

File transfer can be directed into the file area of the destination system or, if the destination file area is in use, to a temporary file area in the destination system. Transfer from the temporary file to the destination file is made when the destination file becomes available.

**FILES TRANSMITTED BETWEEN SYSTEMS**

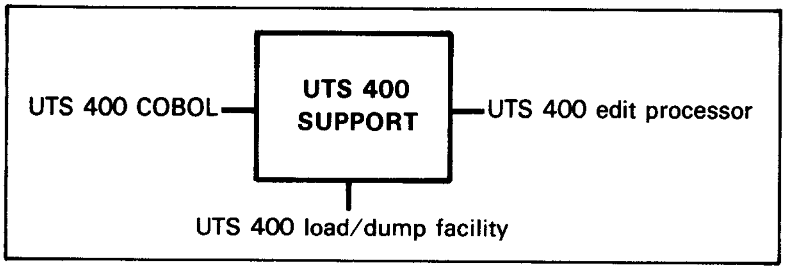
| DATA FILES | PROGRAM LIBRARIES |
|---|---|
| Data formatted in the OS/3 MIRAM format. Files transmitted from OS/3 are MIRAM files; files transmitted to OS/3 will be created in MIRAM format. | Any directly-accessible module in a program library can be transferred between systems, or an entire library can be transferred. |

The file transfer facility permits the user to:

- generate a file directory to catalog the characteristics of files;
- transfer copies of data files and program libraries;
- delete a file from the file directory; and
- obtain a listing of a file's characteristics from the file directory.

## 3.10. UTS 400 SUPPORT

OS/3 offers the following software components to support the SPERRY UNIVAC UTS 400 Universal Terminal System (UTS 400):



### 3.10.1. UTS 400 COBOL

UTS 400 COBOL is a high-level, business-oriented international language with features to complement the capabilities of the programmable UTS 400 terminal. The UTS 400 COBOL compiler meets the ANSI X3.23-1974 standards and the ISO recommendations on COBOL. The compiler also contains extensions to provide for interactive data entry, program control, and screen management. The UTS 400 COBOL compiler executes under control of OS/3 and produces a compiled program that can be downline loaded to the terminal or placed on a diskette.

### 3.10.2. UTS 400 Edit Processor

The UTS 400 edit processor allows the user to create and manipulate text data on a diskette file. It provides an easy and efficient way to create and update line-oriented files of data. Lines may be inserted, replaced, deleted, or changed in any order.

The edit processor permits functions such as string searches, line deletes, insert or delete lines, moves, and print. Error messages advise the user of mistakes made in entering data. Lines up to 118 characters in length may be inserted in an edit processor file.

### 3.10.3. UTS 400 Load/Dump Facility

The UTS 400 load/dump facility provides the following facilities for the use of the UTS 400 COBOL and edit processor:

- a communication program to extract a UTS 400 load module from an OS/3 library and to transmit this module (downline load) to a remote UTS terminal; and

- a communication program to accept UTS 400 dump information and store it on an OS/3 dump file.

## 3.11. APPLICATIONS PROGRAMS

Sperry Univac offers a wide variety of applications programs with the System 80 data processing system. These programs are designed to meet the data processing requirements of particular applications. The list of available packages varies as Sperry Univac institutes changes to meet the needs of its customers.

**APPLICATIONS PROGRAMS AVAILABLE**

UNIVAC Industrial System 80

UNIVAC Industrial System 80-Extended

Accounting Control System 80

Information Collection System 80

UNIVAC Distribution Information System-Wholesale

Order Entry 80

Order Entry 80-Extended

### 3.11.1. UNIVAC Industrial System 80

The UNIVAC Industrial System 80 (UNIS 80) is a comprehensive manufacturing production and inventory planning and control system. UNIS 80 provides the facilities to meet the requirements of a total manufacturing control system, including features for production scheduling, inventory control, and material handling. It is an easy-to-use, interactive system that permits even the most inexperienced personnel to rapidly become proficient in system operation. Terminal screen displays, called menus, lead the user through the system, thus making it available to any member in the organization. UNIS 80 interfaces with an integrated data base and permits multiple interactive users to access the system simultaneously as well as providing noninteractive, or batch, support. UNIS 80 is modular and consists of the following application modules.

# UNIS 80

## PRODUCTION ENGINEERING DATA CONTROL

This feature collects, maintains, and retrieves all basic engineering and production data, used as primary information and as a basis for other applications in manufacturing. It handles bills of material, parts data, manufacturing routings, and work center data. The user is provided with:

- Part engineering data
- Single-level bills of material
- Indented bills of material
- Summarized bills of material
- Manufacturing routings
- Work center data
- Single-level where-used
- Indented where-used
- Summarized where-used
- Highest-level where-used
- Work center where-used

## PRODUCT COSTING

This feature takes information from the data base to accumulate standard costs for materials, labor, and overhead. Areas covered by this application include:

- Cost data entry
- Part cost
- Cost buildup (roll-up or fold-in)
- Costed bill of material

## CUSTOMER ORDER PROCESSING

This feature provides capabilities for gathering and using customer, vendor, and part or end-item information to process customer orders from receipt to shipment. This application includes:

- Customer search
- Customer shipping, billing, and part information
- Entry and control of customer orders
- Inventory checking and availability
- Order status
- Picking slips and confirmation
- Shipping documents
- Invoicing and pre-invoicing
- Returned goods processing

▷▷▷

## INVENTORY STATUS AND CONTROL

This function provides stock and replenishment order transactions and current inventory status. Typical applications include:

- Stock receipts, issues, transfers, adjustments
- Stock location control
- Cycle count
- Order status
- Stock status reporting

## FORECASTING AND ANALYSIS

This feature provides the ability to statistically forecast requirements and to analyze usage. Available features include:

- ABC analysis
- Requirements and scrap factor forecasting – linear, trend, and seasonal models

## MASTER SCHEDULING

The manufacturing user is able to construct a master schedule based on data in the data base and make suitable changes as necessary. The system provides:

- Master schedule report
- Lead-time picture
- Two-level master scheduling
- Master schedule list

## MATERIAL REQUIREMENTS MANAGEMENT

This feature provides the capabilities for both statistical order control and material requirements planning (MRP) processing to be used selectively by part. Among the facilities available are:

- MRP (regenerative and net change)
- Order point control (optional)
- Lot sizing
- Stock and order status reporting
- Order action recommendations
- Requirements by part and order

## PRODUCTION PLANNING

This feature provides the ability to schedule work and determine capacity requirements, such as:

- Work order scheduling
- Capacity requirements planning (infinite)
- Lead-time reduction factors

▷▷▷

**UNIS 80**

WORK ORDER CONTROL

- Work order release
- Material availability checking
- Shop floor reporting
- Scrap reporting
- Work order status

### 3.11.2. UNIVAC Industrial System 80-Extended

The UNIVAC Industrial System 80-Extended (UNIS 80-E) is a version of the UNIS 80 system offering all of the features of that system, plus additional features. In addition, UNIS 80-E allows the user to modify the system to meet particular needs. Included in the system are the following UNIS 80 modules:

**UNIS 80 MODULES**

- Production engineering data management
- Product costing
- Customer order processing
- Inventory status and control
- Forecasting and analysis
- Master scheduling
- Material requirements management
- Production planning
- Work order control

All the modules offer the same features as the UNIS 80 modules; however, a number of modules include additional features. The forecasting and analysis features add model analysis capability. Included in the master scheduling module is the master schedule load analysis feature. In addition, the production planning module incorporates the finite scheduling and loading feature and network scheduling capability.

## 3.11.3. Accounting Control System 80

The Accounting Control System 80 (ACS 80) is a series of packaged applications, written in RPG II, for general business accounting. The application subsystems are: accounts payable, accounts receivable, general ledger, and payroll.

### FUNCTIONAL CAPABILITIES OF ACS 80 SUBSYSTEMS

#### ACCOUNTS PAYABLE

The accounts payable subsystem includes all functions needed to build and maintain the accounts payable files, select invoices for payment, issue the appropriate checks, and print reports reflecting the system operations. The files contain all of the needed vendor information, such as name, address, telephone number, year-to-date purchases and payments, and discount-lost-date.

Control of expenditures is provided by furnishing a vendor disbursements journal. Cash requirement, aged analysis of accounts payable, checks and remittance data, check register, vendor purchase analysis, and a general ledger distribution summary is also provided.

#### ACCOUNTS RECEIVABLE

The accounts receivable subsystem includes all functions needed to build and maintain the accounts receivable files. These files contain all required information on customers, such as name, address, credit limits, year-to-date sales figures, and related invoice information.

Control of cash flow and readily available credit information on customer accounts is provided by furnishing invoice and adjustment registers, aged analysis of receivables, cash receipts register using the open item and the balance forward method, cash expectancy forecast, and sales analysis reports.

#### GENERAL LEDGER

The general ledger subsystem provides for the control of accounting records, including an audit trail of entries and the balancing and the validation of all bookkeeping entries. Balance sheets and income statements are produced.

The general ledger subsystem will provide: trial, detail, and consolidated balance sheets; detail, consolidated, and comparative income statements; and lists of debit/credit entries, and account balance before posting to the general ledger chart of accounts. It accepts input from accounts payable, accounts receivable, payroll, and inventory systems.

#### PAYROLL

The design of the payroll subsystem is patterned after manual methods with which accountants and bookkeepers are accustomed.

The system performs all tasks and maintains all accounting records required by the Internal Revenue Service.

The subsystem will process salaried, hourly, or piece-work employees and compute, remit, and report taxes, deductions, and fringe benefits. Payroll checks and stubs, journal entries to general ledger, job labor costs, and labor distribution reports are also prepared.

### 3.11.4. Information Collection System 80

The Information Collection System 80 (ICS 80) is a highly efficient online information collection system implemented as action programs to IMS. It offers a practical and economical solution to information and data collection problems. The system helps to ensure optimum utilization of computer resources, aids in the introduction of online information processing, and provides the following important capabilities:

**ICS 80 CAPABILITIES**

- Online collection of data and information. The same workstation used for the collection of data and information may also be used for file inquiry and updating.
- A full range of data validation and checking routines. These may be application-dependent checks specified by the user.
- Information and data collection handled simultaneously with other processing in a multiprogramming environment.
- A simple implementation language for specifying the formats to be used to enter data.

### 3.11.5. UNIVAC Distribution Information System - Wholesale

The UNIVAC Distribution Information System - Wholesale (UNIDIS - WHOLESALE) offers a complete distribution control system designed to optimize cash flow, increase profits, streamline operations, and improve customer service. UNIDIS - WHOLESALE ordering strategies maximize inventory while holding down cost, and offer the user positive control over all goods flowing into and out of the user organization. It is a real-time system allowing the user to respond immediately to customer orders and retrieve billing information quickly and easily.

UNIDIS - WHOLESALE functions can be broken down into order entry, stock control, and inventory management.

**UNIDIS - WHOLESALE FUNCTIONS**

ORDER ENTRY

UNIDIS - WHOLESALE offers real-time, online order entry with immediate response capability. In addition, order entry offers:

- Online availability determination/reservation
- Specialized delivery instructions and comments, including item substitution
- Standard ship-to and bill-to address
- Automatic discount and pricing capabilities
- Profitability control
- Customer credit limit control
- Invoice transactions
- Automatic pick list generation and route optimization
- Inventory and demand history updating
- Pre- and post-billing accounting procedures
- Blanket order, back order, and drop shipment processing
- Automatic assignment of customer order identifiers

▷▷▷

## UNIDIS - WHOLESALE FUNCTIONS

### STOCK CONTROL

Stock control provides for the control of goods from their receipt on the shipping dock, through count verification, inspection, repair, and rework until the goods reach stock or scrap. Stock control offers:

- Online processing
- Receipt verification against purchase orders
- Quantity tolerance verification
- System control of goods movement
- Inventory updating
- Generation of financial transactions
- Location control of all goods, stock, and nonstock

### INVENTORY MANAGEMENT

Inventory management provides a sophisticated set of statistical features to analyze demand patterns and suggest replenishment strategies. The system provides the following features:

- Demand models with automatic model analysis
- Service level specification
- EOQ calculation
- Tracking signal/demand filters
- Alarm reports
- Graphic representation of demand patterns and forecast model
- Product group or warehouse processing

## 3.11.6. Order Entry 80

Order Entry 80 is a complete, customer-order processing system that handles all aspects of order handling from initial receipt and entry of an order to final shipping of materials. It is an easy-to-use, interactive system that, through sophisticated application of software, permits any member of an organization to use the system with a minimum of training.

**ORDER ENTRY 80 APPLICATIONS**

- Customer shipping, billing, and pricing
- Customer part information (prices and terms)
- Part information
- Inventory posting
- Inventory availability checking
- Customer order entry
- Order retrieval
- Order changing
- Back order generation
- Order status review and update
- Profitability control
- Picking slips and picking confirmation
- Printing shipping documents
- Invoicing
- Returned goods processing

To assist those who use the system, a series of screen displays, called menus, guides the user through the system. All operations can be performed simply by following the instructions included in the menus.

Order Entry 80 provides a complete data base with data defined to fulfill the order processing and inventory status features. Installation of the system can be performed quickly and easily, and immediate operation can begin after loading user system control data and actual application data into the data base.

### 3.11.7. Order Entry 80-Extended

No two organizations conduct business or run their factory operations in exactly the same manner. Differences in personnel, customers, and products force differences in methods. It is because of these differences that Sperry Univac offers Order Entry 80–Extended. Order Entry 80–E offers all of the advanced capabilities of Order Entry 80 with one significant difference: the user can alter the system to meet unique or changing needs. Thus, instead of having to alter established procedures to fit patterns acceptable to a programming system, the user can change the system to fit the current methods.

Order Entry 80–E includes source code modules that the user implements and makes changes or additions to so as to meet specific business or production environments.

**ORDER ENTRY 80-E CAPABILITIES**

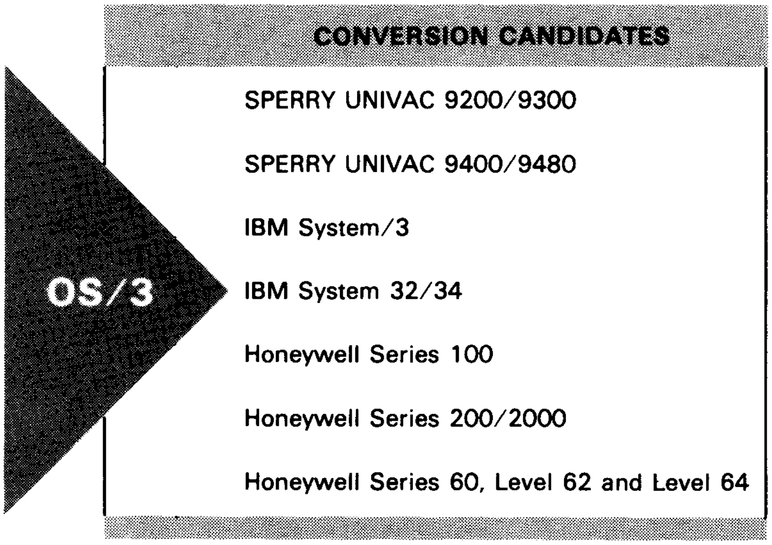
- Multiple interactive access to the system
- Services to cover every aspect of order entry and delivery
- Easy-to-follow screen displays making the system accessible to all personnel

PLUS:

**SYSTEM AVAILABILITY TO MEET UNIQUE OR CHANGING NEEDS**

## 3.12. CONVERSION AIDS

In the rapidly changing world of the computer industry, the need is constantly arising to upgrade the present computing system to one that is more powerful, economical, or that meets changing needs. Because of its advanced design – offering power and versatility with economical operation – System 80 is seen as being the growth path for a number of other systems.



CONVERSION CANDIDATES

SPERRY UNIVAC 9200/9300

SPERRY UNIVAC 9400/9480

IBM System/3

IBM System 32/34

Honeywell Series 100

Honeywell Series 200/2000

Honeywell Series 60, Level 62 and Level 64

As a result, Sperry Univac supplies a number of conversion aids to users who decide to convert from one of these systems to System 80. In addition, there are a number of areas of compatibility between System 80 and the other systems making the conversion a fairly straightforward process.

### 3.12.1. IBM System/3

OS/3 software provides a significant amount of compatibility with the IBM System/3. Among the major areas of compatibility are:

- A System/3 compatible sort, SORT3, that accepts System/3 parameters.
- An access method, MIRAM, that is functionally compatible with the System/3 disk access method.
- A System/3 mode on the OS/3 RPG II compiler that permits direct compilation of System/3 RPG II source programs.
- Parameter specifications for $DELET, $COPY, $KCOPY, $DCOPY, and $MAINT can be used directly on System 80 to duplicate the utility functions.
- An OCL processor that accepts and processes System/3 OCL control streams

In addition, Sperry Univac supplies a number of conversion aids for those areas of incompatibility between OS/3 and System/3:

- Disk data file conversion procedure
- A Model 10, 12, and 15 source and proc transcriber
- OCL to JCL converter (JCLCON801)

#### 3.12.1.1. Disk Data File Conversion

System/3 data files must be first dumped to magnetic tapes or diskettes using one of the System/3 utilities, for example $KCOPY. The tapes or diskettes are then submitted to the OS/3 data utilities to reload the files to disk storage devices.

#### 3.12.1.2. Model 10, 12, and 15 Source and Proc Transcriber

Model 10, 12, and 15 source module and OCL procedure libraries must be copied to either diskette or tape transfer files. These files are in turn transcribed to OS/3 libraries by the source and proc transcriber, COPYS3. Diskette transfer files can be generated directly from the source and procedure libraries, using the IBM library maintenance program ($MAINT). If tape transfer files are to be generated, the libraries must be copied to disk first, using $MAINT. The disk files holding the libraries are then copied to tape, using the IBM copy/dump program ($COPY).

#### 3.12.1.3. OCL to JCL Converter (JCLCON801)

IBM System/3 operation control language (OCL) streams can be quickly converted to OS/3 job control language (JCL) streams through the OCL to JCL converter, JCLCON801. JCLCON801 executes on the OS/3 system, accepts System/3 OCL streams, and generates as closely as possible OS/3 JCL streams. Where direct conversion is not possible, the converter attempts to provide as similar a function as possible. Where translation is not possible at all, the converter makes no changes to the code and issues a warning message indicating the areas not translated.

### 3.12.2. IBM System 32/34

IBM System/32 and System/34 data files can be transcribed to OS/3 disk by first running the IBM TRANSFER procedure to produce a diskette. This diskette is then input to the OS/3 data utility. The data utility can then produce a user-tailored OS/3 disk file.

System 32/34 source and proc files can be transcribed to OS/3 files, using the SPERRY UNIVAC COPYS3 source and proc transcriber (also a System/3 conversion aid). The IBM libraries are first copied to a diskette through the IBM $MAINT utility, and this diskette is used as input to COPYS3.

### 3.12.3. SPERRY UNIVAC 9200/9300 System

The following areas of compatibility exist between the 9200/9300 system and System 80:

- The OS/3 RPG II compiler provides a 9200/9300 mode that permits direct compilation of 9200/9300 RPG programs on System 80 without source code translation.

- 9200/9300 sequential tape files developed on the UNISERVO VI-C or UNISERVO 12 tape devices can be mounted on System 80 UNISERVO 10 tape drives and processed directly by OS/3 programs.

In addition, Sperry Univac supplies a number of conversion aids for those areas of incompatibility between OS/3 and the 9200/9300 system. These aids are:

- 9200/9300 data file transcriber to convert 9200/9300 data files to OS/3 format (UNLOAD/DATA)
- Assembly language translator (TRASM3)
- COBOL and COPY translator (COBTRN305)
- Library transcriber (COPY93)

**9200/9300 CONVERSION AIDS**

9200/9300 DATA FILE TRANSCRIBER

The data file transcriber (UNLOAD) supplied by Sperry Univac is executed on the 9200/9300 system to copy 9200/9300 disk data files to a tape file. This tape file can be used as input to the OS/3 data utility, DATA, which in turn generates the appropriate disk file from the tape.

9200/9300 ASSEMBLY LANGUAGE TRANSLATOR

The 9200/9300 assembly language source statements can be translated into OS/3 basic assembly language statements through the 9200/9300 to OS/3 assembly language source translator (TRASM3).

COBOL AND COPY TRANSLATOR

The 9200/9300 COBOL source programs and COPY library elements can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN305 translator.

9200/9300 LIBRARY TRANSCRIBER

The 9200/9300 library files can be converted to OS/3 format through the OS/3 COPY93 library transcriber. COPY93 accepts a 9200/9300 formatted tape as input and produces an OS/3 formatted disk file.

## 3.12.4. SPERRY UNIVAC Operating System/4 (OS/4)

OS/3 offers a high degree of compatibility to the 9400 and 9480 users who have been operating under OS/4. OS/4 RPG and FORTRAN source programs can, for the most part, be recompiled by the OS/3 compilers. Any changes required will be minor. A conversion guide that details all the steps required to migrate from OS/4 to OS/3 is available. For those areas of incompatibility, Sperry Univac supplies the following conversion aids:

- Job control language translator (JCON1)
- Assembly language translator (ASMTRN)
- COBOL and COPY translator (COBTRN301)
- Data file converter (DCON4)
- Library transcriber (COPY94)

## OS/4 CONVERSION AIDS

### OS/4 JOB CONTROL CONVERTER

The OS/4 to OS/3 job control language converter (JCON1) supplied by Sperry Univac converts OS/4 job control source statements to OS/3 compatible job control statements. Input to the JCON1 utility can be a magnetic tape containing only control streams (no procs) produced by an OS/4 FILE command, cards, or an OS/3 disk file created by an OS/3 FILE command, the COPY94 utility, or the OS/3 librarian. JCON1 outputs to cards, the printer, or to a disk file.

### OS/4 ASSEMBLY LANGUAGE TRANSLATOR

OS/4 basic assembly source statements can be translated into OS/3 assembly statements through the OS/4 to OS/3 assembly translator (ASMTRN).

### COBOL AND COPY TRANSLATOR

OS/4 COBOL source programs and COPY library elements can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN301 translator.

### DISK DATA FILE CONVERTER

Disk data files can be converted to OS/3 format by using the disk data file converter (DCON4) to dump the files onto tape and then inputting the tape to the OS/3 data utility that, in turn, builds the appropriate data file.

### OS/4 LIBRARY TRANSCRIBER

OS/4 library files can be converted to OS/3 format through the OS/3 COPY94 library transcriber. COPY94 accepts an OS/4 formatted tape as input and produces an OS/3 formatted disk file. The input tape must be generated through the OS/4 tape and disk librarians.

## 3.12.5. Honeywell 100 Series

Users migrating from the Honeywell 100 Series data processing systems will find a high degree of compatibility with System 80. For those areas of incompatibility, Sperry Univac offers the following conversion aids:

- COBOL translator (COBTRN304)
- Data file translator (TAPCON)

### 3.12.5.1. COBOL Translator

Honeywell 100 Series COBOL source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN304 translator.

#### 3.12.5.2. Data File Translator

Honeywell data files can be converted to OS/3 format by using the data file translator (TAPCON). The Honeywell data files must be copied to tape or card, then submitted to TAPCON for reformatting. The resultant tape file can be converted to the ultimate intended file media.

### 3.12.6. Honeywell 200/2000 Series

Sperry Univac provides the following conversion aids for those users converting from the Honeywell 200/2000 Series systems to System 80:

- COBOL translator (COBTRN302)
- EASYCODER converter (ETC3)
- Data file transcriber (TAPCON)

#### 3.12.6.1. COBOL Translator

Honeywell 200/2000 Series COBOL source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the COBTRN302 translator.

#### 3.12.6.2. EASYCODER Converter

Honeywell EASYCODER source programs can be converted directly into OS/3 compatible ANSI 1974 COBOL through the ETC3 translator.

#### 3.12.6.3. Data File Transcriber

Honeywell data files can be converted to OS/3 format by using the data file translator (TAPCON). The Honeywell data files must be copied to tape or card, then submitted to TAPCON for reformatting. The resultant tape file can be converted to the ultimate intended file media.

### 3.12.7. Honeywell 60 Series, Level 62 and Level 64

The following conversion aids are available to those users converting to System 80 from the Honeywell 60 Series, Level 62 and Level 64 systems:

- Program library and data file transcriber
- COBOL translator

#### 3.12.7.1. Program Library and Data File Transcriber

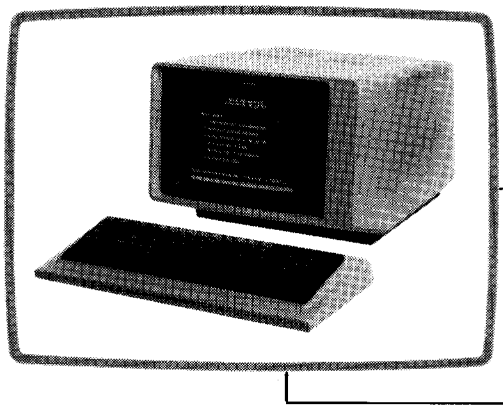
Honeywell data files and program libraries can be transcribed to OS/3 format through the program and data file transcriber (TAPCON). The Honeywell data files and program libraries must be copied to tape or card, then submitted to TAPCON running on OS/3 for reformatting. The resultant tape files can be converted to the ultimate intended file media.

### 3.12.7.2. COBOL Translator

Sperry Univac provides a translator to convert Honeywell 60 Series COBOL to OS/3 compatible ANSI 1974 COBOL.

## 3.13. MAINTENANCE AND DIAGNOSTIC SOFTWARE

The System 80 hardware diagnostics are used by the user and Sperry Univac personnel to isolate and identify system hardware faults. The hardware diagnostic system consists of:



- ▷ Resident diagnostics
- ▷ Offline diagnostics
- ▷ Online diagnostics

The hardware diagnostic routines are designed to operate in an interactive programming environment and make use of displayed dialogs and messages for initiation and entry of control information.

### HARDWARE DIAGNOSTICS

RESIDENT DIAGNOSTICS

The resident diagnostic routines are an integral part of the system control software and perform basic error checking on the central processor complex. The central processor complex includes the central processor, control storage, and the resident disk control hardware. The resident diagnostics also check the paper peripheral, diskette, and workstation controllers along with the single line communications adapter. These checks are performed automatically whenever the system is powered on or reset.

OFFLINE DIAGNOSTICS

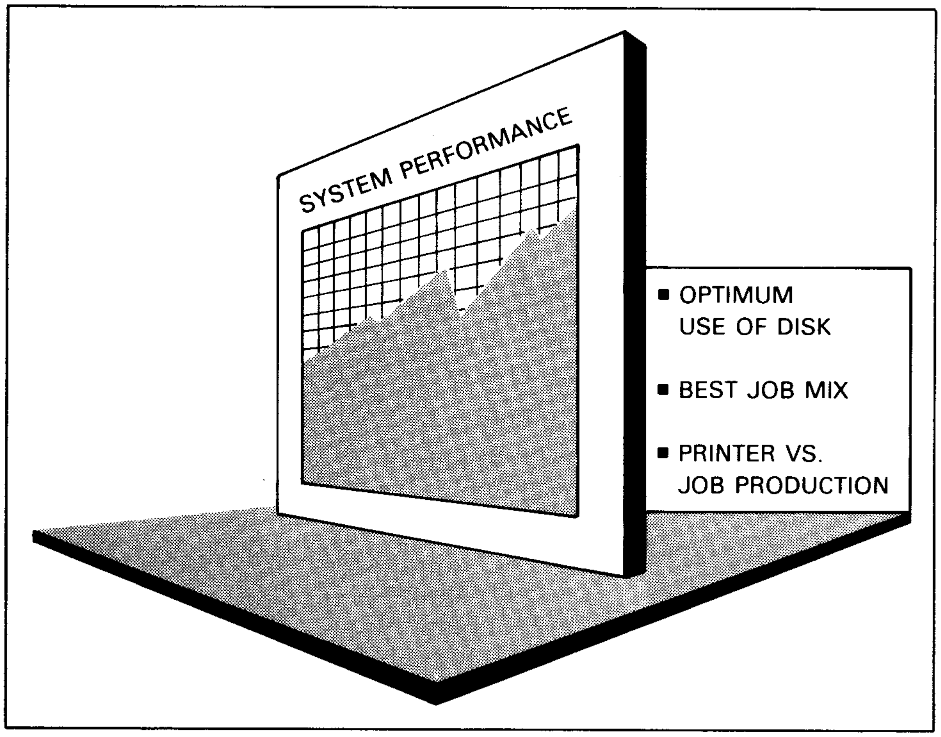
The offline diagnostics consist of microdiagnostic and macrodiagnostic routines. The microdiagnostic routines are used to isolate detected hardware faults to the particular component responsible. Initiated at the workstation, these routines diagnose problems in the central processor, control storage, and main storage. The macrodiagnostics check the operability of the complete repertoire of system instructions. Initiated at the workstation, they can additionally be used to verify the basic operational soundness of the integrated 8417 disk subsystem.

ONLINE DIAGNOSTICS

The online diagnostic routines check all peripheral and communications devices to ensure proper functioning and operation. These routines operate under control of the operating system and run concurrent with user jobs. During normal system operation, errors are logged to allow interrogation and analysis by these routines. Error log edit and analysis programs are available to display individual error events and summary information.

## 3.14. THE SYSTEM ACTIVITY MONITOR

The system activity monitor is used to measure system efficiency by monitoring and recording various system activities. It is an optional software feature configured during system generation. The monitor measures such functions as central processor and main storage usage, disk usage, input/output operations, supervisor interrupts, and, in a system configured for communications, various communications activities. Using the information gathered by the monitor, the user can detect possible production bottlenecks, develop an optimum job mix, change system variables to enhance system performance, and determine if the best use is being made of the system's peripheral devices.



The system activity monitor is activated and controlled interactively through the system console. Information is accumulated continuously and is output to the console at specific time intervals. The monitor can also be instructed to output data to a disk file. By using a report program called SAMRPT, the user can have tailored reports generated from the information placed in the disk file.

# 4. Hardware

## 4.1. SYSTEM 80 MODELS

System 80 is available in two models: Model 3 and Model 5. The two models are identical in capability and available features, with the exception that the Model 5 is configured with high-performance control storage resulting in a 55 percent increase in central processor speed. Model 3 is field upgradable to a Model 5.

## 4.2. HARDWARE CONFIGURATIONS

### 4.2.1. Basic System

The basic, or minimum, System 80 with extended diskette drives consists of a processor complex and a printer. The processor complex contains a console workstation, an 118.2 MB integrated disk subsystem, a diskette drive, three integrated input/output controllers, and 262K bytes of main storage.

### 4.2.2. Expanded System

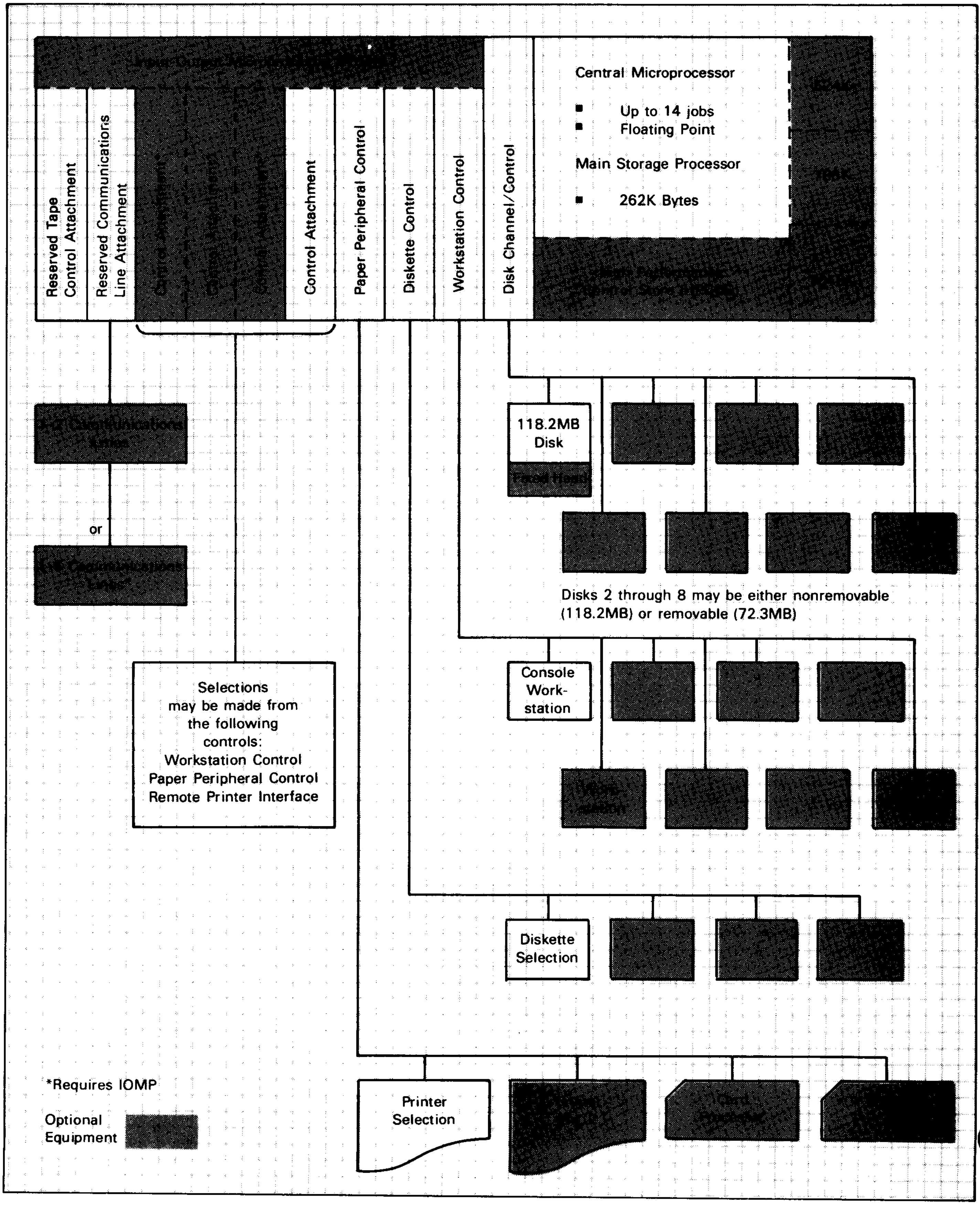
System 80 can be expanded as needed, by the addition of field-installable modules, to increase its processing speed and throughput rate. Maximum expansion is provided with the addition of a high-performance control storage (HPCOS) facility, micrologic expansion, an input/output microprocessor (IOMP), three additional input/output controller connections, six additional communications lines, and three main storage extensions of 262K bytes each.

The HPCOS facility supplants normal control storage and increases the processing speed of the central processor complex by approximately 55 percent. HPCOS is a standard feature of the Model 5. It can be added to the Model 3 to upgrade it to a Model 5.

The IOMP increases the input/output processing capacity of the central processor complex and thereby allows its connection with up to six additional communications lines and three additional peripheral subsystem controller interfaces.

The additional main storage capacity allows for more and larger jobs to be processed concurrently.

## SYSTEM 80 CONFIGURATION



Expansion without any of these features is also possible by simply adding I/O devices in the system up to the maximum number allowable for each required peripheral subsystem. Four peripheral subsystems are required in all systems:

1. Disk subsystem with integrated nonremovable disk drive can control up to seven additional fixed or removable media drives.

2. Diskette subsystem with integrated diskette drive can control up to three additional diskette drives.

3. Workstation subsystem with console workstation can control up to seven additional workstations.

4. Paper peripheral subsystem with printer can control up to three additional devices (i.e., printer readers, and punch).

The basic and expanded system capacities are listed and described in Table 4–1.

*Table 4—1. Basic and Expanded System Capacities*

| Item | Basic System | Expanded System |
|---|---|---|
| Main storage | 262K bytes | 1048K bytes |
| Disk storage | | |
| Number of drives | 1 (nonremovable media) | 8 (nonremovable and removable media) |
| Capacity | 118.2 megabytes (MB) | 118.2 MB per nonremovable media drive;<br>119.06 MB per nonremovable media drive with fixed heads;<br>and 72.3 MB per removable media drive |
| Diskette drives | 1 (manual or autoload) | 4 (2 of which may be autoload) |
| Workstations | 1 (workstation console) | 7 (39 with additional workstation controllers installed) |
| Printers | 1 | 2 (10 with additional paper peripheral controllers installed) |
| Magnetic tape units | 0 | 8 with tape controller installed |
| Card readers | 0 | 2 (10 with additional paper peripheral controllers installed) |
| Card punches | 0 | 1 (5 with additional paper peripheral controllers installed) |
| Communications lines | 2 | 8 |
| I/O controller channels | 4* | 7* |

*Unused I/O controller channels may be used to connect remote printer controllers, paper peripheral controllers, and workstation controllers. However, only one channel may be dedicated for tape use.

## 4.3. CENTRAL PROCESSORS

The processor complex features the use of large scale integration (LSI) microprocessors with microprogramming logic providing higher speed as well as greater reliability. To assure maximum reliability, parity generation and checking is used extensively, and automatic instruction retrying is performed.

The processor complex contains two modular processors and associated storage units: a control processor with control storage, and a main storage processor with main storage unit.

### 4.3.1. Control Processor

The control processor provides arithmetic computations as well as control logic required for instruction execution, system control, and I/O channel support in conjunction with control storage. Control storage is a modular unit containing microinstructions for control of the control processor.

### 4.3.1.1. Basic Instruction Set

The basic instruction repertoire for System 80 consists of 172 instructions. This includes 44 floating-point instructions as well as 22 instructions of various types to increase the power and reliability of the system. Additional unprivileged instructions are available.

**INSTRUCTION FORMATS**

Each format consists of an operation code and two or more fields that specify addresses of the operands. Instructions can be two, four, or six bytes in length. All instructions must start on an even-address boundary.

REGISTER-TO-REGISTER (RR) INSTRUCTIONS

RR instructions process data contained in registers. The maximum length of the data is a double word in floating-point registers and a full word in general registers. Data can be a signed or unsigned binary number, a short- or long-format floating-point number, or a decimal number, depending on the specified operation. Operand 1 specifies either a register or mask. Operand 2 specifies a register.

Some RR instructions use both operands as an immediate data operand.

REGISTER-TO-INDEXED-STORAGE (RX) INSTRUCTIONS

RX instructions process data between registers and indexed storage. A double word is the maximum length of data handled. Data can be a signed or unsigned binary number, short- or long-format floating-point number, or decimal number. Operand 1 specifies a register or mask. Operand 2 specifies a main storage location.

REGISTER-TO-STORAGE (RS) INSTRUCTIONS

RS instructions perform multiple register and storage operations, as well as data shifting. The first and third operands specify the numbers of two general registers or the boundaries for general multiregister usage. Operand 2 specifies a main storage location.

▷▷▷

## INSTRUCTION FORMATS

STORAGE (S) INSTRUCTIONS

S instructions, which have a 16-bit opcode, perform operations on main storage locations, I/O channels, and other System 80 hardware locations. The operand, called operand 2, is specified by a 4-bit base register number and a 12-bit displacement. Although most S instructions form the operand 2 address by combining the base and displacement contents, some use only the base register.

STORAGE-AND-IMMEDIATE-OPERAND (SI) INSTRUCTIONS

SI instructions perform operations on an 8-bit value, called immediate data, and an operand in main storage. Operand 2 specifies the immediate data or mask. Operand 1 specifies a 1-byte or half-word main storage location.

STORAGE-TO-STORAGE (SS) INSTRUCTIONS

SS instructions perform operations on two operands located in main storage. In logical operations, the operands are assumed to be equal in length and can vary from 1 to 256 bytes. In decimal operations, the operands can be of different lengths and can vary from 1 to 16 bytes.

### 4.3.1.2. Nonprivileged Instruction Set

Nonprivileged instructions process fixed-length binary numbers, floating-point numbers, packed and unpacked decimal numbers, and EBCDIC or ASCII characters. Data can be transferred between main storage and the user program set of general registers, as well as from one location in main storage to another main storage location. The operations of shifting, branching, and logical functions also are included.

### 4.3.1.3. Privileged Instruction Set

Privileged instructions are used by the operating system software when operating in the supervisory state. In this state, all installed instructions are valid and can be executed. This set of instructions includes facilities to load and store the contents of low-order main storage and to load the writable section of the microprogram control storage. Privileged instructions cannot be included in a user program.

### 4.3.2. Main Storage Processor

The main storage processor (MSP) controls the main storage unit (MSU). This processor provides preprocessing of the main storage addresses and conversion from relative to absolute addresses. Protection is provided to assure data and program integrity in the system. Data alignment and error correction code (ECC) logic for automatic single-bit error detection and correction, and double-bit error detection, is provided by the MSP.

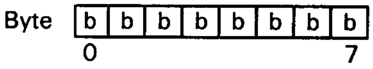
The MSU stores data in word format. The MSU can be configured at a minimum of 262K bytes and can be expanded in increments of 262K bytes to 1048K bytes maximum.

The exact storage sizes available are:

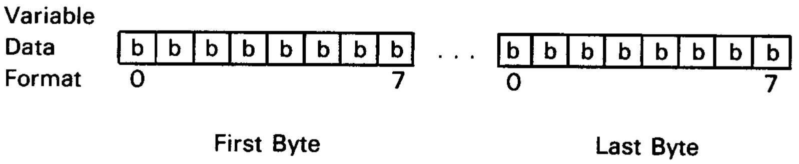
- 262,144 bytes
- 524,288 bytes
- 786,432 bytes
- 1,048,576 bytes

Address and data parity checking is provided.

Locations in main storage are addressed consecutively from 0 through a maximum of 1,048,576. Bytes may be accessed separately or in groups. A group of bytes is addressed by the leftmost byte of the group. The bits in a byte are numbered from left to right starting with 0.



Variable formats consist of a variable number of consecutive bytes.



### 4.3.2.1. Low-Order Main Storage

The low-order 640 bytes of main storage are reserved for specific operating information. This information is accessed by the hardware and operating system during execution of appropriate functions. The operating system provides for loading and protecting the information in these locations.

### 4.3.2.2. Storage Protection

Program integrity in a multijobbing environment is guaranteed by the storage protect feature. Fifteen keys are provided for this purpose. Each 1024-byte segment of main storage may be protected on a write or read/write basis.

## 4.4. SPERRY UNIVAC PERIPHERAL SUBSYSTEMS

The expanded system configuration allows a variety of SPERRY UNIVAC peripheral subsystems to be connected to the system. These I/O facilities provide for the communication of commands, status, and data between the system and its users.

All subsystems are configured with an I/O subsystem controller for disk drives, diskette drives, workstations, and paper peripherals. Additional controllers and communications lines can be attached. The controllers may be considered as individual processors for their associated peripheral devices and, together with their devices, form a subsystem.

### 4.4.1. Disk Subsystems

A unique microprocessor controls the channel that directly accesses main storage for disk drives. All system configurations contain an internal, nonremovable disk. Up to seven additional disk drives that may use removable or nonremovable disk storage media may be added to the system.

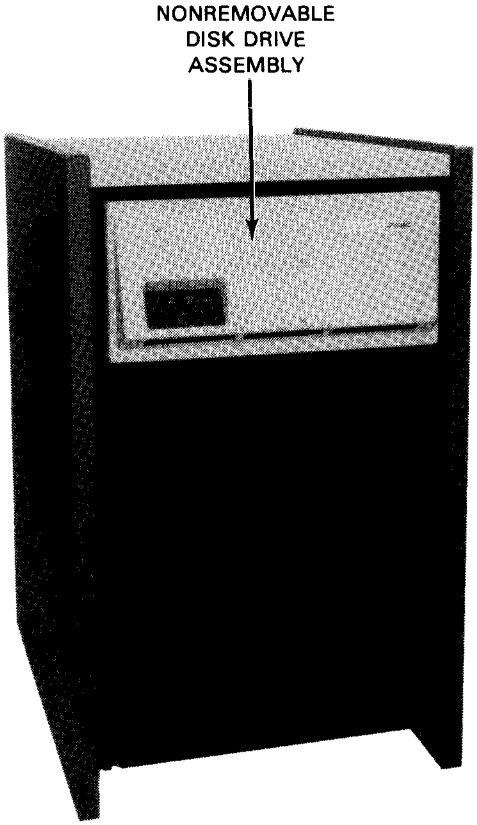
The disk channel/controller (DC/C) connects the internal, nonremovable media disk drive, and optional fixed or removable media disk drives, to the integral microprogrammed microprocessor. The DC/C controls disk drives to read, write, and search for data on the disk surface. It can position the accessor arm to any software-specified location on the active portion of the disk surface and perform other operations specified by the software. The DC/C also provides error correction codes for error bursts of five bits or less and performs diagnostic checks of the subsystem's integrity. The DC/C data transfer rate is up to 1.2 megabytes (MB) per second.

DC/C characteristics are:

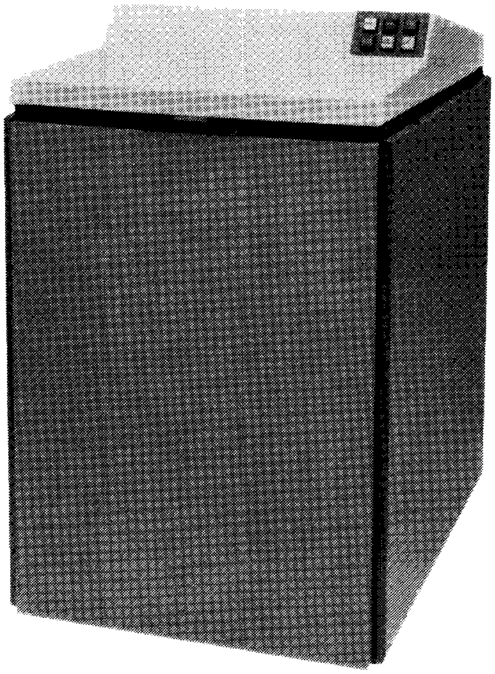
- Data rate of up to 1.2 MB/second
- Record format of 256 bytes/sector
- Error checking and recovery facilities
- Selection of 1 to 8 disk drive attachments
- Intermix of nonremovable and removable disks
- Direct access to main storage

**DISK SUBSYSTEMS**

NONREMOVABLE FREESTANDING DISK DRIVE



REMOVABLE DISK DRIVE



### 4.4.1.1. Nonremovable Disk Drive

The nonremovable disk drive is a freestanding storage device that includes its own power supply, interface cables for direct connection to the DC/C, and an operator control panel. The disk drive is a random access storage device capable of storing up to 118.2 MB of information. Up to three nonremovable disk drives may be housed in a single cabinet.

The nonremovable disk drive features fixed disk media. Each data surface contains two read/write heads that are positioned by an actuating mechanism.

An optional feature employing 60 fixed heads and an additional storage capacity of 860,160 bytes may be included to significantly reduce disk seek time and thereby improve its I/O response time.

The disk drive has a single head/disk assembly (HDA) containing the media consisting of four disks, each 14 inches (355.6 mm) in diameter, mounted on a single spindle and enclosed in a metal case. Included in the disk drive are four movable data head/arm assemblies, as well as a single positioning head/arm assembly. The HDA is a nonremovable module enclosed with the necessary electronics and hardware for head movement. Table 4-2 lists the nonremovable disk subsystem characteristics.

### 4.4.1.2. Removable Disk Drive

The removable disk drive is a freestanding, removable media device that includes its own power supply, interface cables for direct connection to the DC/C, and an operator control panel. The disk drive is a random access storage device, capable of storing up to 72.3 MB of information. Four disk platters are used, and seven of these surfaces are used for storing data. Table 4-2 lists the removable disk subsystem characteristics.

*Table 4—2. Disk Subsystem Characteristics (Part 1 of 2)*

| | Nonremovable Disk Subsystem | Removable Disk Subsystem |
|---|---|---|
| **Characteristics** | | |
| Data capacity (8-bit bytes) per pack (MB) | 118.2 | 72.3 |
| Number of disk platters | 4 | 4 |
| Disk speed (rpm) | 3400 | 2800 |
| Data transfer bit rate (megabytes per second) | 9.05 | 6.272 |
| Bit density (bits per inch) | 6366 (250.6 bpmm) | 5050 (198.8 bpmm) |
| Track density (tracks/inch) | 476 (18.7 tracks/mm) | 370 (14.5 tracks/mm) |
| Number of surfaces per disk unit | 7 | 7 |
| Positioning time*(ms)<br>Minimum<br>Average<br>Maximum | <br>7<br>35<br>70 | <br>10<br>33<br>60 |
| Bytes per sector | 256 | 256 |
| Sectors per track | 60 | 50 |
| Tracks per cylinder | 14 | 7 |
| Cylinders per drive | 550 + 10 alternate cylinders | 808 + 7 alternate cylinders |

*Table 4—2. Disk Subsystem Characteristics (Part 2 of 2)*

| | **Nonremovable Disk Subsystem** | **Removable Disk Subsystem** |
|---|---|---|
| **Characteristics** | | |
| Tracks per drive | 7700 + 126 alternate tracks | 5656 + 49 alternate tracks |
| Data capacity per track (kilobytes) | 15.4 | 12.8 |
| Data capacity per cylinder (kilobytes) | 215 | 89.6 |
| Data heads per drive | 14 | 7 |
| Data transfer byte rate (nominal) | 1.13 MB/second | 784 kB/second |
| Latency time**(ms) | 8.82 (average) | 10.7 (average) |
| Optional fixed heads:<br>Number of data heads<br>Tracks per drive<br>Data surfaces | <br>56<br>56<br>1 | None |

* Positioning time is the time interval from issuing a seek-start command to the generation of a gated attention signal by the drive. Maximum access time for minimum and maximum seek length is the average of at least 1024 seeks. Average access seek time is the sum of the time required to do all possible combinations of seeks divided by the number of such seeks.

** Time from reception of command to response with data.

## 4.4.2. Workstation Subsystems

A workstation subsystem consists of a microprocessor controller and up to eight workstations. Up to five workstation controllers may be included in an expanded system. Additionally, each workstation included in a system can be configured with a directly connected corresponding output printer. Sperry Univac offers two workstation printers: the 80-column printer and the 132-column printer.

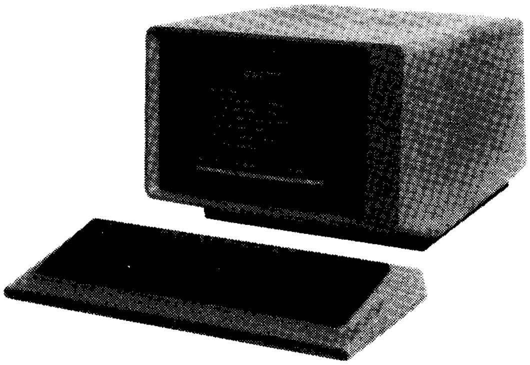
Workstations may be located up to 5000 cable feet (1524 meters) from the processor complex. The controller uses dedicated buffers to allow data transfers at a rate of 9600 bits per second.

### 4.4.2.1. System 80 Workstation

The workstation is designed for ease of operation. Many operating capabilities are included for maximum versatility and access to the system. Indicators inform the operator of the workstation mode of operation (system or workstation), when power is on. They also inform the operator when there are messages waiting to be transferred to the workstation from the system.

Text and system messages are provided on 24 vertical lines. A system status line indicates system status or operating mode. Each line allows up to 80 characters. The display screen is 12 inches (304.8 mm) diagonally. A natural viewing angle prevents glare from ambient light while the operator is seated. Large clear characters permit an excellent view.

# SYSTEM 80 WORKSTATION



## FEATURES

PROTECTED CHARACTERS

REVERSE VIDEO DISPLAY

WORKSTATION PRINTERS

- 80-Column Printer
- 132-Column Printer

## CAPABILITIES

### PROTECTED CHARACTERS

Characters displayed on the screen may be protected so that the operator cannot position the cursor on protected characters. If all character locations are protected, the cursor moves to the home position. Characters to be protected are selected by programming commands. Protected characters are identified on the display by a different intensity or reverse video.

### CURSOR SCAN

The operator can scan the cursor backward, forward, up, or down by pressing appropriate keys on the keyboard. Rapid corrections or changes may be made by entering the desired character location via the shortest path.

### OPERATING MODES

The workstation operator maintains complete control of his operating environment by allowing use of console commands to the operating system without restricting use of the display screen. The operating modes are:

- Workstation Mode

  In normal operation, the workstation mode provides communications to applications programs.

- System Mode

  Communications directly to the OS/3 operating system are made via system mode. Workstation mode is suspended, and the top two lines of the display indicate system information when the particular workstation is in system mode. These two top lines are used for communication with the system. Lines 3 through 24 continue to display workstation mode information. The second line is used by the operator to enter inputs to the system, and the first line displays the system response messages.

  In system mode, the operator uses authorized OS/3 facilities to inquire on system operation, activate particular jobs, and perform other functions useful for operation. After using system mode, the operator may easily return to workstation mode simply by pressing a single key on the keyboard. The display returns to its original workstation status, including the original cursor position.

▷▷▷

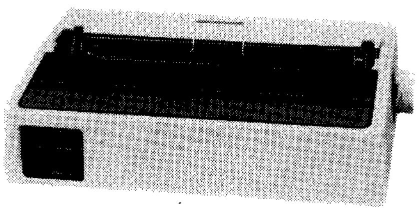
## BLINK MARKER CHARACTERS

Special characters may be selected for blink (delta) display fields. This assists the operator in changing contents of unprotected locations within blink fields. The delta symbols indicate the beginning and end of these fields.

## REVERSE VIDEO DISPLAY

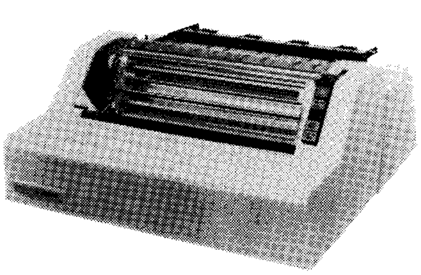
Special areas of a workstation display may be designated by programming for reverse video display. This provides special attention to the operator by highlighting a particular point on the screen that should be noted. Programming of special areas may also be at a lower intensity so that either, but not both, highlighting methods may appear on a display. Using reverse video does not affect other operations of the workstation in either mode.

## WORKSTATION PRINTERS



- 80-Column Printer

  The 80-column printer is an inexpensive printer, lightweight and compact. A tension-loaded printhead forms 7- by 9-dot matrix characters. The printer prints a speeds up to 80 characters per second. The printer is available with a friction-feed platen to print cut sheets or with a pin-feed platen for printing on continuous-feed forms.



- 132-Column Printer

  The 132-column printer maintains the economy of character-by-character serial printing while offering high-speed printing rates of up to 200 characters per second. The same character sets available for the workstation are offered for the 132-character printer. The USA character set is available as a standard 64-character set, a standard 96-character set, and a 96-character set with descenders. The latter includes certain letters (i.e., j, p, and q) formatted so that the tail of the letter descends below the normal print line.

In addition to a standard keyboard arrangement, the workstation contains a number of keys that allow the operator maximum versatility in operations. Table 4-3 lists the keys, and Table 4-4 lists the workstation characteristics.

*Table 4—3. Workstation Keys*

| Key No. | Function | Key No. | Function |
|---|---|---|---|
| 1 | Home Cursor | 18 | Position SOE |
| 2 | Scan Backwards | 19 | Message Waiting |
| 3 | Scan Forward | 20 | Transmit Unprotected |
| 4 | Scan Up | 21 | Transmit Display |
| 5 | Scan Down | 22 | Start Blink Field |
| 6 | Insert In Line | 23 | End Blink Field |
| 7 | Delete In Line | 24 | Start Protect Field |
| 8 | Insert Line | 25 | Stop Protect Field |
| 9 | Delete Line | 26 | Insert in Display |
| 10 | Erase To End of Line | 27 | Delete in Display |
| 11 | Erase Unprotected Display | 28 | Erase Field |
| 12 | Erase Display | 29 | Unlock Keyboard |
| 13 | Lock Keyboard | 30 | Data Message Header Byte (automatic firmware function) |
| 14 | Cursor Return | | |
| 15 | Tabulate | 31 | System Mode |
| 16 | Tab Set | 32 | Workstation Mode |
| 17 | Positioning Cursor | 33 | Message Waiting |

*Table 4—4. Workstation Characteristics*

| Characteristics | |
|---|---|
| Type of display | Cathode ray tube (CRT) |
| Number of display lines | 24 data plus 1 indicator |
| Characters per line | 80 |
| Character size | 0.14-inch (3.5 mm) high,<br>0.07-inch (1.8mm) wide |
| Keyboard arrangements | Three, as follows:<br>■ Typewriter layout<br>■ Typewriter layout with numeric and function pads<br>■ Katakana/English |
| Direct connect communications interface characteristics | ■ Serial<br>■ Asynchronous<br>■ 9600 baud<br>■ Optically isolated |
| Character sets | Nine, as follows:<br>Domestic, United Kingdom, Germany, France, Spain, Denmark/Norway, Sweden/Finland, Italy, Katakana/English |

### 4.4.2.2. Console Workstation

The console workstation is a specially adapted workstation used to perform functions required by the system operator. The display screen and keyboard are similar to the workstation screen and keyboard.

Basically, the console workstation performs all the functions of a standard workstation, as well as functions required by a system operator. An additional function is also included for use by Sperry Univac customer engineers for maintenance purposes. The operator can switch into any one of five modes simply by pressing the appropriate keyboard key.

**CONSOLE WORKSTATION OPERATING MODE FUNCTIONS**

SYSTEM CONSOLE MODE

Provides communications between operator and system software for system console functions.

SYSTEM CONTROL MODE

Provides communications between operator and system control functions, such as for initial microprogram load (IMPL) or initial program load (IPL).

MAINTENANCE MODE

Provides communications between the Sperry Univac customer engineer and system hardware for maintenance-panel functions.

SYSTEM RESPONSE MODE

Allows the operator to communicate with the software operating system, as well as to receive system messages for special operations.

WORKSTATION MODE

Provides standard workstation displays and keyboard functions.

### 4.4.3. Diskette Subsystem

The diskette subsystem, comprised of the diskette controller and integrated diskette drives, is a standard system component. The diskette controller is capable of supporting up to four concurrently operating diskette drives, which may be an intermix of manual or autoload models. A microprocessor in the diskette controller buffers all diskette operations.

The autoload diskette drive allows loading and automatic processing, in sequential order, of up to 20 standard diskettes. Flexible recording selections facilitate data interchange and allow efficient recording capacity. The operator only needs to place 20 diskettes into the hopper and remove them from the stacker after use.

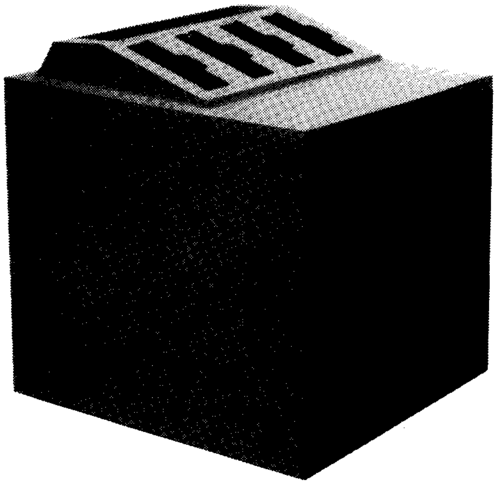
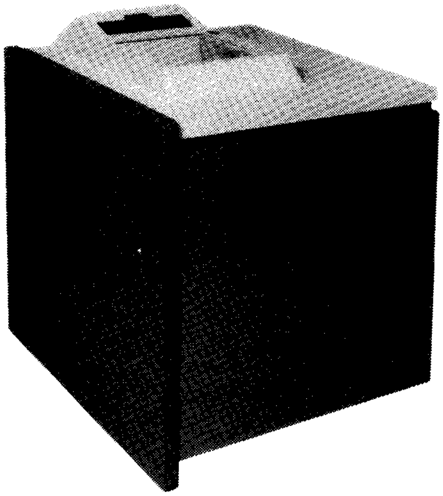
Flexibility in the selection of recording methods allows:

- IBM compatible basic data exchange (BDE) format (with 128 bytes per sector) for data interchange, having a capacity of 256K bytes

- IBM compatible, double-sided, double density (with 256 or 512 bytes per sector), having a capacity of 1 MB.

The manual and autoload diskette drives may be intermixed as required for flexible use. There may be one, two, or four manual diskette drives configured in a system or one or two autoload diskette drives. An autoload diskette drive can be configured with an additional manual drive.

DISKETTE SUBSYSTEMS

| MANUAL LOAD<br>DISKETTE DRIVE CABINET<br>WITH FOUR DISKETTES | AUTOLOAD<br>DISKETTE DRIVE<br>WITH MANUAL DISKETTE |
|---|---|
|  |  |

To summarize, operation of the diskette subsystem is facilitated by:

- the intermix of diskette drives of different recording modes;
- the capability of configuring up to four independent diskette drives for system operating flexibility;
- the automatic detection by the system of the diskette drive recording mode; and
- an interchange of data between data entry systems and other computer systems.

Table 4–5 lists the diskette subsystem characteristics.

*Table 4—5. Diskette Subsystem Characteristics*

| Characteristics | |
|---|---|
| Drive mechanisms per cabinet | 1 autoload and 1 manual load, or 4 manual load |
| Data read/write heads per drive | 2 |
| Access time (ms):<br>Track-to-track seek<br>Seek settling<br>Head loading | <br>3<br>15<br>35 |
| Physical tracks per diskette | 77 |
| Rotational speed (rpm) | 360 |
| Number of sectors per track | Dependent on density method |
| Bytes per sector | 128, 256, or 512 dependent on density method |
| Data transfer rates (kB/s) | 31 @ 256 bytes/sector or;<br>62 @ 512 bytes/sector, dependent on sector sequence arrangement |
| Density (bpi) | 3400 or 6800 (8704 or 17,408 bpcm) |
| Storage capacity per diskette | Recording is on one or both sides of the diskette, depending on the version used.<br>■ Double density, 2 sides approximately 1 MB<br>■ Double density, 1 side approximately 500 KB<br>■ Single density, 2 sides approximately 500 KB<br>■ Single density, 1 side approximately 250 KB |
| Data format | The standard data character is an 8-bit byte and can be any code.<br>EBCDIC should be used for IBM Basic Data Exchange compatibility. |
| Data processing compatibility: | Bytes/sector Sectors/track |
| ■ SPERRY UNIVAC | 256 26<br>512 15 |
| ■ IBM | 128 26<br>256 15<br>512 8 |
| Data access mode (DAM) | DAM provides the user with unrestricted access and operation with the mounted diskette. |
| Data set mode (DSM) | DSM allows the user to process data arranged in a file structure. |
| Data set label checking | Checks certain fields of a data set label and declares invalid labels when certain conditions exist. |
| Autoload hopper | Holds 20 diskettes maximum |
| Autoload stacker | Holds 20 diskettes maximum<br>The load or unload time per diskette is:<br>■ 5 seconds maximum (60 Hz model)<br>■ 6 seconds maximum (50 Hz model) |

LEGEND:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| bpcm | = | bytes per centimeter | KB | = | kilobytes | ms | = | milliseconds |
| bpi | = | bytes per inch | kB/s | = | kilobytes per second | rpm | = | revolutions per minute |
| Hz | = | Hertz (cycles) | MB | = | megabytes | | | |

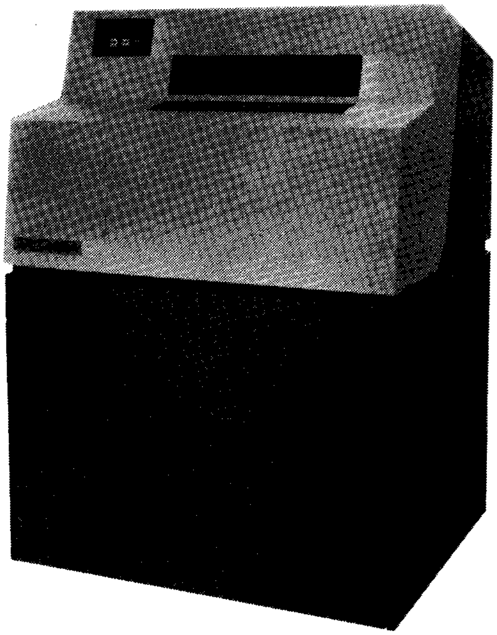
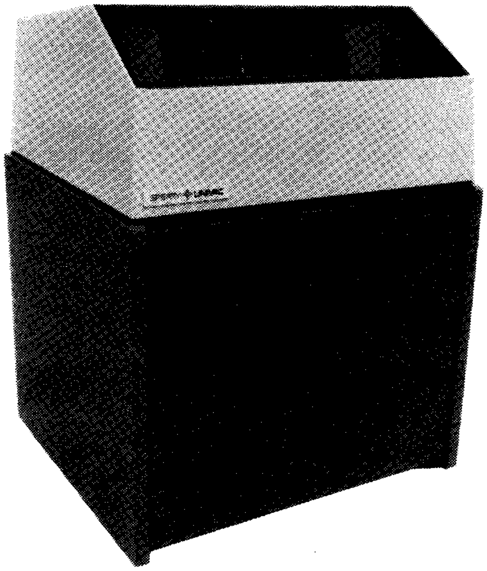
### 4.4.4. Printer Subsystems

Printer subsystems are connected to the system via a paper peripheral controller or a remote printer controller. Two local printers may operate from a single paper peripheral controller, provided their combined printing capacity does not exceed 1500 lines per minute (lpm). One printer, located up to 5000 cable feet away, may be operated from a remote printer controller.

Both line printer types include operator convenience capabilities and allow form-format control to be directed by programming via the vertical format buffer. Line spacing may be at 6 or 8 lines per inch (lpi), with 132 or 136 characters per line.

Form sets used on one printer or model may not necessarily produce acceptable print quality results when used on another printer or different model of the same printer. Adequate testing should be made under operating conditions. Printing of up to 6-part forms is possible, but multiple-part forms over four parts should be tested to verify the acceptability of the results.

**PRINTER SUBSYSTEMS**

| HIGH PERFORMANCE LINE PRINTER | LINE PRINTER |
|---|---|
|  |  |

### 4.4.4.1. High Performance Line Printer

The high performance line printer is a freestanding impact line printer. It employs a horizontal-moving print band in a print cartridge assembly that permits easy interchangeability by the operator. The print characters are etched on the print band and arranged in arrays (type fonts) repeated around the periphery of the print band.

The print throughput rate of the high performance line printer is dependent on the number of lines spaced or skipped and the arrangement of characters in the arrays on the print band. For example, a print band containing 3 arrays, with 128 characters in each array, prints 560 lpm; whereas, a print band containing 16 arrays with 24 characters (nonrepeated) in each array prints 1200 lpm. The time between starts of consecutive print operations or line feeds is limited to a minimum of 48 milliseconds. The 560- and 1200-lpm rates are the minimum/maximum for single line spacing at 6 lpi.

The high performance line printer can only be connected to System 80 via a paper peripheral controller. Table 4–6 lists the high performance line printer characteristics.

### 4.4.4.2. Line Printer

The line printer is a freestanding impact line printer. It employs a horizontal-moving print band in a print cartridge assembly that permits easy cartridge interchangeability by the operator. The print characters are etched on the print band and arranged in arrays (type fonts) repeated around the periphery of the print band.

Three printer speeds are available with the line printer: 180, 300, and 640 lpm. The exact print speed is determined by the band speed, paper advance time, data transfer rate, the number of times an individual character appears on a band, and the character density (characters/inch) at which printing takes place.

The line printer can only be connected to System 80 via a paper peripheral or remote printer controller. Table 4–6 lists the line printer characteristics.

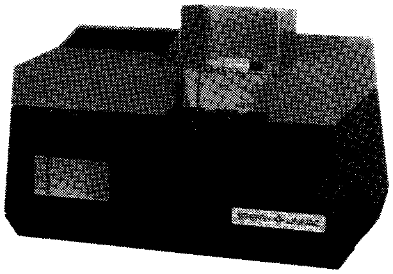
*Table 4—6. Printer Subsystem Characteristics*

| | High Performance Line Printer | | | Line Printer | | | | |
|---|---|---|---|---|---|---|---|---|
| **Characteristics** | | | | | | | | |
| Print speed | 210 to 1250 lpm, depending on character contingencies | | | 180, 300, and 640 lpm, depending on printer selection | | | | |
| | Available character sets | Number of sets per band | Normal print rate (lpm) | Available character sets | Number of sets per band | Maximum print rate | | |
| | | | | | | 180 lpm | 300 lpm | 640 lpm |
| | 384 | 1 | 210 | 48 | 4 (plus 16 characters) | 188 | 306 | 640 |
| | 192 | 2 | 395 | | | | | |
| | 128 | 3 | 560 | 64 | 3 (plus 16 characters) | 163 | 255 | 605 |
| | 96 | 4 | 710 | | | | | |
| | 64 | 6 | 980 | 96 | 2 (plus 16 characters) | 137 | 191 | 452 |
| | 48 | 8 | 1200 | | | | | |
| | 32 | 12 | 1250 | 128 | 1 (plus 80 characters | 89 | 109 | 360 |
| | 24 | 16 | 1250 | | | | | |
| Line advance timing | Advance and print | Time (ms) | | Advance and print | Time (ms) | | | |
| | | 6 lpi | 8 lpi | | 6 lpi | | 8 lpi | |
| | 1 | 16.7 | 14.1 | 1 line | 40 | | 40 | |
| | 2 | 24.6 | 20.9 | 2 lines | 52 | | 52 | |
| | 3 | 30.9 | 25.9 | 3 lines | 64 | | 64 | |
| | 4 | 35.0 | 30.9 | n+1 lines | 76+12 | | 76+12 | |
| | 5 | 38.9 | 34.1 | | | | | |
| | 6 | 42.6 | 37.1 | | | | | |
| | 7 | 45.9 | 39.9 | | | | | |
| | 8 | 49.3 | 42.6 | | | | | |
| Number of print positions | 136 standard (columns) | | | 132 standard (columns) | | | | |
| Form advance control | Vertical format buffer | | | Vertical format buffer | | | | |
| Form advance rate | 50 inches (127 cm) per second | | | 15 inches (38.1 cm) per second | | | | |
| Form dimensions | 4 to 18.75 inches (10.16 to 47.62 cm) wide, 1 to 18 inches (45.72 cm) long | | | 3 to 15 inches (7.62 to 38.10 cm) wide, 1 to 22 inches (55.88 cm) long | | | | |
| Horizontal spacing | 10 characters per inch | | | 10 characters per inch | | | | |
| Vertical spacing | 6 or 8 lpi, operator selectable | | | 6 or 8 lpi, operator selectable | | | | |

### 4.4.5. Card Processor Subsystems

The card readers and card punch are controlled by the paper peripheral controller. All I/O activity is controlled via the PPC. The PPC processes all the I/O requests directed to these devices. The information in these control elements is used and is completely responsible for all command interpretations, data transfers, status presentations, and error reporting sequences. This relieves main storage of addressing functions, maintenance of I/O control elements, and status reporting to the processor.

**CARD PROCESSORS**

| CARD READER | CARD PUNCH |
|---|---|
|  | |

### 4.4.5.1. Card Reader

The card reader is a table-top device capable of reading 80-column cards at the rate of 300 cards per minute (cpm). The hopper and stacker can store 1000 cards each. The card reader is connected to the host system through the paper peripheral controller. Characteristics of the card reader are listed in Table 4-7.

### 4.4.5.2. Card Punch

The card punch is a freestanding device capable of punching 80-column cards at a rate of 75 cpm, or at a rate of 160 cpm when punching only the first 28 columns of the card. The hopper holds up to 700 standard-thickness cards. The primary stacker, with a capacity of 700 cards, accepts all correctly punched cards. The auxiliary stacker, with a capacity of 100 cards, accepts cards in which errors were detected. Table 4-8 lists the card punch characteristics.

*Table 4—7. Card Reader Characteristics*

| Characteristics | |
|---|---|
| Card orientation | Face down, column 1 to left and row 9 facing away (80, 66, and 51-column cards) |
| Card rate (cpm) | 300 |
| Read technique | Two columns of photosensitive sensors and light-emitting diodes. Dual redundant column amplifier checking. |
| Read modes | Image mode: 160-six bit characters per card.<br>Translate mode: 80 characters per card |
| Read stating sensing | Column by column |
| Hopper capacity (cards) | 1000 |
| Stacker capacity (cards) | 1000 |

*Table 4—8. Card Punch Characteristics*

| Characteristics | |
|---|---|
| Punch speed | 75 (full columns)<br>160 (punch first 28 columns)<br>120 columns per second advance speed |
| Card punch code | Image or compressed |
| Punch cycle time (ms) | 16.6 |
| Card stock | 80-column standard-thickness |
| Punch mode | 2-column serial |
| Checking mode | Punch motion check |
| Feeding mode | Column serial, on demand |
| Hopper capacity (cards) | 700 |
| Stacker capacity (cards) | 700 (primary stacker)<br>100 (auxiliary stacker) |
| Card read/ validity check | Reads prepunched cards @ 160 cpm |

### 4.4.6. UNISERVO 10 Magnetic Tape Subsystem

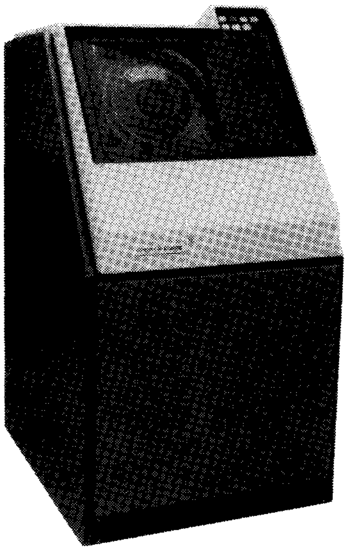
The UNISERVO 10 Magnetic Tape Subsystem is a low-cost unit using magnetic tape conforming to standards specified by the American National Standards Institute (ANSI). This subsystem is provided in two basic versions, with power requirement variations in each version to accommodate individual operating needs. The UNISERVO 10 can operate with industry-compatible magnetic tapes in any version or variation.

The 9-track tape version operates with phase-encoded (PE) recording format at 1600 bits per inch (bpi) (630 bits per centimeter, bpcm). Reading or writing on tape is at a speed of 25 inches per second (ips).

The 9-track NRZI version operates with a density of 800 bpi.

The 7-track version operates with NRZI format and a density of 800, 556, or 200 bpi.

**UNISERVO 10 MAGNETIC TAPE SUBSYSTEM**



**CAPABILITIES**

▷ PERFORMANCE

At a tape speed of 25 ips (63.3 centimeters per second, cmps), it has a throughput rate of up to 40 kilobits with PE format and up to 20 kilobits with NRZI format. When 9-track recording is used, these rates provide the throughput needed in normal data processing.

▷ AUTOMATIC TAPE LOADING

The UNISERVO 10 provides automatic tape loading, which simplifies operator setup or other intervention. This capability is present whether or not the wraparound tape cartridge, which is standard in the industry, is used.

▷ WRAPAROUND TAPE CARTRIDGE

The ANSI-standard, industry-compatible wraparound tape cartridge can be used in the UNISERVO 10. Besides enhancing automatic tape loading, the cartridge also serves as a dust cover for the tape reel while in storage. The operator no longer requires time to manipulate tape in and out of its container.

The UNISERVO 10 is attached to the central processor via a special interface controller port located in the processor complex. Table 4–9 lists the UNISERVO 10 characteristics.

*Table 4—9. UNISERVO 10 Characteristics*

| Characteristics | |
|---|---|
| Tape units per subsystem | 1 to 8 |
| Tape speed | 25 ips (63.5 cmps) |
| Recording mode | ■ 9-track PE<br>■ 9-track NRZI<br>■ 7-track NRZI |
| Recording density | ■ 1600 bpi (630 bpcm) on 9-track PE<br>■ 800 bpi (315 bpcm) on 9-track NRZI<br>■ 200, 556, 800 bpi on 7-track NRZI |
| Read access time | 17 milliseconds (ms) at 25 ips (63.5 cmps) on 9-track tape having 0.60-inch (1.524 cm) interblock gap |
| Start/stop time | Addressed during stop cycle:<br>■ 17 ms at 25 ips (63.5 cmps) without reversed tape direction; or<br>■ 33 ms at 25 ips (63.5 cmps) with reversed tape direction |
| Nonstop interblock gap passing time | 24 ms at 25 ips (63.5 cmps) on 9-track tape |
| Reversal time | Tape unit conditioning for tape motion reversal is 16 ms at 25 ips (63.5 cmps) |
| Rewind time | 180 seconds (maximum) for 2400 feet (731.5 m) of tape, or 200 ips (508 cmps) |
| Data transfer rate | ■ 40 kB/s* on 9-track PE<br>■ 20 kB/s on 9-track NRZI<br>■ 5 kB/s on 7-track, 200 bpi NRZI |
| Tape runaway protection | Load operation terminated if beginning of-tape (BOT) marker is not sensed within 24 seconds after LOAD POINT switch is pressed. Another load sequence is not initiated automatically by the tape unit, even with a tape cartridge. |
| Tape reel size | 10.5 inches (26.67 cm) diameter |
| Tape type | ANSI X3.40-1973, 0.498-inch (1.264 cm) wide, 2400 feet (731.5 m) long. |

*kB/s = kilobytes per second

### 4.4.7. Communications Facilities

Use of workstations and other communications devices is implemented through the communications attachment in the central processor complex. Two communications lines are provided in the basic system, and up to eight lines can be used in a system containing an input/output microprocessor. The communications lines can be used for many services, including:

**COMMUNICATIONS LINES**

- Digital data network for distributed processing
- Public telephone networks
- Private leased lines
- Automatic answering
- Telegraph
- Military communication networks

Communications line characteristics are determined by selection of an appropriate single line communications adapter (SLCA). Each line can accept devices that operate at speeds up to 9600 bits per second (bps). Communications devices attached to the line are controlled by the SLCA, which interfaces with the communications equipment and provides the data path. Other control functions, such as special character recognition, integrity checks, and data transfer control between main storage and device, are also performed by the SLCA. Communications terminals and workstations basically use the same software to assure availability of easy access to the OS/3 operating system.

Table 4–10 lists the basic protocols used with the communications attachment. Communications devices available from Sperry Univac that may be attached to the system communications lines are briefly described in subsequent subsections.

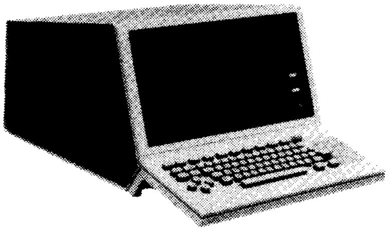
*Table 4—10. Communications Functionality*

| Protocol/Function | Line Speed | Mode | Interface |
|---|---|---|---|
| UNISCOPE 100/200,<br>UTS 400, UTS 4000 | 2000 - 9600 bps | Half duplex, synchronous | RS-232-C or MIL-188-100 or UNISCOPE and UTS 400 only |
| BSC | 2000 - 9600 bps | Half duplex, synchronous | RS-232-C or MIL-188-100 |
| UTS 10/TTY (or equivalent) | 50 - 9600 bps | Half duplex, asynchronous | RS–232–C or MIL–188–100 |
| Universal Data Link Control (UDLC) | 9000 - 9600 bps | Half duplex or full duplex, synchronous | RS-232-C/X.21. BIS |
| NTR | 4800 | Full duplex, synchronous | |

### 4.4.7.1. UNISCOPE Display Terminals

The UNISCOPE 100/200 Display Terminal is a 2-way remote terminal device that is keyboard-operated and contains a cathode-ray tube for message display.

**UNISCOPE DISPLAY TERMINAL**



▶ CATHODE-RAY TUBE

Displays:

- System output messages
- Operator input messages for composition and editing before transmission to the system

▶ KEYBOARD

Includes:

- Alphanumeric keys
- cursor control
- editing function keys

Each character entered by the operator is immediately displayed and stored in the display terminal hardware. Characters are generated by a read-only storage digital stroke generator.

The terminal provides input/output message buffering, refresh storage, character generation, and control logic. Special interfaces for hardcopy output are available. A variety of presentation formats provides display capacities of 960 or 1024 characters for the UNISCOPE 100 Display Terminal, or 1536 or 1920 characters for the UNISCOPE 200 Display Terminal. The terminal can be used as a data entry device or as a display device.

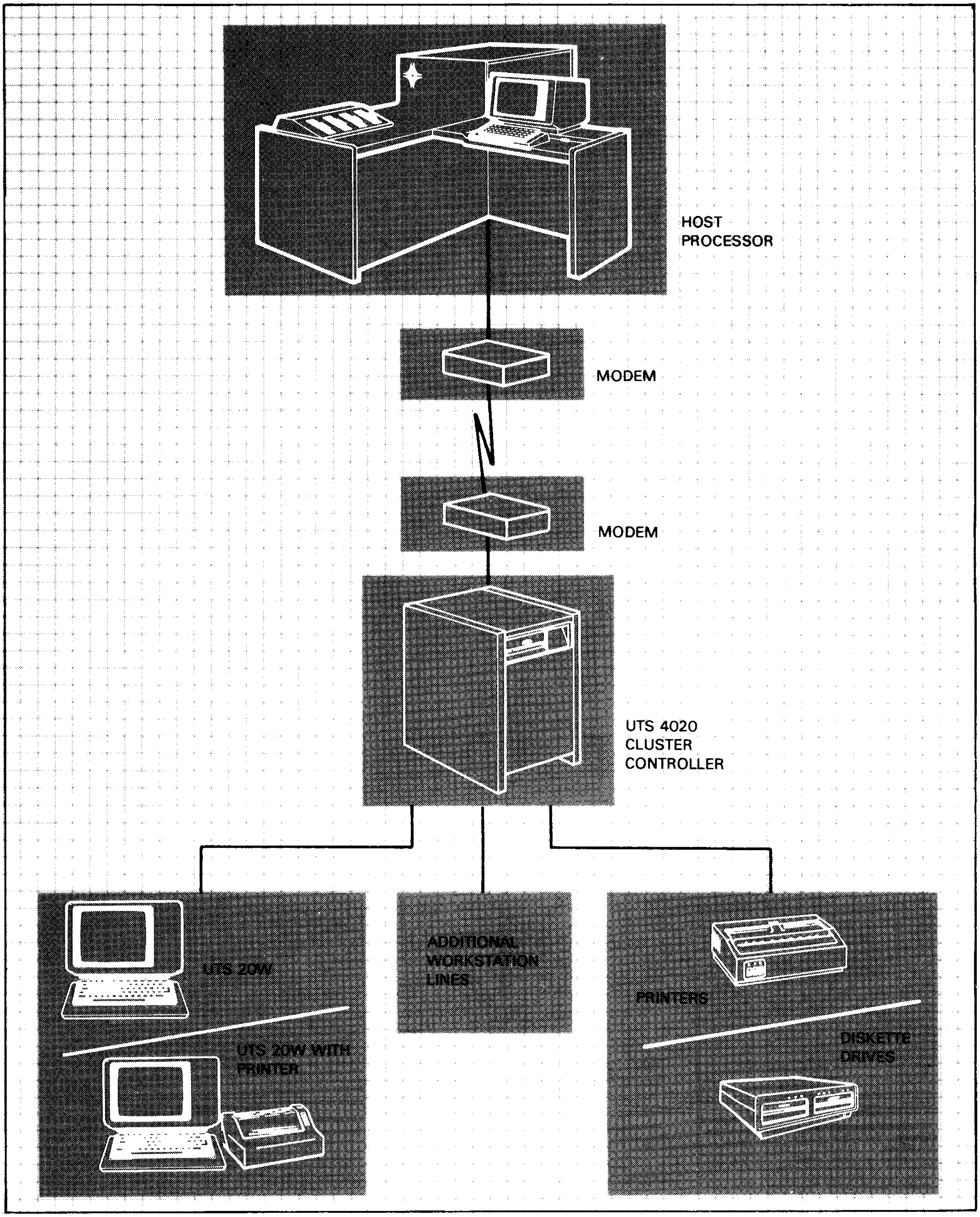
### 4.4.7.2. UTS 4000 Universal Terminal System

The SPERRY UNIVAC UTS 4000 terminal system is a comprehensive family of communications devices that offers a wide range of interactive communications capabilities. The UTS 4000 system can be used for a variety of applications, from simple data entry and retrieval to sophisticated distributed data processing. Through use of interchangeable components, the system can be configured to meet the needs of any communications user.

The UTS 4000 system employs a terminal cluster architecture using a programmable controller unit attached to the host processor. Terminals and other peripheral devices are connected through the controller. All available terminal types are video display devices and include editing and programmable terminals. Peripheral devices include a desk-top diskette unit supporting 1-megabyte diskettes and a variety of terminal printers.

The UTS 4000 control unit, the UTS 4020, is a programmable, microprocessor-based unit capable of acting as a stand-alone processor as well as a communications controller. Provided with an integrated diskette unit, random access storage, and a complement of software elements, it is one of the most advanced communications devices available. The stand-alone processing capabilities of the UTS 4020, coupled with its advanced communications features, make the UTS 4000 system ideally suited for inclusion in a distributed data processing network.

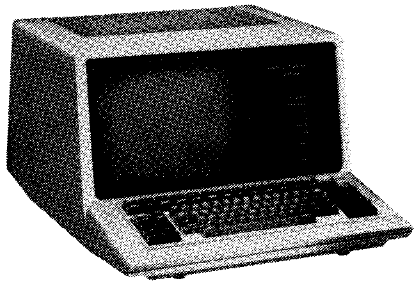
## TYPICAL UTS 4000 TERMINAL SYSTEM CLUSTER



### 4.4.7.3. UTS 400 Universal Terminal System

The UTS 400 Universal Terminal System is a powerful, programmable terminal display system.

**UTS 400 UNIVERSAL TERMINAL SYSTEM**



PROVIDES:

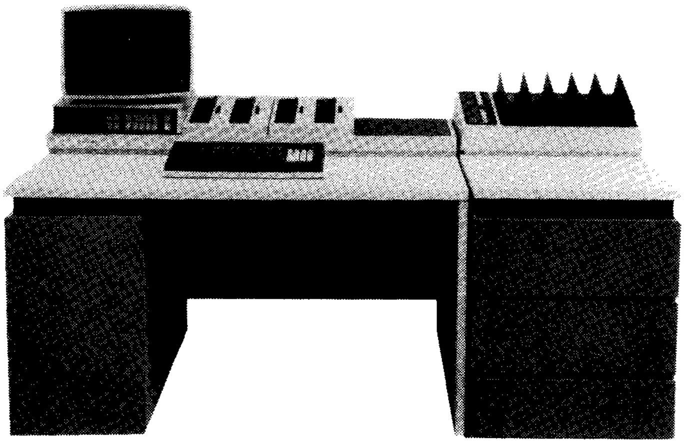
- Input/output message buffering
- Character generation
- Control logic

A variety of presentation formats is offered, that provide a total display capacity of 960, 1024, 1920, or 1536 characters. Due to its modular construction, the UTS 400 can operate as a data entry device or as a display device and can be conveniently located at the central computer site or at a remote station. Various configurations are possible. The UTS 400 can be configured as a master terminal with up to two slave terminals or as a controller with up to six slave terminals. It can be used with the UNISCOPE 100 or UNISCOPE 200 Display Terminals in multidrop or multiplexer configurations.

### 4.4.7.4. BC/7 Business Computer System

The BC/7 Business Computer System is a low-cost, powerful terminal system built into a desk configuration.

**BC/7 BUSINESS COMPUTER SYSTEM**



FEATURES:

- LSI circuitry and MOS technology throughout
- Sophisticated operating system to complement hardware

Together, they form a flexible, easy-to-use terminal system that operates as a stand-alone batch system or a remote batch terminal.

### 4.4.7.5. Additional Communications Devices

In addition to the SPERRY UNIVAC devices listed, the following devices can also be attached as communications terminals:

- IBM 3270 terminal system

- TELETYPE Models 33, 35, 37, and 38

While these devices can be attached as communications terminals, all services available to the normal communications user may not be available to users of these devices.

## 4.5. HARDWARE SUMMARY

Table 4-11 summarizes System 80 hardware characteristics.

*Table 4—11. System Characteristics (Part 1 of 2)*

| Characteristics | |
|---|---|
| System orientation | Disk |
| Central processing unit | ■ Versatile instruction set under microprogram control<br>■ Floating point<br>■ Interval timer<br>■ Up to 14 jobs<br>■ Dynamic storage relocation, including I/O<br>■ Eight interrupt levels |
| Main storage capacity (kilobytes) | 262, 524, 786, or 1048 |
| Main storage performance | A cycle time of 400 nanoseconds for a 4-byte access |
| Data organization | 8-bit bytes; 4 bytes per main storage access |
| Registers | ■ 16 control (4 bytes each)<br>■ 16 supervisor (4 bytes each)<br>■ 16 problem (4 bytes each)<br>■ 4 floating point (8 bytes each) |
| Control storage | 16K words, 32 or 64 (Model 5) bits wide plus 4 parity bits |
| Average cycle time | 180 nanoseconds |
| System architecture | ECL with LSI circuits |
| Internal control storage data path width | 8 bytes |

*Table 4—11. System Characteristics (Part 2 of 2)*

| Characteristics | |
|---|---|
| Aggregate system data rate | 6.0 MB/s* |
| Control method | Microprogramming |
| Communications attachment | Up to 8 communications lines |
| Disk channel/controller | ■ Controls up to 8 disk drives<br>■ Data transfer rate up to 1.2 MB/s<br>■ Burst mode of operation |
| Workstation controller | Controls console workstation and up to 7 workstations. Additional workstation controllers can be used to control 8 additional workstations each, to a maximum of 39 total workstations and one console/workstation. |
| Console workstation | ■ Monitors and controls system operation<br>■ Provides keyboard and visual display screen<br>■ Keyboard up to 96 characters<br>■ Display screen of 24 lines, 80 characters per line, plus one status/mode indicator line. |
| Diskette controller | ■ Up to 4 diskette drives; manual or autoload drives with capacities of 256K byte or 1 MB<br>■ Date rate of 31 kB/s** at 128 bytes/sector, or 62 kB/s at 256 or 512 bytes/sector |
| Paper peripheral controller | ■ Multiplex mode<br>■ Maximum aggregate data transfer rate up to 200 kB/s<br>■ Up to 2 printers with a combined print capacity of up to 1500 lines/minute<br>■ Up to 2 card readers; or 1 card reader and 1 card punch |
| Remote printer interface | One additional printer per interface |

* MB/s = megabytes per second

** kB/s = kilobytes per second