3280 AND Micro3200 PRODUCT OVERVIEW

50-045 R00

Disclaimer

The information contained in this document is subject to change without notice. Concurrent Computer Corporation has taken efforts to remove errors from this document, however, Concurrent Computer Corporation's only liability regarding errors that may still exist is to correct said errors upon their being made known to Concurrent Computer Corporation.

License

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include all copyright notice, trademarks, or other legends or credits of Concurrent Computer Corporation and/or its suppliers. Title to and ownership of the described software and any copies thereof shall remain in Concurrent Computer Corporation and/or its suppliers.

The licensed program described herein may contain certain encryptions or other devices which may prevent or detect unauthorized use of the Licensed Software. Temporary use permitted by the terms of the License Agreement may require assistance from Concurrent Computer Corporation.

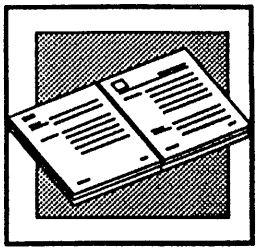
The hardware description contained in this document is intended solely for use in operation, installation, maintenance, or repair of Concurrent Computer Corporation equipment. Use of this document for all other purposes, without prior written approval from Concurrent Computer Corporation is prohibited.

Concurrent Computer Corporation assumes no responsibility for the use or reliability of this software if used on equipment that is not supplied by Concurrent Computer Corporation.

Acknowledgments

Concurrent Computer Corporation
106 Apple Street
Tinton Falls, NJ 07724

Printed in the United States of America



About this Book

Overview

This manual introduces the systems that comprise the 3280 and Micro3200 Families, including the supported software. The text emphasizes the features and capabilities of the systems rather than the in–depth technical details.

Before you Start

This manual provides an overview of the 3280 and Micro3200 systems. Although a general understanding of computers may be helpful, such knowledge is not necessary to understand most of the manual's content.

Using this Book

This manual may be read sequentially, from beginning to end, or in a random manner.

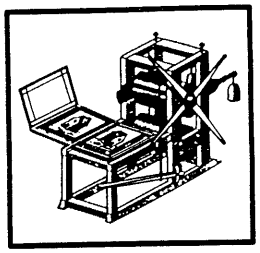
Document Organization

- Chapter 1 offers a broad overview of the 3280 and Micro3200 systems and discusses some of their applications.
- Chapter 2 discusses the system software offerings and provides brief functional overviews of each.
- Chapter 3 describes the functions, capabilities, and where applicable, the respective performance metrics of various system hardware elements.
- Chapter 4 discusses the systems individually, and for each system provides a functional synopsis and physical profile.
- Chapter 5 profiles some of the services that we offer, specifically, Customer Service, Training, and the System Products Division.

- Appendix A lists the document titles for the products discussed in this overview.

Other Sources of Information

This manual discusses a broad range of topics that can be further investigated using the related documentation listed in Appendix A.



Revision History

Manual	Revision	Release Date
50-045	R00	August 1989

How Can I Track Changes?

Each time this manual is updated or reissued, it is added to the chart above. This chart includes the manual number, revision, and release date. It helps to track all issues of this manual. If further assistance is needed, call your local sales representative.

How Are Changes Shown?

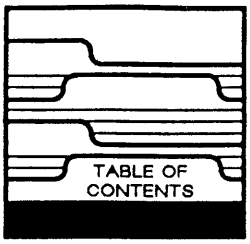
Changes to the documentation occur when there are product changes. This causes the document to be reissued or updated (the RXX number changes). The RXX number is located at the bottom of each page. Technical changes within the document are indicated by a vertical bar (|) shown in the right hand margin.

Updates

Update packages are issued between reissues and contain replacement and additional pages that are merged into the manual by you. For each engineering change notice (ECN) that affects the manual, an update package is issued and the RXX number is increased by one. (For example, if the manual was released at R00, the first update package that was issued would receive an R01.)

Reissues

Each time this document is reissued, it is replaced in its entirety, and the RXX number is increased by the number of ECNs against the manual. (For example, if the first release of the document was R00, the reissue number would be the number of ECNs issued against the manual since the first release.)



Contents

Conventions ix

Chapters

1 The 3280 and Micro3200 Families

Applications for Your System 2
- Real–Time Processing 2
- On–Line Transaction Processing 3
- Development Environment 3
- Data Communications 3

Modularity 4
Software Compatibility 5
Availability 5
- Data Protection 6

2 System Software

Using OS/32 8
- Program Development 9
- Task Scheduling 10

Programming Languages 12
- System Support Run–Time Libraries (RTLs) 13
- FORTRAN VII 14
- Interfacing FORTRAN VII and Assembly Language 15
- C^3Ada 15
- COBOL Development Environment (CoDE) 16
- Other Software Languages 17

Scheduling Tasks For Real–Time Applications 17
- Multitask Model 18
- Consolidation Model 20

Business Applications For OS/32 22
- Database Integrity 23
- Securing Your Database 23
- Multiprocessing and Reliance PLUS 23
- Reliance Office 25

Chapters (Continued)

3 System Organization

Hardware Overview 28
Processing 29
Processor Hardware 29
Pipelining Instructions 30
Cache 32
Instruction and Data Cache 34
Instruction Prefetch 34
Parallel Multiplier 34
Register Allocation 34
Control Store 35
Real–Time Clock 35
Instruction Set 36
Floating Point 37
Other Processor Features 37
System Buses 37
Power Subsystem 39
Power Distribution and Fail Soft Considerations 40
System Memory 41
Memory Modules 41
Input/Output (I/O) System 43
I/O Configurations 44
Mass Storage 44
Distributed System Bus Architectures 45
Expanded Configurations for Single Systems 46
Shared and Shadowed Memory 47
Control/Diagnostic System (CDS) 48

4 System Specifics and Configurations

The Micro3200 Compact Systems 52
System Organization 54
Physical Profile 56
Peripherals 58

Chapters (Continued)

The Micro3200 Expanded System 59
System Organization 61
Physical Profile 62
The Micro3200 MPS Multiprocessor 65
System Organization 67
Physical Profile 68
The 3280MPS Multiprocessor 70
System Organization 72
Physical Profile 73
The 3280E MPS Multiprocessor 75
System Organization 77
Physical Profile 79

5 Concurrent Services

Customer Service 84
Understanding Your System 85
Custom Engineering 85

A System Documentation

OS/32 Documentation 87
OS/32 – System Operations 87
Operator Reference 88
System Utilities 88
General Utilities 88
OS/32 – System Level Programming 89
OS/32 – Application Level Programming 90
OS/32 – Data Communications 90
OS/32 Languages – FORTRAN (Package) 90
OS/32 Languages – C3Ada 91
C3Ada Debugging and Tuning Tools 91
C3Ada Compiler and APSE Tools 91
OS/32 Languages – C (Package) 92
OS/32 Languages – CAL 92
OS/32 Languages – Other High Level 92
Reliance Plus Documentation 93
Reliance Builder Documentation 94
Hardware Documentation 95

Figures

4–1 Micro3200 Compact Systems 52
4–2 Micro3200 Compact Systems Block Diagram 54
4–3 Micro3200 Expanded System 59
4–4 Micro3200 Expanded System Block Diagram 61
4–5 Micro3200 MPS 65
4–6 Micro3200 MPS Block Diagram 67
4–7 3280MPS 70
4–8 3280MPS Block Diagram 72
4–9 3280E MPS 75
4–10 3280E MPS Block Diagram 77

Index In–1

Conventions

Conventions

The following conventions are used throughout this guide.

Type Style

Constant Width Type

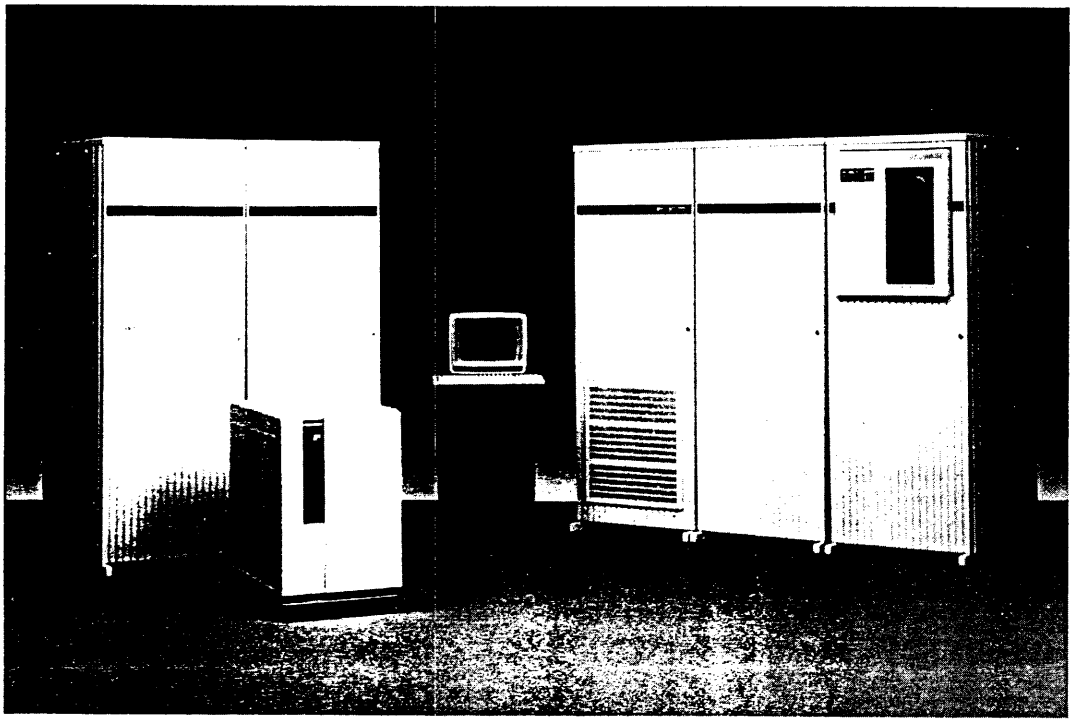
`Constant width type` is used to show source code segments.

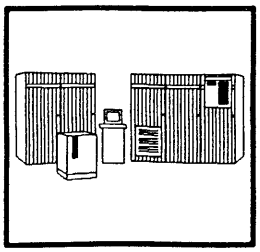
Italic Type

Italic type is used to cite references to manuals, to chapters, or to section titles in the manual.

Bold type

Bold type is used for emphasis.





1

The 3280 and Micro3200 Families

In this chapter

We discuss some of the features and characteristics of Concurrent Computer Corporation's (Concurrent) 3280 and Micro3200 systems.

Topics include:

- application profiles,
- system modularity,
- software compatibilty, and
- system availability.

Concurrent's 3280 and Micro3200 super–minicomputers offer a diverse range of computational performance and I/O throughput. From the uniprocessor Micro3200 Compact Systems to the 3280E MPS multiprocessor, these processors satisfy a wide spectrum of needs and applications.

All 3280 and Micro3200 systems feature:

- modular building block architectures,
- software compatibility between 3280 and Micro3200 machines, and their Series 3200 predecessors, and
- high system availability.

Combined, these attributes produce a series of processors which avail themselves to a diversity of applications such as real–time processing (including on–line transaction processing), program development, and data communications.

Applications for Your System

The flexibility of your 3280 and Micro3200 system allows you to tailor a system that closely matches your application's requirements. For example, if your application needs extensive number crunching capabilities and relatively minimal I/O throughput, you can configure your system to deliver the required processing power without the unnecessary I/O bandwidth. This flexibility makes these processing systems suitable for a wide variety of applications, some of which are discussed in the following sections.

Real–Time Processing

Real–time processing presents some of the most pressing demands on a computer system's resources. These systems must perform their functions within predetermined time spans. The amount of time depends on the criteria established by the application designer(s). For many applications such as satellite data acquisition and avionics simulation environments, the time constraints and interrupt response requirements impose enormous demands on computer systems. Because deficiencies in either aspect can at the least result in erroneous results, system builders require real–time systems with proven reputations.

With our proprietary OS/32 operating system, 3280, and Micro3200, processors are known for their real–time capabilities. Their tightly–coupled processor–memory architecture assures rapid task–to–task communication both within and between computer systems. Furthermore, the multilevel I/O interrupt structure with dedicated register sets substantially reduces context switching for fast I/O responsiveness.

On–Line Transaction Processing

Concurrent offers solutions for those whose real–time needs are in on–line transaction processing. Coupled with our OS/32 operating system and Reliance PLUS™ software, our Series 3200 systems already have a proven track record in the gaming, banking, financial brokerage services, hospital, and health care arenas.

With OS/32 and Reliance PLUS, you have a relational database that provides not only responsive database queries and updates, but simplified report–generation, user access security, transparent communication protocols, and high system availability.

Together, the responsiveness of Reliance PLUS software and the high I/O capacity of our machines yield a debit/credit benchmark in excess of 15 transactions/sec*. When combined with life–cycle costs, our systems offer an extremely cost–effective on–line transaction processing (OLTP) solution.

Development Environment

One general requirement of these application areas is the ability to service a large number of interactive users. Our OS/32 operating system and multi– terminal monitor (MTM), provide an ideal multi–user, timesharing environment. MTM supports up to 64 interactive users and a maximum of 65,535 user accounts, each with password protection.

Data Communications

Regardless of the application (real–time, OLTP, or general development), it is likely that you may want to move data not only within, but between computers and computer complexes. This is where the need for data communications arises.

Reliance PLUS is a trademark of Concurrent Computer Corporation
*Benchmark result of 3280MPS with 1 APU

To answer this need, Concurrent offers RTnet™, a comprehensive suite of data communications products that:

– interconnect your systems through packet–switching networks via the CCITT X.25 standard,

– interconnect remote terminals with the X.29 standard,

– connect to LANs using the IEEE 802.3 (ETHERNET®) protocol, and

– connect to personal computers.

Furthermore, with our IBM® Gateway products, you can gain the complementary strengths of both Concurrent and IBM by connecting your Series 3200 system to IBM mainframes and other makes.

All IBM Gateway products are transparent to the user.

Modularity

One of the key properties common to all 3280 and Micro3200 systems is their inherent modularity. These systems employ a building block philosophy that allows you to expand your current system if your computational needs increase in the future. Therefore, you can start with more modest configurations without the apprehension of expensive upgrades or replacements should unanticipated computing demands arise.

The level of modularity that these systems possess offers you numerous ways in which to tailor a system that delivers only the necessary performance parameters. Depending on the system, all major functions of the Micro3200 and 3280 systems (processing, memory, and I/O) are modular and therfore are expandable. If additional processing power is needed, you can easily add more processors (in multiprocessing systems) simply by plugging them in. To increase system memory in terms of both throughput and capacity, simply install additional memory modules. Similarly, you can expand the I/O system not only to accommodate more devices, but achieve a higher I/O bandwidth. None of these expansions necessitate changes to system software.

RTnet is a trademark of Concurrent Computer Corporation
ETHERNET is a registered trademark of Xerox Corporation
IBM is a registered trademark of International Business Machines Corporation

Software Compatibility

Those who already have a significant software investment in previous Series 3200 processors, can easily migrate their applications to Micro3200 and 3280 processors. All standard software packages such as run time libraries are supported on 3280 and Micro3200 machines. Additionally, 3280 processors support the microcoded FORTRAN enhancement package (FEP) and the hardware equivalent transcendentals. Any code changes that you want to perform to further optimize your current application is at your convenience.

Availability

The availability of a system is a measure of the time that a computer is at the disposal of its intended users. A system that accumulates little system downtime, offers high availability.

Certain failures, though undesirable, are tolerable to some extent. Naturally, the level of tolerance depends on the application. Some malfunctions, when they occur, cause degraded performance, but do not necessarily result in system downtime. System bus parity errors comprise one such example. If these errors occur, the recipient module signals bad status back to the initiating device. In response, the initiating device retrys the operation.

Even if component malfunction causes an auxiliary processor to fail, the operating system can maintain system service by redistributing the task load to the remaining processors. This resilience is one of the unique benefits of multiprocessing.

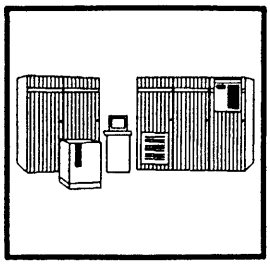
To a degree, memory failures are tolerated as well. If any portion of main memory is defective, the CPU marks the region off and restricts access to read–only.

Fault tolerance is also a factor in the delivery of power for our multiprocessing systems. These systems come equipped with modular power supplies that channel power to dedicated processors. Because power can be isolated, this form of power delivery (known as dedicated power) has distinct service advantages in addition to the fault tolerant aspects.

Data Protection

One of the more important considerations is the protection of data during a system malfunction. All 3280 and Micro3200 systems are standardly equipped with a back–up battery (OEM versions, battery optional). If AC line service fails, the system's power supply detects the falling line voltage and activates the battery back–up circuitry. Although normal system service is suspended during power outages, data held in main memory remains valid. Upon AC line recovery, the power supply immediately restores normal system power and recharges the back–up battery.

Data losses can also be induced by mechanical failure, such as a disk head crash. Although performing periodic back–ups may lessen the consequences of disk failures, some applications consider any data loss to be intolerable. Using the mirror–disk facility, non–volatile data is copied onto a back–up disk. In the event of a disk failure, operations can still proceed using the back–up disk.



2

System Software

In this chapter

We discuss the software offerings for the 3280 and Micro3200 Systems.

Concurrent provides programmers and application designers with a diversity of software products through which they can realize their goals. The following sections highlight these products and their capabilities.

Topics include:

- a discussion of the OS/32 operating system,
- descriptions of the available languages,
- task scheduling models for real–time applications, and
- an overview of Reliance PLUS software.

Using OS/32

OS/32, our proprietary operating system, provides field proven software support for both uniprocessor and multiprocessor implementations. Specifically designed to handle rapid task–to–task communication and provide immediate interrupt responsiveness, OS/32 offers the ideal solution for real–time computing and on–line transaction processing.

OS/32 acts as a central coordinator for all system activities such as error handling, task managing etc. This coordination can be achieved not only through the system console, but through user–written application tasks as well. Similar to many operating systems, OS/32 does provide default operation; however, that is where the similarity ends. Through user–written application programs, programmers can override the defaults and establish an operating environment that is more responsive to the dynamics of a specific application.

OS/32 is a versatile operating system that:

- supports multi–user environments (through the multi–terminal monitor (MTM)),
- supports database management and office automation (through Reliance PLUS and Reliance™ Office),
- supports uni and multiprocessor configurations,
- provides for task–handled traps,
- facilitates intertask communication,
- offers a number of utilities not only for fundamental operations but for system performance analysis and tuning (through System Performance Monitor), and
- designed to meet C–2 level security requirements established by the U.S. Department of Defense.

In the following sections, we take a look at a few of the facilities provided by OS/32. In addition we examine several ways in which to implement tasks scheduling algorithms for data acquisition and control applications.

Reliance is a trademark of Concurrent Computer Corporation

Program Development

The MTM operates under OS/32 and provides a batch processing, time–sharing environment ideally suited for program development. MTM allows up to 64 interactive users to simultaneously write (code), compile, and execute unrelated tasks. With EDIT/32, our line editor or MEDIT, OS/32's full screen editor, programmers have two powerful resources for performing code modifications.

To accelerate the development process, OS/32 offers the new language environment. The new language environment, which is available to FORTRAN VII, CAL, COBOL, Pascal PLUS, and RPG users, performs a number of functions that would otherwise be performed by the programmer, such as setting the applicable file extensions and appropriate column tabs for coding. In short, the new language environment eliminates some of the more tedious aspects of program development.

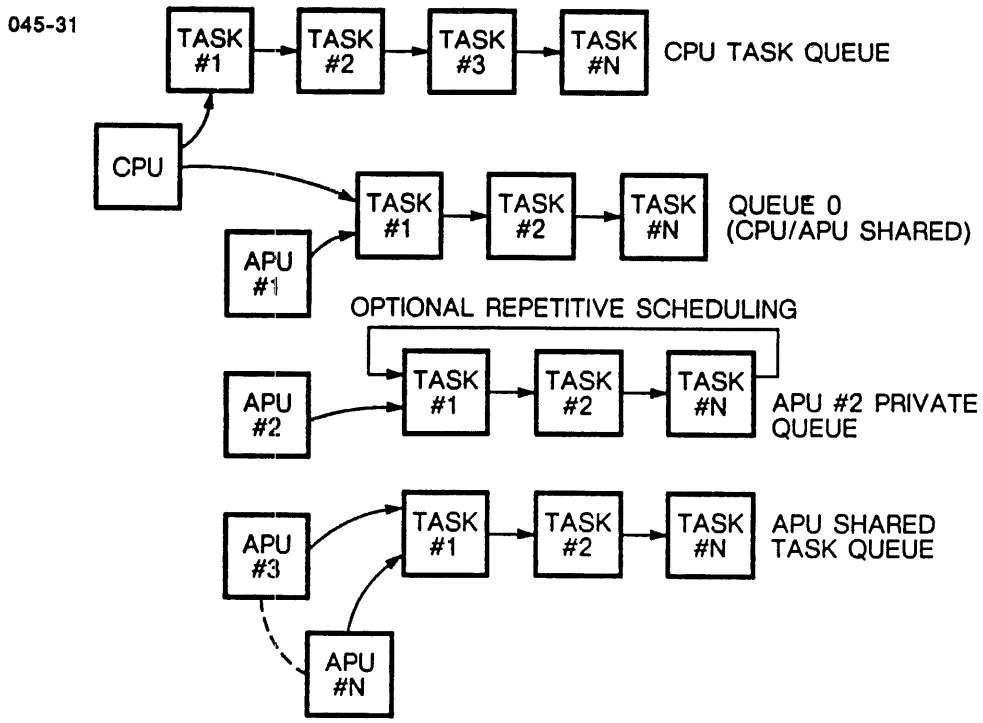
Another tool for developers is the command substitution system (CSS). The CSS is a command processor that allows you to simplify repetitive system procedures such as compiling and linking. By entering the desired command sequences and saving them in a file, you can automate any process that you would otherwise have to perform manually through successive command entries. To make a CSS process more universal, users can pass parameters to CSS variables.

Multiprocessor systems provide the optimum environment for program development. Since the auxiliary processors execute the job requests, the CPU can devote its resources to the I/O intensive program coding and editing phases. The result is a responsive system both in terms of program execution and user terminal activity.

Task Scheduling

Programmers using OS/32 can coordinate the execution of their jobs through a number of task dispatch queues. Four kinds of queues exist:

- CPU task dispatch queue
- CPU/APU shared task dispatch queue
- Private APU dispatch queue
- APU shared task queue



The CPU's task queue controls the execution of its assigned tasks on either a priority basis or an enforced priority basis. The APU's shared and private queues, dispatch tasks on a priority, enforced priority, or on a first–in/first–out (FIFO) basis. The assumed priority is either a default value, the value assigned by the console operator, or the value assigned by the application.

The priority scheduler dispatches the highest priority task for execution and allows it to run to completion. It then dispatches the task with the next highest priority. Enforced priority differs in that the currently executing task does not necessarily run to completion. Instead, if a task of higher priority is placed on the queue, the scheduler preempts task execution and dispatches the high-priority task. Upon completion the preempted task resumes execution. The FIFO scheduler dispatches tasks purely on a first come, first serve basis. There are 253 available task priorities, ensuring that the critical tasks execute before the non–critical tasks.

The dispatching mechanism differs between the CPU and APU queues. On the CPU queue, the operating system organizes the queue and releases the highest priority task to the CPU. On the APU queue, the operating system only forms the specified queue (FIFO, priority, or enforced priority), but does not actually dispatch the tasks for execution. Instead, the APU, using its microcoded scheduler, automatically accesses the front of its assigned queue and begins execution. Because the operating system does not intervene in the dispatch of APU tasks, system overhead is reduced substantially. In addition, you can maintain a cyclical queue whereby jobs reschedule themselves after execution, without incurring any operating system overhead. Further details on cyclical queues are provided in the section entitled *Multitask Model.*

Programming Languages

Programmers using 3280 and Micro3200 systems have an extensive array of programming languages from which to choose. Available languages consist of:

- FORTRAN VII
- C³Ada™
- C
- Pascal
- Assembler
- COBOL
- RPG II
- BASIC
- CORAL 66

Because previous Series 3200 computers also support these languages, programmers can easily migrate their previous applications to current machines.

One of the most powerful features that OS/32 provides is the ability for programmers to control system resources through their application programs. This capability can be provided by the OS/32 System Support Run–Time Library (RTL), discussed in the following section.

C³Ada is a trademark of Concurrent Computer Corporation

System Support Run–Time Libraries (RTLs)

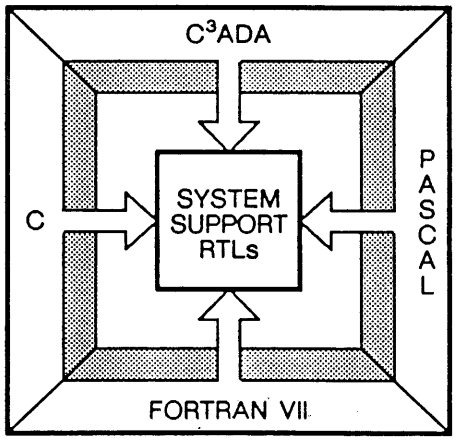
Using simple call statements, your application can:

- conduct system I/O, including error handling,
- control the execution of itself and other tasks,
- generate and handle task traps,
- manage processing resources within a multiprocessing environment, and
- perform analog/digital conversions.

Generally, these capabilities are similar to those of supervisor calls (SVCs) which are in the domain of assembly language programs.

System Support RTL routines can be called from any FORTRAN, C^{3}Ada, C, or Pascal application.

1045-32



FORTRAN VII

FORTRAN programmers have three FORTRAN compilers from which to choose:

- a standard development compiler (F7D),
- a globally optimizing compiler (F7O), and
- a universally optimizing compiler (F7Z).

With a compile speed of over 3000 lines per minute, the standard F7D compiler allows programmers to quickly evaluate the success of their programs.

In time–critical applications (flight simulators, nuclear power monitors, etc.), the requirements of the application can impose tremendous constraints on task execution. For these applications, lower task execution speeds result in degraded performance of the general application: flight simulators with slower frame rates and nuclear powerplant monitors that sample fewer test points. Using optimizing compilers, FORTRAN programmers can streamline their applications to produce highly efficient source code and minimize code expansion after compilation.

The F7O globally optimizing compiler performs many optimizations such as:

- rearranging mathematical expressions to produce more efficient machine code
- removing invariant operations from DO loops and placing them before the loop (thereby reducing the total number of operations)
- eliminating variables by substituting locally assigned values.

Several of the optimizations are switchable, thereby allowing programmers to disable individual optimizations.

The F7Z universally optimizing compiler goes one step further. At the programmers request, F7Z crosses program unit boundaries and optimizes the subprogram code called by the main program. This capability extends optimized code to even highly structured applications.

Interfacing FORTRAN VII and Assembly Language

FORTRAN programmers who prefer to code some sequences in Assembly language can do so using the FORTRAN/Assembly interface. Using Assembly language calling sequences or FORTRAN compiler directives, programmers can either call Assembly language subroutines from their FORTRAN program or they can place Assembly code directly in their FORTRAN mainline.

The ability to call Assembly language subroutines allows FORTRAN programmers to use existing Assembly language libraries rather than recode the same functions in FORTRAN.

C³Ada

Concurrent offers a fully validated Ada language, the C³Ada Language System.

C³Ada is fully integrated with our OS/32 operating system, providing programmers with access to the powerful real–time System Support RTL functions, such as:

- event handling,
- device I/O,
- task control and intertask communication,
- memory management, and
- multiprocessor control.

Although C³Ada, with the 3280 and Micro3200 systems, is aptly suited for large–scale real–time applications, most such applications are presently written in FORTRAN. To protect your investments in field–proven FORTRAN libraries, C³Ada incorporates the pragma INTERFACE, which allows you to call your FORTRAN and Assembly language subroutines without complex calling sequences. Not only does this feature facilitate the use of existing libraries, but as those libraries are converted to Ada, programmers can easily adapt their Ada source code by deleting the INTERFACE statement.

To aid in the program development cycle, the C³Ada Language System offers a comprehensive suite of Ada Programming Support Environment (APSE) tools. These tools allow you to:

- perform a preliminary syntax check on your source code prior to compilation,
- tailor the appearance of your source program listing, and
- locate non–initialized local scalors.

To debug your application or evaluate its run–time performance, the C³Ada language system offers a symbolic run–time monitor (SRTM) and a symbolic debugger. The SRTM works with your **running** application without affecting its execution. Using the SRTM, you can change the values of program variables on the fly, thus allowing you to observe and evaluate run–time program behavior.

You can operate the SRTM either in an interactive scroll mode or in a page mode. The scroll mode allows you to enter commands one at a time, interactively. In the page mode, the SRTM displays desired variable names with their current values in real–time. Up to 64 pages may be defined, permitting you to monitor a large number program variables.

If run–time operation is not possible, you can debug your application using the symbolic debugger. With the symbolic debugger you can set program breakpoints, tracepoints, and variable values. You can also single–step through your code, line by line to closely monitor program activity and locate bugs. Also, because Ada permits overloading, the symbolic debugger allows you to pinpoint or locate specific variables.

The SRTM and the symbolic debugger share a common Ada–like command language. Using this command language, you can develop your own procedures.

COBOL Development Environment (CoDE)

CoDE offers a comprehensive development environment for implementing COBOL applications. CoDE is offered as a package and contains a compiler, symbolic debugger, forms generator, conversion utilities, and various tools.

The compiler is tuned for Series 3200 architecture, and as a result, takes advantage of its native functions. Consequently, highly efficient object code is produced providing for rapid task execution.

The symbolic debugger allows you to trace program execution by displaying source code statements on the screen as they execute. Under user control, programs execute continuously, one statement at a time, or at a variable rate.

To speed the development process, the forms generator provides a powerful tool for users who want to create screen forms under MTM. Using the keyboard, users "paint" the desired application form on their terminals. Then the software generates the necessary COBOL source code to handle the screen. For transaction processing applications, programmers can use Reliance PLUS software to perform the same functions.

Compatible with many other COBOL dialects (e.g., RM/COBOL, Concurrent's COBOL, IS COBOL and IBM COBOL), CoDE eases migration for existing applications. Additionally, CoDE works with Reliance PLUS, our transaction processing and relational database management software. Using CoDE and Reliance PLUS, you can design and implement efficient, high–performance on–line transaction processing applications.

Other Software Languages

In addition to FORTRAN and C^{3}Ada, Concurrent provides programmers with a wide selection of programming languages. Our C compiler is a full implementation of the industry–standard C language. Furthermore, C along with Pascal have built–in interfaces to the System Support RTLs, providing programmers with a powerful system control resource. For the laboratory and the office, we also offer RPG II, BASIC, and CORAL 66.

Scheduling Tasks For Real–Time Applications

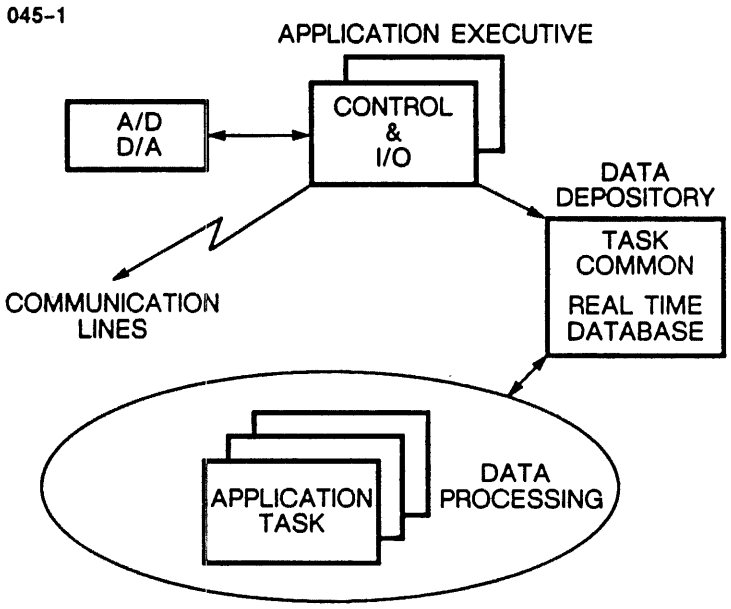
Implementing an efficient task scheduling scheme is one of the more fundamental aspects of designing real–time applications.

The following discussions focus on two models for scheduling tasks on our multiprocessor systems. Although there are other ways in which to schedule tasks, the intent here is to demonstrate some of the powerful resources inherent in our software and hardware.

Multitask Model

The first model, the multitask model, lends itself to applications with low to moderate frame rates (1 second to 1ms) and where there is a small number of scheduled tasks. As the name implies, the multitask model schedules the execution of several tasks to perform some overall function. Additionally, the multitask model is particularly well suited for migrating an application from a uniprocessor system to a multiprocessing environment. Normally, data acquisition and control applications developed for a uniprocessor employ large numbers of asynchronous tasks that timeshare the processor. Transition to a multiprocessor allows the inherent parallelism of most applications to be fully realized.

Three entities comprise this model; an application executive, a data depository, and a data processing element.

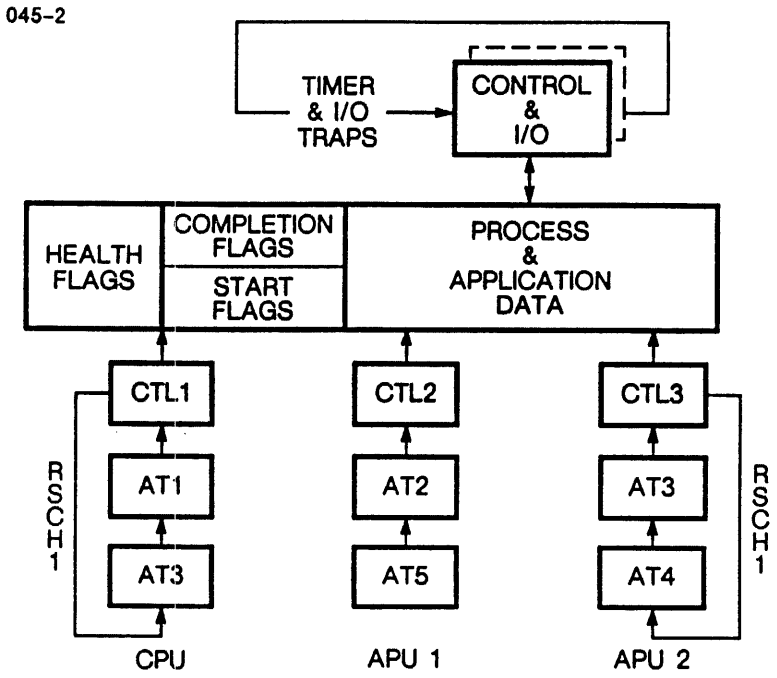


The application executive serves as a task administrator controlling program execution and performing I/O processing when required by the application. Assigned to the CPU, it consists of a single or a set of event–driven, high priority task(s). Using synchronized timer traps, provided by the operating system and clock hardware, the executive establishes and maintains the system heartbeat. When computational processing is required, the executive informs the appropriate set of tasks.

The application executive interfaces the external world (A/D hardware etc.) using a proceed I/O call mechanism. When an I/O interrupt occurs, the request is logged by the operating system and the system returns to the executive for continued I/O processing. Upon completion of the request, the executive experiences an asynchronous trap into a specific service routine.

The data depository, consisting of application variables and operational flags, resides in a system wide task common area. Task common is an area in main memory that is shared by all tasks and accessed in accordance with read and write parameters established at link time. Data passing between the executive and data processing is temporarily stored in buffers. Input data required for computational processing is stored in input buffers and is used by application programs when needed. When processing is complete, the programs store their processed data and outgoing information in output buffers for subsequent transfer to external devices by the application executive. The task common area also provides a communication medium between the executive and the computational tasks. Specifically, it supports the start, completion, and optional health flags that allow the executive to determine the operational state of the system.

Data processing is accomplished by several application tasks that execute on the CPU and available APUs. Each task uses a start flag to initiate execution. The tasks are distributed among the available processors and are scheduled on FIFO queues according to application dependencies. Tasks on different queues may run asynchronously and concurrently while tasks on the same queue follow a set order.



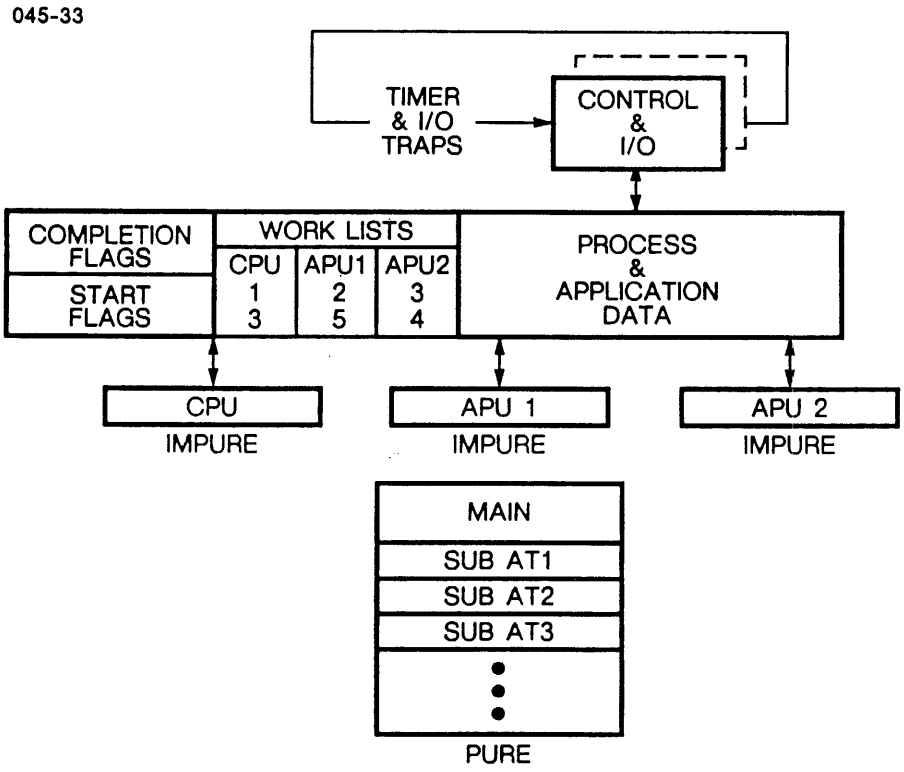
At the top of each queue is a simple control task that loops until the beginning of the next frame. When the frame begins, the control routine executes a re-schedule (RSCH) instruction which moves the task to the end of the queue. Each task in turn, examines its start flag in task common to determine if it is scheduled to run. If the flag is set, the task starts to execute. Upon completion, the task places itself at the end of the queue by performing a RSCH instruction. After all the tasks in the queue have examined their start flags, the control task becomes current again and loops until the next frame begins.

Consolidation Model

The consolidation model provides an optimized approach for implementing a data acquisition and control application on a tightly–coupled multiprocessor system.

Like the multitask model, the consolidation model is similarly partitioned into three main elements. An application executive controls the system and performs the I/O functions as needed. A data depository, implemented in a system–wide task common memory segment, provides the data and communication medium for the executive and other application routines. Finally, data processing performs the computational exercises required by the application.

The main difference between this and the multitask model is the number and structure of data processing routines and the scheduling mechanism that controls them. The data processing element is a consolidation of processing routines grouped into a single task. Each routine performs a function corresponding to a task in the multitask model. The routines execute asynchronously on the various processors.



In this model, task scheduling is realized as subroutine branching. This is quicker than the task scheduling approach because context switching is avoided.

All processors can execute this single task by the pure code facility of OS/32. Each processor contains an impure section in memory for task context and privileges, but shares a single memory segment that contains the executable code and application work lists. By scheduling the sequence of subroutine execution, work lists control the application functions. Each processor examines its work list to determine the subroutine sequence it must execute. Modifying the list contents provides a means for adjusting the sequence. This mechanism also simplifies on–line error recovery. Instead of remapping application tasks to a working processor, the failed processor's work is distributed among the other lists.

By reducing the overhead required by the application, this model permits a substantially increased system heartbeat. This allows the application to use more sophisticated equipment and control more data points.

Business Applications For OS/32

Reliance PLUS software focuses on the commercial user who requires high–performance transaction processing, database management, and database query capabilities. The efficiency of Reliance PLUS combined with the rapid system hardware ensure a responsive transaction processing environment.

Reliance PLUS comprises a transaction processing monitor and a database manager called the database management system (DMS/32). Both software components are resident (fixed in memory) to enhance response. The monitor services the user community and handles the real–time aspects of the environment. DMS/32 provides for multi–keyed data retrieval and update for multiple user files. It also manages real–time concurrent data access while maintaining database integrity through automatic on–line recovery features, record and file locking, and facilities for rapid database restoration in the event of a system failure.

Aside from supporting the highly interactive environment, DMS/32 also provides for background transactions. In addition, off–line batch processing, which can be useful for performing sizable noninteractive tasks such as large database updates, are similarly supported.

Database Integrity

Two types of general malfunctions can jeopardize the integrity of a database: incomplete transactions and database media failure. Reliance PLUS software ensures database integrity through its automatic rollback and rollforward mechanisms. If the system fails before a transaction completes its file updates, Reliance PLUS initiates a rollback and restores the database to the same state as it was prior to the failed transaction.

The roll–forward capability protects your data resource in the event of a database media failure. If this occurs, Reliance PLUS can rebuild your database by updating the database security copy with the log file. The security copy is simply an older version of the database. The log file records all transactions that occurred since the last update to the security copy.

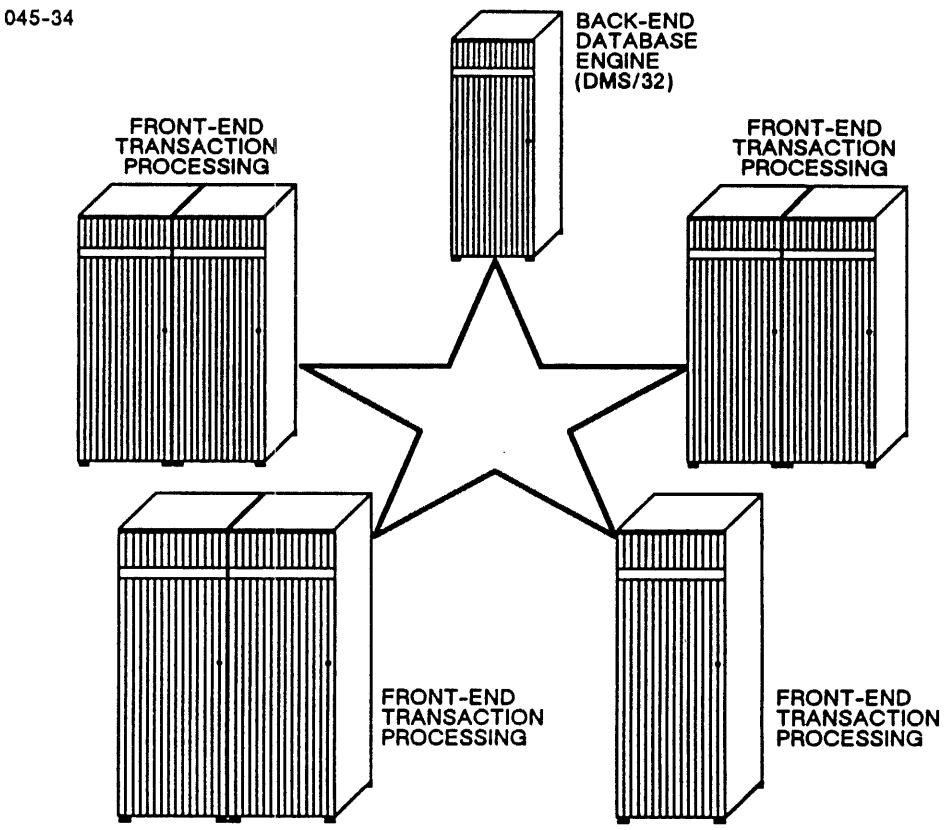
Securing Your Database

Reliance PLUS secures your database and application in general through an integral software component called the distributed security system (DSS). Using DSS, you can selectively control access to your application's screen forms, files, and data views from within the transaction processing environment and from external tasks running under OS/32 and MTM. DSS also guards against unauthorized access from remote systems.

Multiprocessing and Reliance PLUS

For time–critical, high–volume transaction processing applications, we offer Reliance STAR™, a multiprocessor version of Reliance PLUS. Reliance STAR provides a unique solution for those who want to expand the capacity of their present Reliance PLUS application without code modifications. Furthermore, because Reliance STAR is modular, upgrades are built upon existing hardware, thereby preserving initial investments.

Reliance STAR is a trademark of Concurrent Computer Corporation



Reliance STAR networks use a single database engine called the DMS/32. For the systems discussed in this manual, DMS/32 serves up to five front–end processors each of which executes the application programs and maintains the application's user interface. A fully configured Reliance STAR system can support up to 1000 active user terminals and printers distributed over five systems.

Reliance Office

Reliance Office automates the essential aspects of the office environment. By offering word and data processing, through LEX™, electronic mail, and agenda facilities, through NEM/32, Reliance Office provides you and your staff with an integrated office environment.

LEX is a menu–oriented, interactive full screen editor. It possesses all the features necessary for document preparation. With a what–you–see–is–what–you–get display, users can easily create the documents they need using single–key function select editing features.

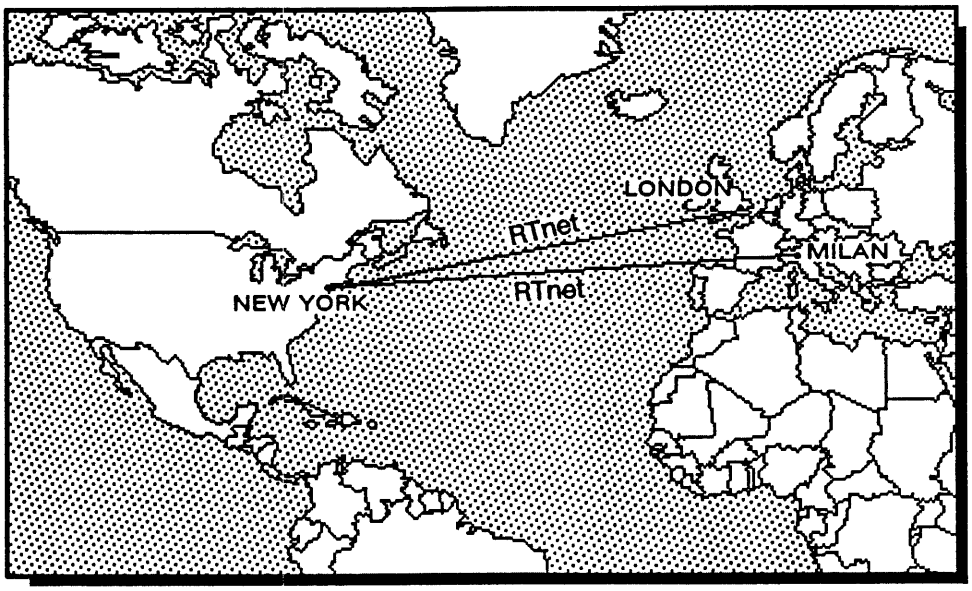
Network electronic mail (NEM/32) provides a convenient communication medium for Reliance PLUS users. Using NEM/32 you can:

- send mail or memos electronically to any member of the general user community. Furthermore, NEM/32 can inform you of when your messages are received.
- send messages in advance at predetermined times and dates.
- send the same message repeatedly at specified times.
- receive NEM/32 mail and delete, save or print any message.
- use the Diary/Calendar, a feature of NEM/32 that allows you to coordinate your business and personal activities. If you want, you can grant access rights to another colleague.

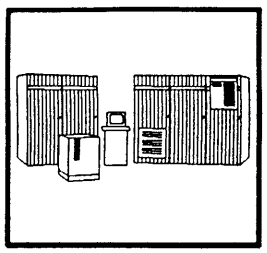
Users of NEM/32 require no special training. All operations are menu–driven, and use prompts to assist with proper entries.

LEX is a trademark of ACE Microsystems Limited

1045-35



The benefits of NEM/32 are further realized when remote Reliance PLUS systems are interconnected via RTnet, Concurrent's network system. With RTnet, NEM/32 permits even geographically dispersed user communities to maintain close contact.



3

System Organization

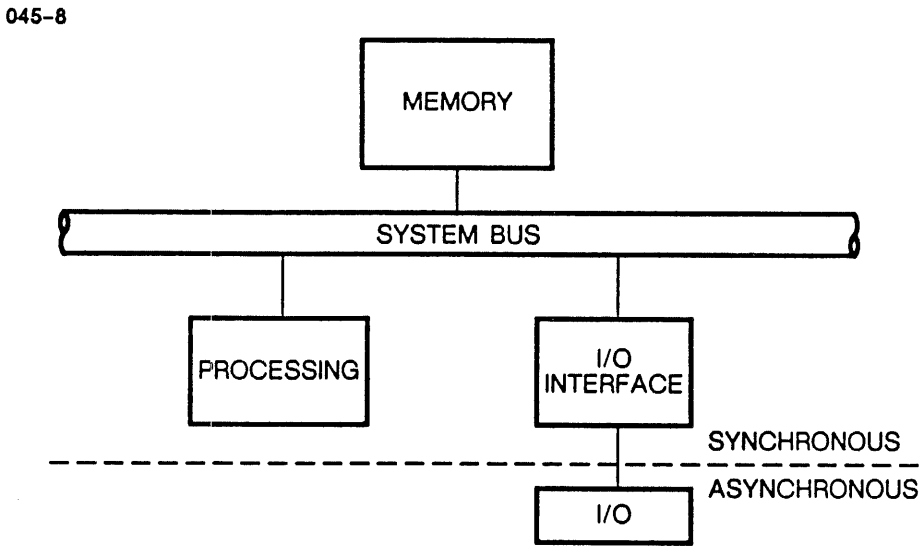
In this chapter

We examine some of the hardware that comprises the 3280 and Micro3200 systems.

Topics include:
- descriptions of the:
  processing system,
  system buses,
  memory system,
  I/O system, and
  Control/Diagnostic system.
- a brief account of distributed modular architectures.

Hardware Overview

Each system distinguishes itself in terms of performance, physical size, and configuration flexibility; however, there are similarities between the systems that establish them as a family of processors. Perhaps the single most significant commonality between the systems from a hardware standpoint, is their synchronous bus structure. Each system organizes its primary functional properties (processing, memory, and I/O) around a synchronous system bus.



The remainder of this chapter discusses each of the primary functional elements that comprise the 3280 and Micro3200 processing systems.

Processing

Depending on the system (uniprocessor or multiprocessor), two kinds of processors can share processing responsibilities: a central processing unit (CPU) and an auxiliary processing unit (APU). The CPU is a fully functional processor, capable of singly performing all the processing functions of a computer system. Alone it executes the operating system, services the I/O subsystem, and responds to synchronous and asynchronous events within the application environment. In uniprocessor systems, the only processor present is the CPU. In a multiprocessing system, the CPU serves as the system coordinator either by dispatching tasks to the auxiliary processors or by performing their I/O requests.

An APU, on the other hand, is optimized primarily for computational processing. With no I/O responsibilities, the APU concentrates its resources on algorithmic processing. In addition, its microcoded scheduler diminishes CPU interaction and facilitates fast context switching.

A CPU and its subordinate APUs differ only in the microcode present in each processor's control store. Because their hardware is identical, both are described collectively in the following paragraphs.

Processor Hardware

CPUs and APUs each are available in three processor models:

- a MicroThree rated at 3.7 MIPS*,
- a MicroFive rated at 6.0 MIPS*, and
- a 3280 rated at 6.4 MIPS*.

To a large extent, it is the processor models that define the individual systems.

* Benchmark result of single–precision Whetstone

The 3280 processors consist of four processor boards and are used in the 3280MPS and 3280E MPS machines. MicroFive and MicroThree processors, collectively known as the Micro3200 processors, use only a single PC board and are used by the Micro3200 machines. Although the four board processor offers slightly greater performance over its one–board counterpart, both have similar features such as:

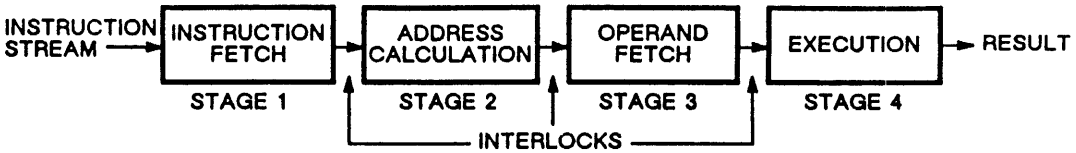
- pipelined architectures,
- data and instruction caches,
- instruction prefetch circuitry,
- parallel multipliers,
- on–board floating point execution units,
- numerous registers,
- prioritized interrupt structure

Pipelining Instructions

The 3280 and Micro3200 processors pipeline their instruction streams into four stages of general execution:

- instruction fetch,
- address calculation,
- operand fetch, and
- execution.

045-9



Although nonpipelined processors devote all their resources to the execution of a single instruction, pipelined processors target only those resources required for the given stage. This allows the processors to overlap the execution of four consecutive instructions and boost processor throughput.

The stages operate simultaneously on different instructions. For example, while the processor fetches an operand for one instruction, it computes an operand address for the next instruction. Under some conditions, like a cache miss, more than one cycle may be needed to complete a given stage. To prevent subsequent instructions from advancing prematurely to the next stage, interlock circuitry is used.

The benefits of pipelining depend on the type of instructions most frequently encountered in the program and also by the sequence of instructions. For example, the simplest instructions using register–to–register (RR) formats require little pipelining; hence, the properties of pipelines are not fully exploited. Other instructions, which reference memory addresses (register and index storage (RX) formats), tap the pipeline resources and execute much faster than they would on nonpipelined processors.

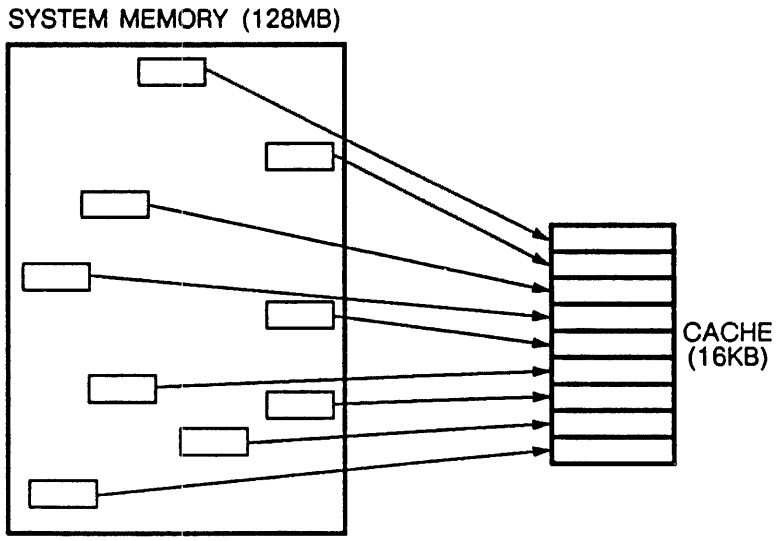
Cache

The benefits of caching stem from a program's tendency to reuse the same memory location or use memory locations close to those currently being used. This characteristic is known as time and space locality.

All of our processors, CPUs and APUs, use caches to enhance system performance. Cache is an on–board local memory that, when valid, is essentially a carbon copy of main memory. Its proximity to the processor coupled with the fact that processors have immediate cache access, results in data and instruction fetches at processor clock speed.

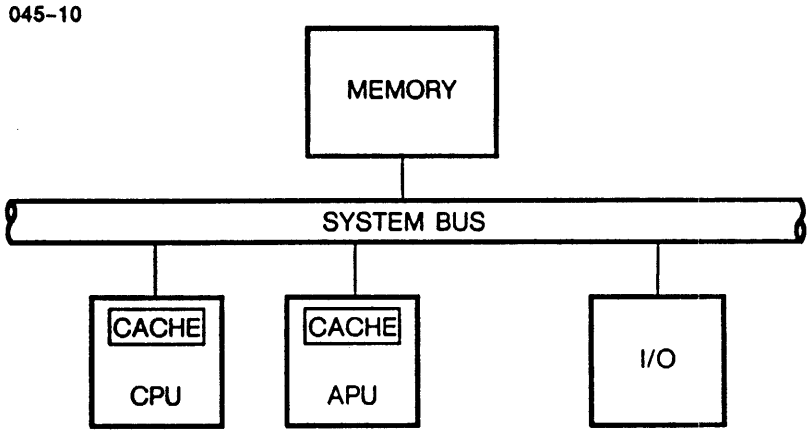
While cache is a redundancy of some portion of main memory, that portion need not be composed of contiguous memory locations. In fact, cache data can be distributed over many memory locations.

045–11



Our processor caches operate on a quadword basis. When a processor reads from a memory location, it loads an entire quadword of data in its cache. Regardless of the amount of data needed, the processor always requests a quadword. This tactic not only exacts high bus efficiency, but because subsequent memory locations are likely to be used (locality), the processor attains higher cache hit ratios.

On subsequent reads to the same location or locations within the quadword block, a cache hit occurs, and the cache rather than main memory responds to the request. Cache not only facilitates processor fetches, but because the system's memory bus is less burdened, bus availability increases for other system elements such as mass storage devices or other processors.



Caches have write–through organization, meaning that if a processor writes to a cached memory location, it updates both memory and its cache. Again, because programs have a tendency to access the same locations (i.e., variables etc), this technique produces higher hit ratios.

Micro3200 processors use a direct–mapped instruction cache and a 2–way set associative data cache. 3280 processors deploy 2–way set–associative caching for both the instruction and data caches. A 2–way set associative cache actually consists of two caches, one being redundant of the other. Due to the redundancy, rather than overwrite the old, but valid data, the processor stores the new data in the other cache. Through the use of set associative caches, processors attain yet higher hit ratios.

Instruction and Data Cache

Micro3200 and 3280 processors implement their caching functions using two specialized caches: an instruction cache, for holding program instructions and a data cache, for storing the program's data. Because these caches are independent of each other, processors can fetch instructions and operands simultaneously. This design eliminates cache contention and the delays of the processor's interlock circuitry.

Instruction Prefetch

Processor's usually fetch their instruction streams from sequential memory locations. The instruction cache exploits this pattern by using look–ahead or prefetch logic. Prefetch logic anticipates the fetching of instructions by retrieving them from memory and caching them before they are actually needed by the processor. In this manner, processors can fetch instructions almost entirely from cache and avoid the delays involved with memory fetches.

Parallel Multiplier

Processors execute multiply instructions using parallel multipliers. By using integrated multiply hardware, significant performance gains are realized over the algorithmic alternatives.

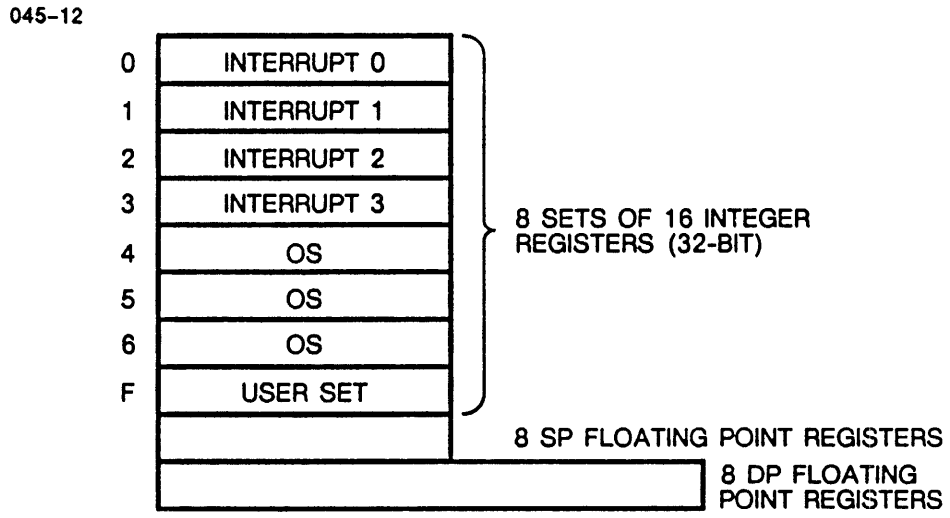
The multipliers operate on both single and double precision operands. Furthermore, they multiply both integer and floating point numbers.

Register Allocation

The number of registers to which programs have access, can substantially impact program execution. In general, a sizable register allocation results in fewer memory fetches and increased throughput.

3280 and Micro3200 processors have a rich supply of user registers. User tasks have access to:

- 16 single–precision integer registers (32–bit)
- 8 single–precision floating point registers (32–bit)
- 8 double–precision floating point registers (64 bit)



Our processors deploy seven additional register sets with each set comprising 16 integer registers. Sets four through six facilitate internal OS usage and sets zero through three service each of the four I/O interrupt levels. By assigning discrete register sets to individual functions, the delays associated with saving and restoring registers (context switching) are avoided.

Control Store

The processors' control store retains the microprogram used to emulate user–level instructions. 3280 processors offer the added benefit of providing 8k locations of user accessible control store. Using assembly language skills, programmers can custom tailor their more critical applications by writing proprietary microcode.

Real–Time Clock

All processors come equipped with a real–time clock (RTC). The clock is hardware driven from the system oscillator and has an accuracy of 0.001%.

To system software, the RTC appears as a 32–bit register, incrementing each microsecond. The RTC provides a time tag for those programmers who want to identify the sequence of events within the system, such as, timelogs and timed program sequences. It can also be used for decision making in time–critical framing applications.

Instruction Set

3280 and Micro3200 processors execute a set of instructions that optimize both uniprocessor and multiprocessor performance in a number of application arenas. For scientific applications, all transcendental operations such as sine, cosine, and natural logarithm exist as primitive instructions for both single– and double–precision operands. Because these functions are implemented in microcode, programs avoid the delays of branching to user–level run–time library algorithms.

Commercial applications, which often operate on large data strings, have access to a number of decimal and alphanumeric string instructions. These instructions allow you to move, compare, initialize, and change strings of data.

For all application arenas, usual branch instructions allow processors to branch without waiting for condition code results. Although it ultimately tests the condition code for verification, usual branches result in a quick, 200ns branch.

User programs containing loops that perform many iterations benefit the most from usual branch instructions. For example, program loops like

```
DO 100 IREC = 1,500
```

in FORTRAN execute quickly because the processor's initial assumption is correct 500 out of 501 times.

To facilitate system handling and improve operating efficiency, the instruction set includes operations that simplify, among other things, task queuing, processor synchronization, and task dispatching.

Floating Point

Floating point processing units, frequently optional equipment, come standard with all 3280 and Micro3200 processors. Floating point hardware is an integral processor function. This eliminates the handshaking protocols required by external floating point processors, and thus boosts floating point performance. Single precision floating point loads and adds take 100ns, the same time required for the corresponding integer operations. To attain maximum precision, floating point values are normalized before and after execution.

Other Processor Features

Several other processor features facilitate program execution speed and accuracy. For example, shift and rotate instructions no longer pose degraded processor performance. The Micro3200 processor employs a barrel shifter which shifts any number of bits in a single cycle. The 3280 processor uses a shift register, and shifts up to eight bits per cycle.

Another example is demonstrated by the way in which the processors execute string instructions; specifically those that move byte strings from one location to another. Rather than writing to sequential byte locations, the processors buffer the bytes up to a quadword and then write the quadword intact to the proper starting location. This tactic promotes system bus efficiency and increases bus availability for competing modules.

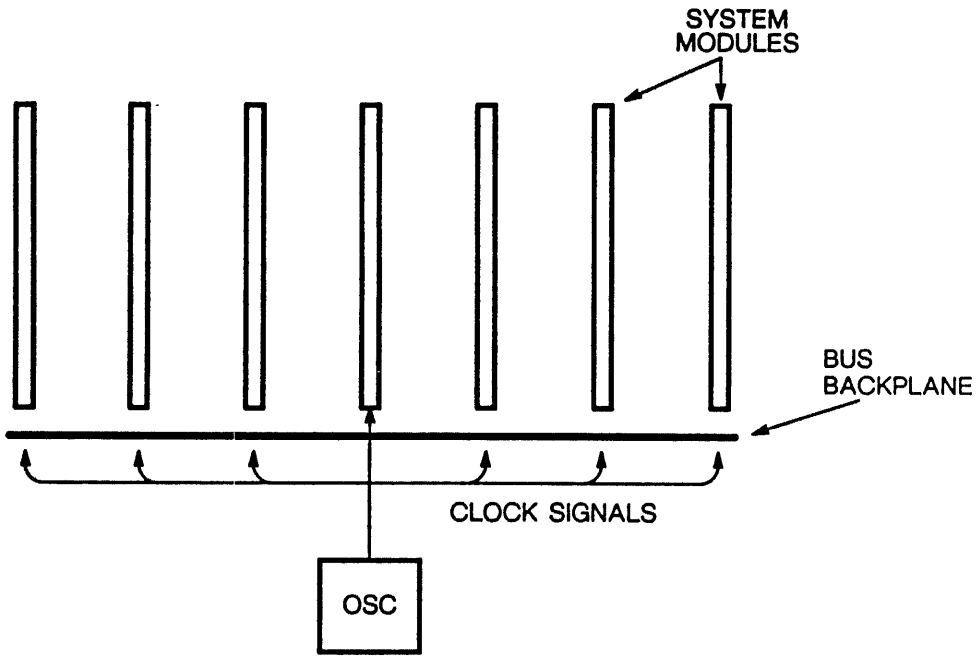
System Buses

Each of the 3280 and Micro3200 systems are organized around a central system bus. The system bus, which depending on the individual system is either an S–bus or an E–bus, carries all memory requests/data, I/O, and interprocessor messages.

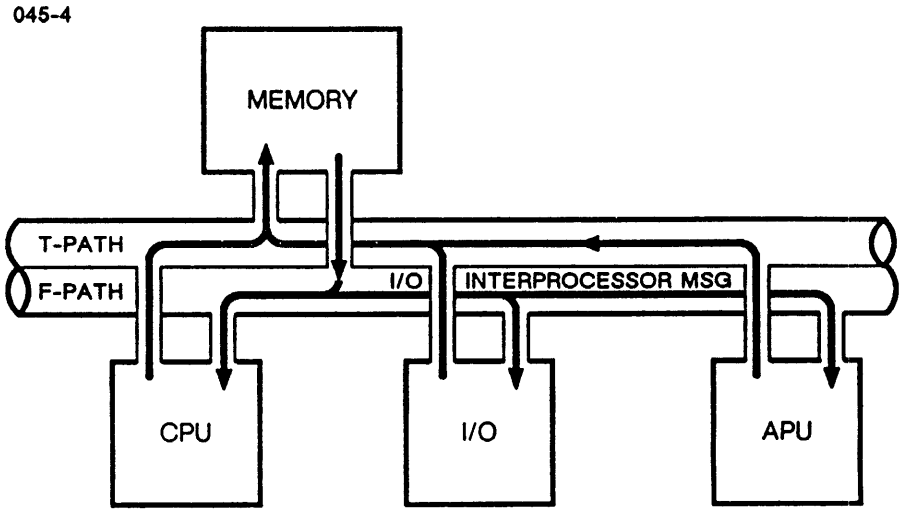
The S–bus is used in the Micro3200 systems and 3280 MPS while the E–bus is used in the 3280E MPS. Architecturally identical, the two buses differ only in clock speed. The S–bus clocks at 10MHz and yields a usable aggregate read/write bandwidth of 64MB/s. The E–bus is clocked at four times the S–bus rate and provides a 256MB/s bandwidth.

Both system buses are synchronous. They receive their clock signals from an oscillator and radially distribute timing signals to system modules such as processors and memory boards. By delivering clocks radially rather than serially, clock skew is kept to a minimum and system timing maintains precision.

045-3



Each system bus comprises two, independent 32–bit data paths: the T–path, for To memory and the F–path, for From memory. The T–path carries memory requests and write data from the initiating device, like a processor or DMA device, to main memory. Memory responding to these requests places the returned data on the F–path.



In addition to memory operations, the F–path also carries processor I/O commands, broadcasts, and interprocessor messages.

Both buses, the S–bus and the E–bus, have a positional priority scheme whereby the placement of a board within the chassis determines its ability to acquire the bus. By installing system modules in an appropriate arrangement, you can introduce biases for those elements that require more immediate bus access.

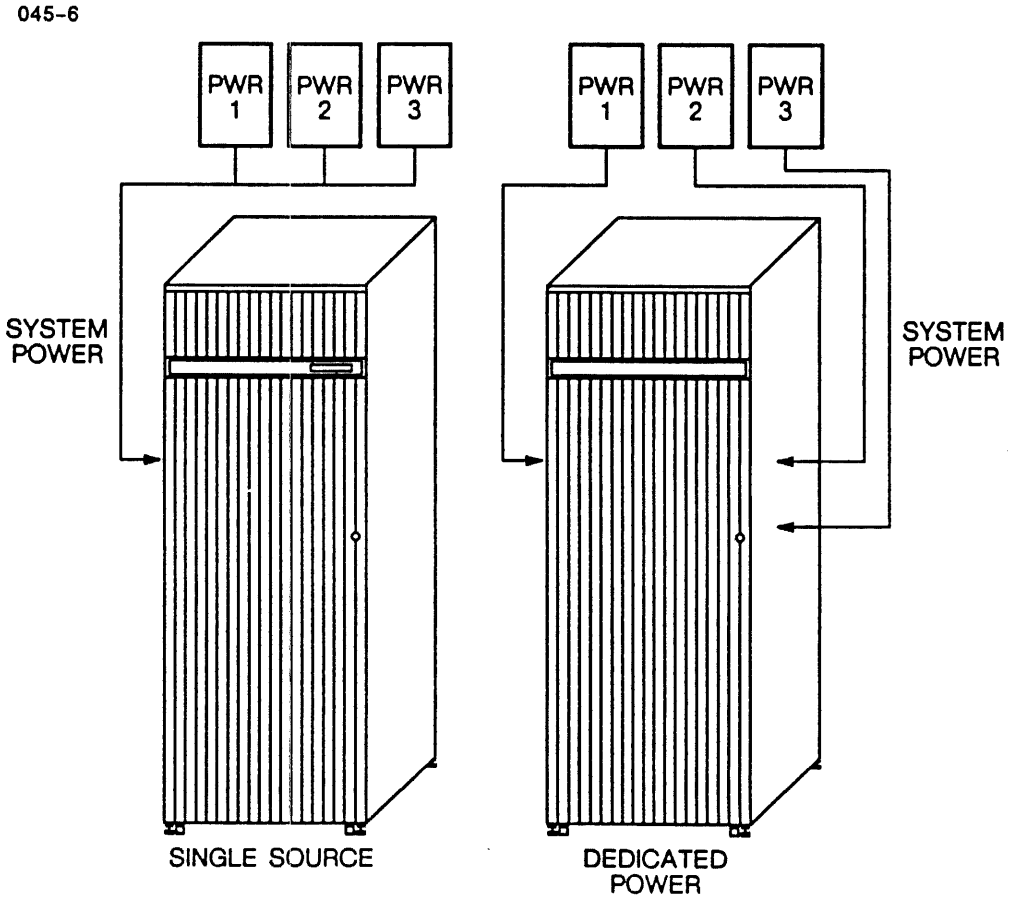
Power Subsystem

In keeping with the building block approach, our larger systems, from the Micro3200 Expanded System through the 3280E MPS, come equipped with modular power supplies. Their modular design permits you to install only the necessary amount of power needed by the system. When you need to expand your system, you can easily add more power modules to satisfy the new power requirements. While there are limits to the amount of power that can be delivered, all power supplies are designed to provide ample power for the maximum cabinet configuration.

The Micro3200 Compact System, unlike the other systems, has a single, non–modular power subsystem. This is because the differences in power consumption between maximum and minimum configurations are small and do not warrant a modular power subsystem.

Power Distribution and Fail Soft Considerations

The distribution of power within a computer system can substantially impact system availability and serviceability. Instead of providing a single source of system power, dedicated power schemes channel the outputs of isolated power modules to specific regions of system cabinetry.

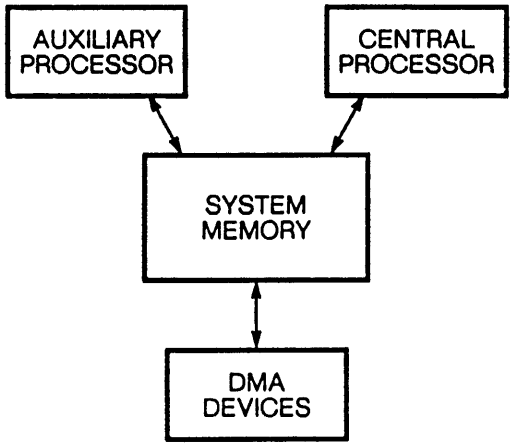


This lessens the potential for total system failures since the disruption of one power module affects only those components to which it supplies power. Further, dedicated power facilitates system maintenance because only a fraction of the system needs to be shut down.

System Memory

System memory is accessible to all memory requesting devices. These include all processors (CPU and APUs), plus direct–memory access (DMA) devices, such as disks and magnetic tape drives. By keeping memory uniform, all devices, under program control have immediate access to their neighbor's data.

045-7



Memory Modules

3280 and Micro3200 systems use either ECL–based memory modules, called EMMs, or composite memory modules, known as CMMs. Although the EMMs are faster than CMMs, both types of memory modules operate in the same manner and have similar attributes.

The memory modules incorporate the functions of a memory controller and a RAM array on to a single printed circuit board. This high level of integration simplifies installation, increases reliability, and contributes to the compactness of the total system.

The RAM array stores the actual memory data while the controller portion coordinates the memory requests and performs memory overhead functions such as, refresh, scrubbing, and error check and correction.

Memory modules queue write and read requests in an internal buffer. Although the request may not be immediately processed, the memory module acknowledges receipt of the request back to the initiating device. This type of deferred response operation increases T–path throughput by diminishing the number of repeated memory attempts.

To ensure data integrity, the memory module pipes all data entering and leaving the RAM array through its error check and correction (ECC) circuit. The ECC circuit detects and corrects all single–bit errors, detects all double–bit and some multiple–bit errors. When the ECC detects an error, it reports the occurance to an on–board error logger that is under CDS control. This error logger is periodically interrogated by system software to identify faulty memory components at the chip level. If a non–correctable error occurs, the memory module informs the requesting devices so that they do not use the corrupted data. In addition, the operating system marks off the corrupted memory region, taking it out of service.

Scrubbing, another memory function, adds another measure of reliability to memory data. Its function is to exercise all memory locations to prevent decay through dormancy. The memory scrubber continuously reads quadwords from sequential memory locations. As each quadword is read from the RAM array, it is piped through the ECC circuit like all other memory requests. If a single–bit error is detected, the error is corrected and the data is written back to memory. The scrubbing function tends to keep single–bit errors from developing into non–correctable multi–bit errors. Because the scrubber exercises memory at a rate much slower than usual system requests, it's operation is essentially transparent to system activity.

Depending on the system, memory modules are available in various sizes currently, 8MB, 16MB, and 32MB capacities.

Input/Output (I/O) System

System I/O is interrupt driven. Typically, some device interrupts the processor which services the pending request. I/O devices, such as terminals, disks, etc., can be linked to one of four priority interrupt levels.

The I/O subsystem is responsible for moving data between the computer system and the attached peripheral devices. Peripheral devices are divided into two main categories, low– and high–speed. Low–speed devices, such as printers and terminals, are controlled exclusively by the CPU. High speed devices, also known as direct memory access (DMA) devices, such as, magnetic tape drives and disks, receive direction from the CPU but transfer their data directly to memory without CPU intervention.

The I/O subsystem is implemented using two main buses, the multiplexor (MUX) bus and the DMA bus. The MUX bus with a maximum bandwidth of 800kB/s supports low speed, asynchronous data transfers and handshaking for high–speed devices. The DMA bus provides the high–speed artery for transferring large volumes of data between disk and tape devices and memory. Each DMA bus can support a sustained transfer rate of 10MB/s. Furthermore, the modularity of the I/O subsystem permits system designers to add more DMA buses to achieve a greater I/O bandwidth.

The I/O subsystem interfaces the 3280 and Micro3200 systems through the direct memory interface (DMI) module. This module generates the two primary buses, the MUX and DMA buses. By connecting the I/O system to the central paths rather than directly to the CPU, the console operator can perform relatively simple corrective measures in the event of a CPU failure.

I/O Configurations

The available offering of peripheral devices and their combinations are too extensive to discuss individually. However, some general guidelines are explained below.

The MUX bus (low–speed), with 10 address lines, supports up to 1024 devices. These devices typically consist of:

- multiperipheral controller (MPC) – provides eight RS–232C ports for user terminals, a programmable clock, a line frequency clock, and a line printer interface (parallel)
- additional line printer interfaces
- eight–line communication multiplexor – for additional RS–232C user ports
- real–time clock – provides more programmable clocks
- data communication devices (SSA, QSA, Procom and EDLC)

The DMA bus supports the controllers of the disk and tape devices described in the following section.

Mass Storage

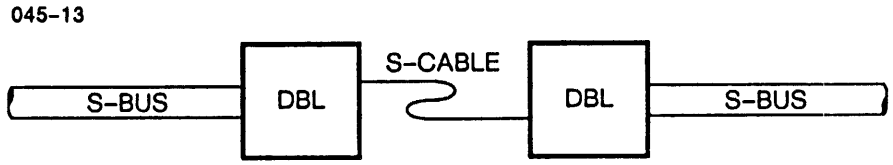
Concurrent offers customers a vast selection of disk and tape devices. For disk drives, system builders can choose from a variety of form factors: $5^1/_4$", 8" and 14". The smaller disk drives can be doubled on each rack mounting to conserve cabinetry and facility space. With formatted disk capacities in excess of 1GB, several gigabytes of user data can be stored in a single cabinet.

The responsiveness of our I/O subsystems are attributable in part to the performance of our disk drives. Our disk drives have burst transfer rates of up to 3MB/s and average seek times as quick as 16ms. Because certain applications can require extreme security measures, customers can also choose disks with removable media.

Customers have a similar selection of magnetic tape storage systems. Your 3280 and Micro3200 machines can be outfitted with both cartridge and reel–to–reel tape media. Cartridges insert into a $5^1/_4$" tape drive. Those choosing rack-mounted reel–to–reels select either phase encoded PE/1600bpi or group coded recording GCR/6250bpi tape formats. Storage capacities for the PE/1600bpi and GCR/6250 are approximately 40MB and 150MB (user data) respectively.

Distributed System Bus Architectures

The distributed bus link (DBL) further promotes the building block architectures of the systems. The DBL is a module that connects one S–Bus to another.



Although the concept sounds simple enough, the added functionality of the board itself provides the vehicle for a variety of applications.

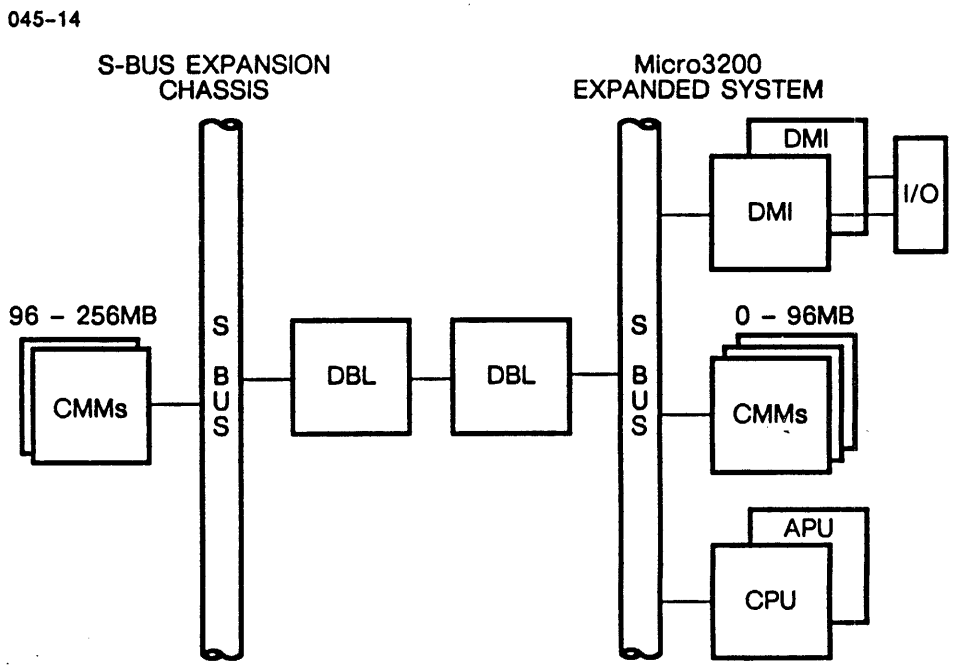
When installed in a system, DBLs can:

- expand single system configurations,
- implement several forms of shared memory,
- achieve shadow memory systems, and
- provide system redundancy and fault tolerance.

In short, DBLs allow the system builder to construct large computing complexes based on S–bus systems like the 3280 MPS and Micro3200 machines. The DBL achieves this functionality by selectively passing memory references, I/O operations, broadcasts, and interprocessor messages through the S–bus link.

Expanded Configurations for Single Systems

The requirements of some applications can exceed the configuration limits of single S–bus systems. For example, consider an upgraded application that uses a Micro3200 Expanded System that now requires more memory modules than the chassis can physically accommodate. By connecting another S–bus to the host system (via DBLs), we can easily augment the capacity of main memory to 256MB.



Shared and Shadowed Memory

DBL pairs can translate the memory addresses that they pass, providing them with the functionality required for implementing multisystem shared memory scenarios.

System designers assign memory access privileges for blocks of memory 64kB in size. The privileges are read, write, translate, tag, shadow, and don't cache. Installers can set one or several privileges depending on the type of access they want to enable. When translation is enabled, the designer also sets the desired translated address.

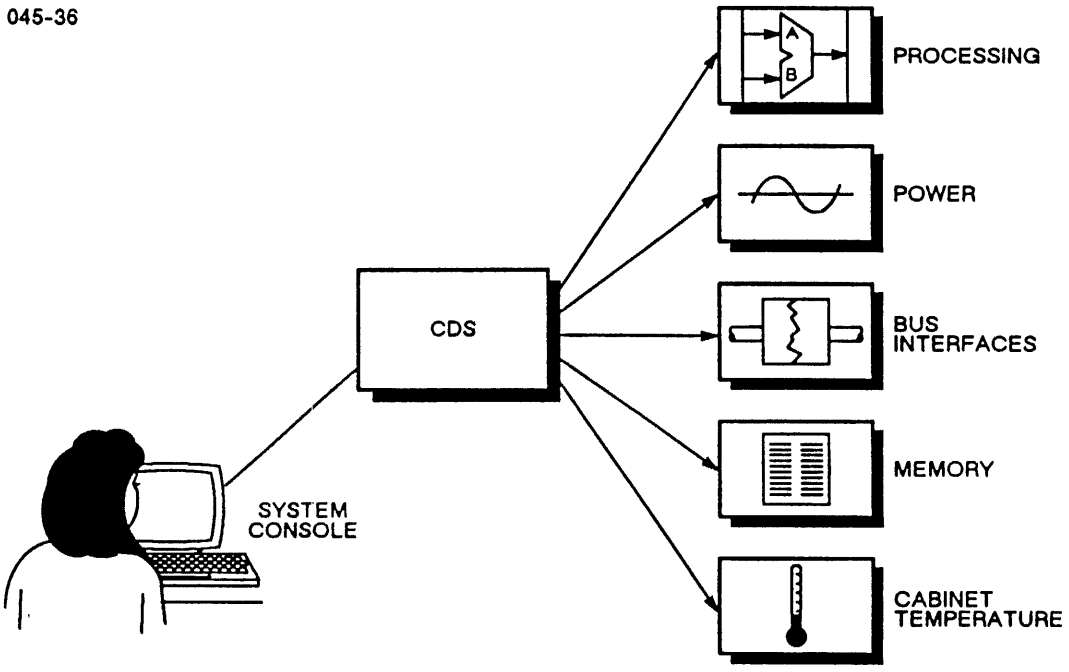
One method of achieving a multisystem shared memory configuration is by connecting your system to the Shared Memory Cabinet. By housing up to eight DBLs (8 half pairs) and a maximum eight CMMs, up to eight S–bus systems (i.e., 3280MPS and Micro3200 Expanded Systems) can share and exchange large amounts of data.

Control/Diagnostic System (CDS)

Console operators configure, control, and monitor their 3280 and Micro3200 systems with the aid of the control/diagnostic system (CDS). The CDS comprises an internal network of microprocessors. These microprocessors reside on various system modules and configure and monitor the module to which they are attached.

During installation, technicians configure system modules from the console through CDS command entry. For example, they assign starting addresses and interleaving factors to memory modules and loading parameters to the CPU.

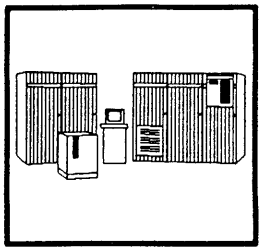
Because configuration is established by software rather than hardware (straps and switches), reconfiguring your system is fast and efficient. Furthermore, if a module fails, console operators can undertake corrective action without necessarily opening the system cabinet.



During system operation, the CDS acts as a monitor detecting internal malfunctions in real–time. If failures are detected, the CDS formats an explanatory message to the system console.

The CDS also allows you to determine the state of your system at the hardware level. For example, you can query memory modules to determine their starting addresses or to see if data errors have occurred. On the processor you can examine and modify all user–level registers (integer, single– and double–precision floating point). Power supplies reveal such data as backup battery voltages, regulated system voltages, and cabinet temperatures.

In short, the CDS provides you with a central resource from which to manage and control system hardware.



4

System Specifics and Configurations

In this chapter

We briefly discuss each system in the 3280 and Micro3200 family.

The following sections merely sketch the functional characteristics and physical make–up of the systems. For functional details on system software or on individual hardware components (e.g., processors, memory modules), please refer to Chapters 2 and 3.

Topics include:

- Micro3200 Compact Systems,
- Micro3200 Expanded System,
- Micro3200 MPS,
- 3280MPS, and
- 3280E MPS.

The Micro3200 Compact Systems

The Micro3200 Compact Systems are available in two general forms: a completely packaged system enclosed within a cabinet and an unbundled OEM version.

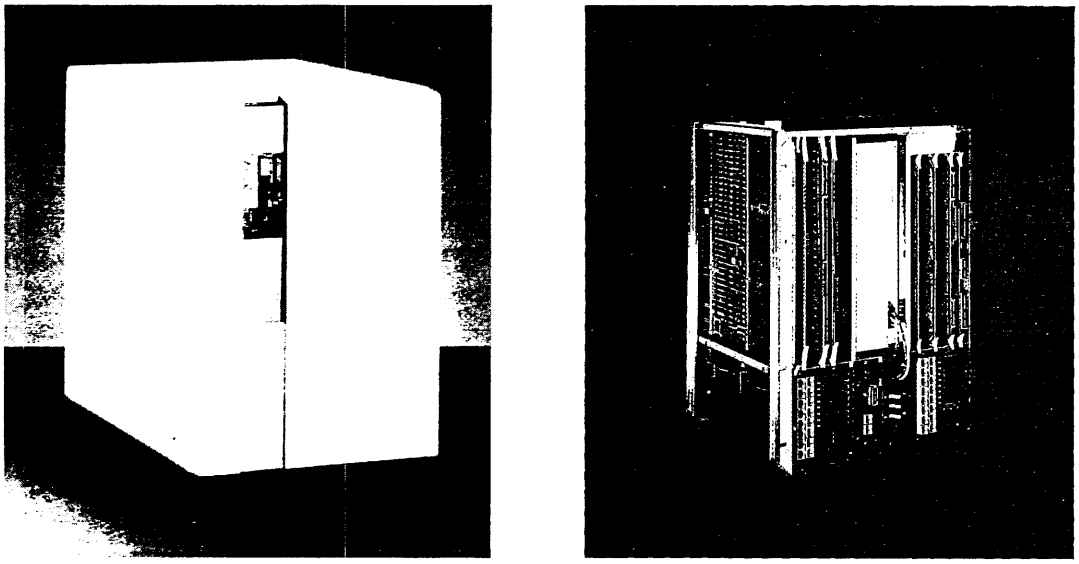


Figure 4–1. Micro3200 Compact Systems

Both versions can be equipped either with a MicroFive processor or a MicroThree.

The packaged version, called the Compact System, is a highly integrated machine. It confines all components, including disk and tape storage, to a 3.75 ft^2 footprint. At desk height, this complete and self–contained system lends itself most aptly to those facilities with limited space and power allocations. The OEM version,known as the Compact Chassis is rack–mountable and allows the system builder to control system packaging.

1045-43

System Snapshot – Micro3200 Compact System

Processing System
Host processor – MicroThree or MicroFive processor
Attached processor – None
Maximum attached processors – None
Maximum total processors – 1
Cache system – instruction & data caches
Data cache – 16kB, 2-way set associative
Instruction cache – 16kB, direct mapped
Cache access – 100ns
Memory read (cache miss) – 700ns (min)
Floating point hardware – On-board (standard)
Control store (user accessible) – none
Interrupt method – 4-level priority

System Bus
Type used – S-bus
Timing – synchronous (100ns cycles)
Path bandwidth – 64MB/s (usable read/write aggregate)

Memory System
Memory system – CMM, non- or 2-way interleaving
Number of modules – 3 (max)
Memory capacity – 96MB (max)

I/O System
Type of interface – DMI
Number of DMA channels – 1
Number of DMA (SELCH) ports – 8 (max)
DMA throughput – 10MB/s (max)

Software
Operating system – OS/32
Data communications – RTnet Family, IBM Gateways
Interactive processing - MTM or Reliance PLUS

System Organization

The Micro3200 Compact Systems, depicted in Figure 4–2, are S–bus based systems meaning that all memory and I/O operations pass via the S– bus. A single CPU is responsible for processing and I/O interrupt response. Memory, which comprises up to three composite memory modules (CMMs), can either be noninterleaved or two–way interleaved. The CMMs can be of any capacity and intermixed, provided the rules of interleaving are upheld.

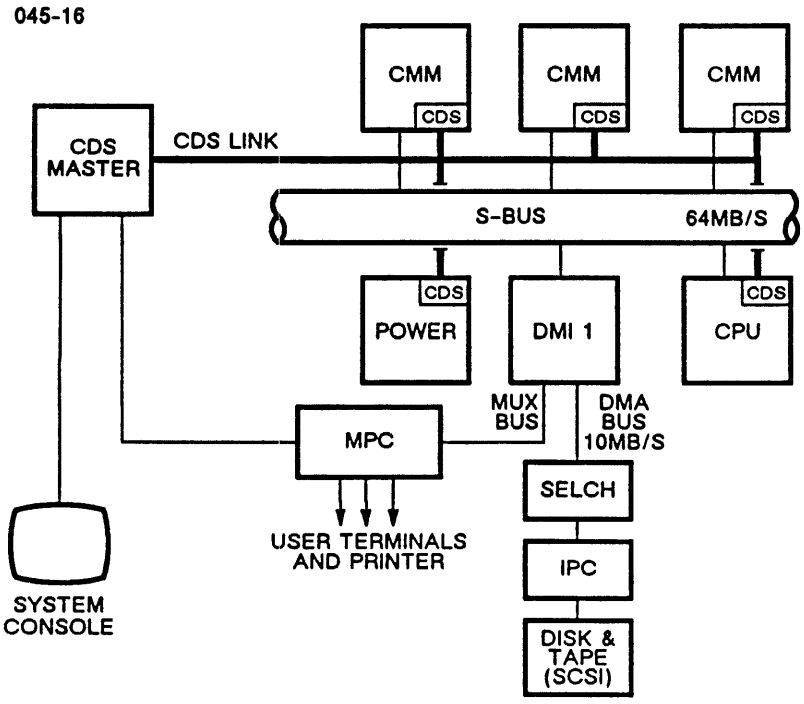


Figure 4–2. Micro3200 Compact Systems Block Diagram

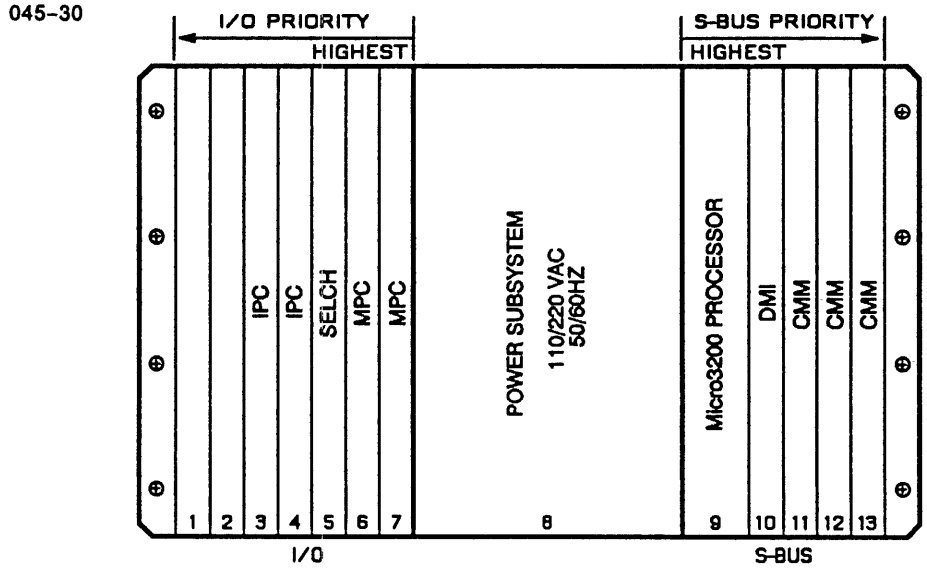
System I/O enters and exits the S–bus through the direct memory interface (DMI) module which can maintain a 10MB/s aggregate I/O bandwidth. In the Compact System, the intelligent peripheral controller (IPC) provides the interface and controls the accompanying SCSI mass storage devices. The Compact Chassis does not necessarily use the IPC nor SCSI devices. Rather, it allows the system builder to choose the storage devices and controllers best suited to their specific application.

In the Micro3200 Compact Systems, the control/diagnostic system (CDS) consists of an individual master module that polls its slave modules for status checks during system operation. CDS slaves in these systems are the memory modules (CMM), the CPU, and the power subsystem.

Power is delivered through a single power subsystem. This power subsystem fulfills the power requirements for all system components including SCSI disk and tape units (when installed). In addition, it can operate in back–up battery mode, thereby preserving memory during brief power outages. The power subsystem is a fully integrated unit and has no expansion options.

Physical Profile

All Micro3200 Compact Systems use a common backpanel that accepts the S–bus modules, the power subsystem, and I/O boards. This design reduces cabling and general complexity, providing a more reliable and serviceable system.



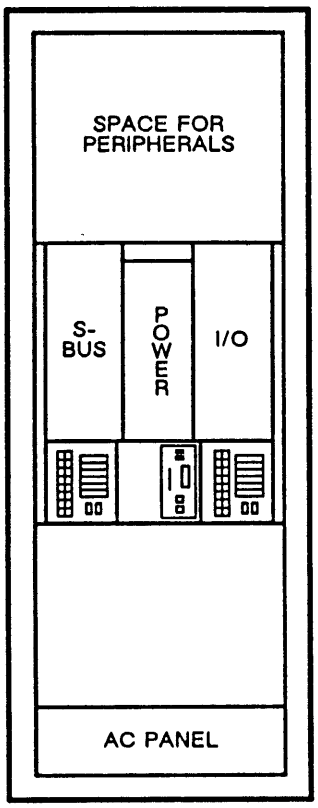
The S–bus portion supports at least one processor board, a CMM, and a DMI for I/O. Leaving two available slots, this minimal configuration can accommodate two more CMM's for increased RAM storage. Furthermore, through memory interleaving, you can also increase memory throughput.

The I/O portion of the backpanel supports up to seven 15" I/O boards. A minimal configuration contains an MPC, for console and printer interfaces, and a SELCH and two IPCs which provide the interface for the disk and tape drives. Three additional MPCs may be added to extend user terminal capacity to 32 users.

The Compact System resides within a single cabinet. Because the cabinet also houses the mass storage devices, no expansion cabinetry is necessary.

In terms of system packaging and options, the Compact Chassis offers more flexibility than the packaged version. For example, you can mount the Compact Chassis in our 71" cabinets, 56" cabinets, or any standard 19" (wide) Retma cabinet with adequate vertical clearance.

045-42



The OEM version also allows you to choose from our full range of disk and tape drives and controllers.

Peripherals

The maximum peripheral configuration for the Compact System consists of two cartridge tape drives and four disk drives. All peripherals conform to a $5^1/_4$" form factor and reside within the cabinetry. Using high density disks, the Micro3200 Compact System can provide over a gigabyte of disk storage.

The Compact Systems have two I/O connector panels that accommodate both synchronous and asynchronous devices. Combined, the panels provide 32 asynchronous terminal ports and 4 synchronous ports.

The Micro3200 Expanded System

The Micro3200 Expanded System provides the entry point to Concurrent's multiprocessing systems. With its maximum two–processor configuration and combined S–bus I/O backplane, the Expanded System provides a compact, yet powerful system solution.

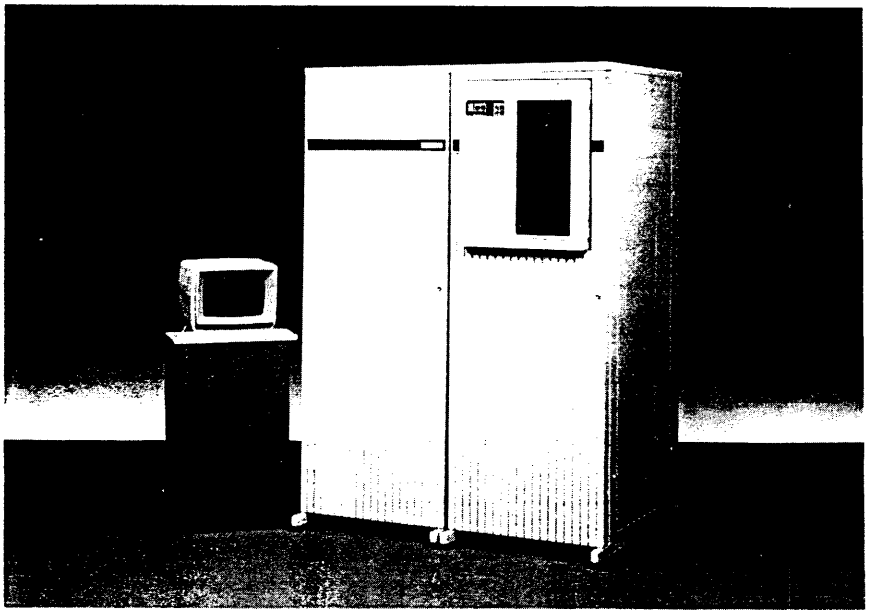


Figure 4–3. Micro3200 Expanded System

The Expanded System is not standardly equipped with any cabinet. Rather, it is an unbundled system offering that gives customers the ability to define the system packaging. This is an obvious advantage for the system builder who wants to use only proprietary packaging. Of course, you can also order the Expanded System fully packaged in one of our cabinets.

The Micro3200 Expanded System comes equipped either with a MicroFive processor or a MicroThree.

1045-44

System Snapshot – Micro3200 Expanded System

Processing System

Host processor – MicroThree or MicroFive processor
Attached processor – Same as host processor
Maximum attached processors – 1
Maximum total processors – 2
Cache system – instruction & data caches
Data cache – 16kB, 2-way set associative
Instruction cache – 16kB, direct mapped
Floating point hardware – On-board (standard)
Cache access – 100ns
Memory read (cache miss) – 700ns (min)
Control store (user accessible) – none
Interrupt method – 4-level priority

System Bus

Type used – S-bus
Timing – synchronous (100ns cycles)
Path bandwidth – 64MB/s (usable read/write aggregate)

Memory System

Memory system – CMM, non-, 2- or 4-way interleaving
Number of modules – 4 (max)
Memory capacity – 128MB (max)

I/O System

Type of interface – DMI
Number of DMA channels – 2 (max)
Number of DMA (SELCH) ports – 16 (max)
DMA throughput – 20MB/s (max)

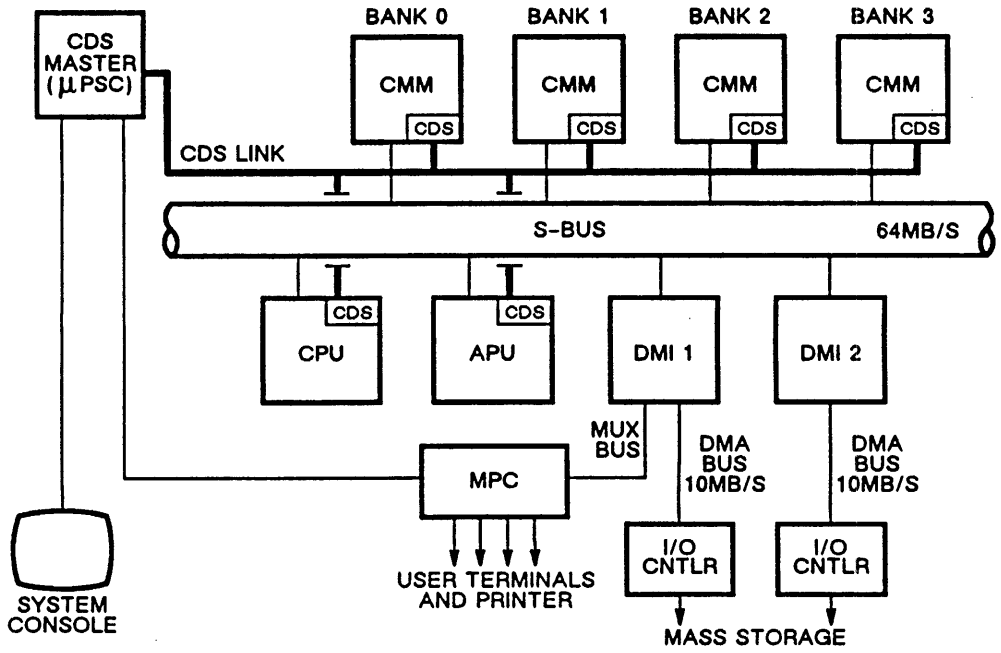
Software

Operating system – OS/32
Data communications – RTnet Family, IBM Gateways
Interactive processing – MTM or Reliance PLUS

System Organization

Similar to the Compact Systems, the Micro3200 Expanded System, shown in the following figure, is based entirely on S–bus architecture. All processing can be performed by a single MicroFive or MicroThree CPU. If higher processing power is desired, an APU can be plugged in easliy.

045-27



Note: 7 S–bus modules maximum

Figure 4–4. Micro3200 Expanded System Block Diagram

System memory comprises up to four CMMs that can be 4–way interleaved for maximum memory throughput. Using 32MB CMMs, the Expanded System can accommodate up to 128MB. CMMs can be of any capacity and intermixed, provided the rules of interleaving are upheld.

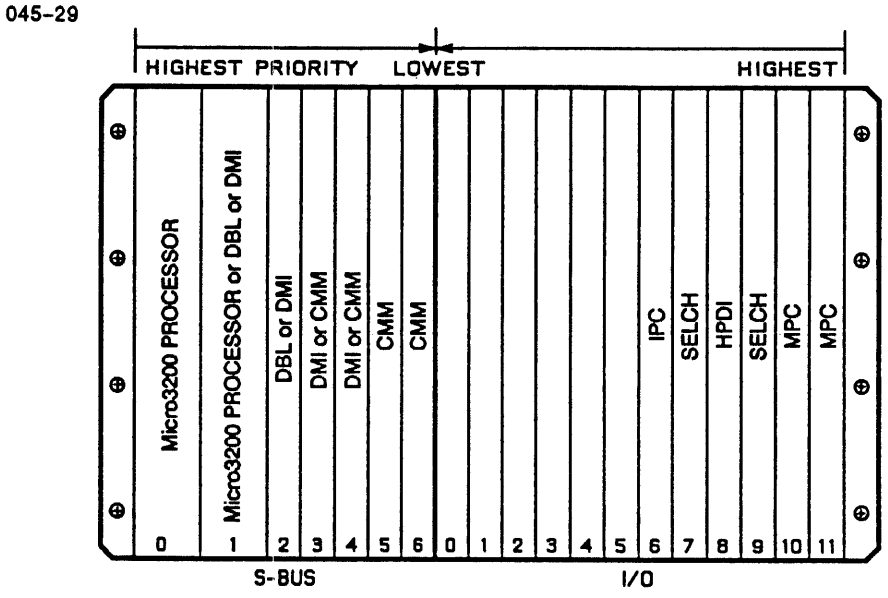
The Expanded System interfaces its I/O subsystem through the DMI. A maximum two DMIs can be installed, providing a maximum 20MB/s of I/O bandwidth. If, in the future, you need to increase the number of I/O devices and interfaces, the Expanded System can accommodate and connect to expansion I/O chassis and cabinets.

The CDS monitors critical system functions and consists of the CDS master and the slave modules. In this system, the CDS master is implemented on the micro–based power subsystem controller (PSC) where it monitors key power functions as well as the slave boards. The slave modules are the memory modules (CMM) and the processor(s).

The Micro3200 Expanded System deploys a modular power subsystem that can be expanded should system requirements grow.

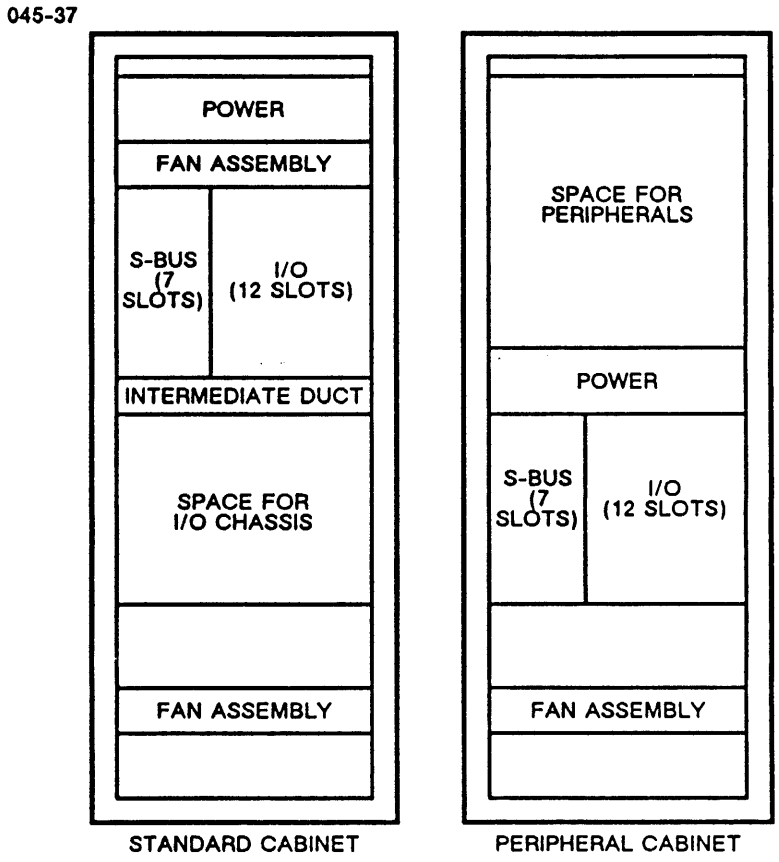
Physical Profile

The Micro3200 Expanded System deploys a hybrid S–bus/IO backpanel. The S–bus section supports up to seven S–bus modules which includes, at the least, a MicroFive or MicroThree CPU, a DMI for I/O, and a CMM.



This leaves four available slots in which you may place an additional processor, DMI or up to four more CMMs. To expand system resources further or share data with another system, you can also install a DBL.

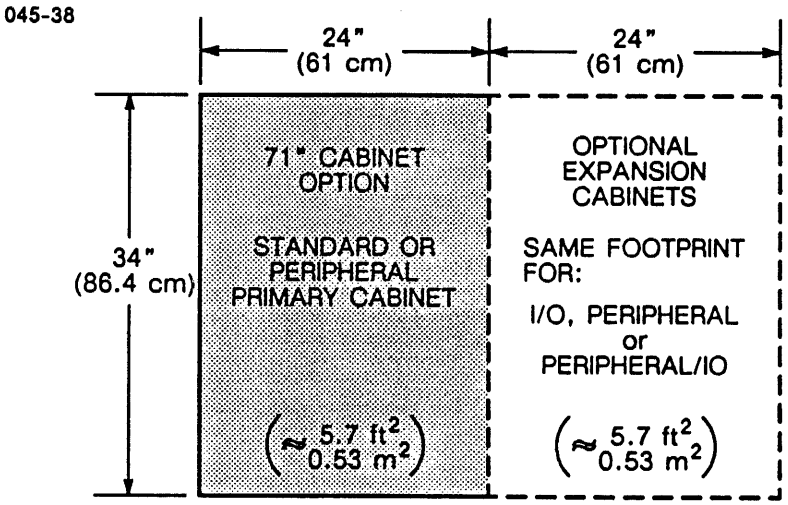
The I/O section of the backpanel can support up to twelve 15" boards. System builders may choose from a variety of I/O boards such as MPCs, for asynchronous communications, SELCHs and DMA controllers, for high–speed I/O, and communications interfaces for system–to–system data communications.



As stated previously, cabinetry for the Micro3200 Expanded System is optional. You can place the chassis in any 19" (wide) Retma cabinet with adequate vertical clearance or you can choose from two cabinet options offered by us: a plain 71" cabinet or a 71" cabinet capable of supporting peripherals.

The plain cabinet has space allotted for an I/O expansion chassis. The peripheral cabinet permits you to house most disk or tape devices as well as the rest of the system's components in the same cabinet. Such a configuration provides an extremely compact, yet expandable system.

The following figure shows the footprint of a Micro3200 Expanded System housed by our 71" cabinet options. Included are the footprints of other optional cabinetry.



The Micro3200 MPS Multiprocessor

The Micro3200 MPS offers full configuration flexibility by accommodating system expansion in all functional dimensions: processing, memory, and I/O.

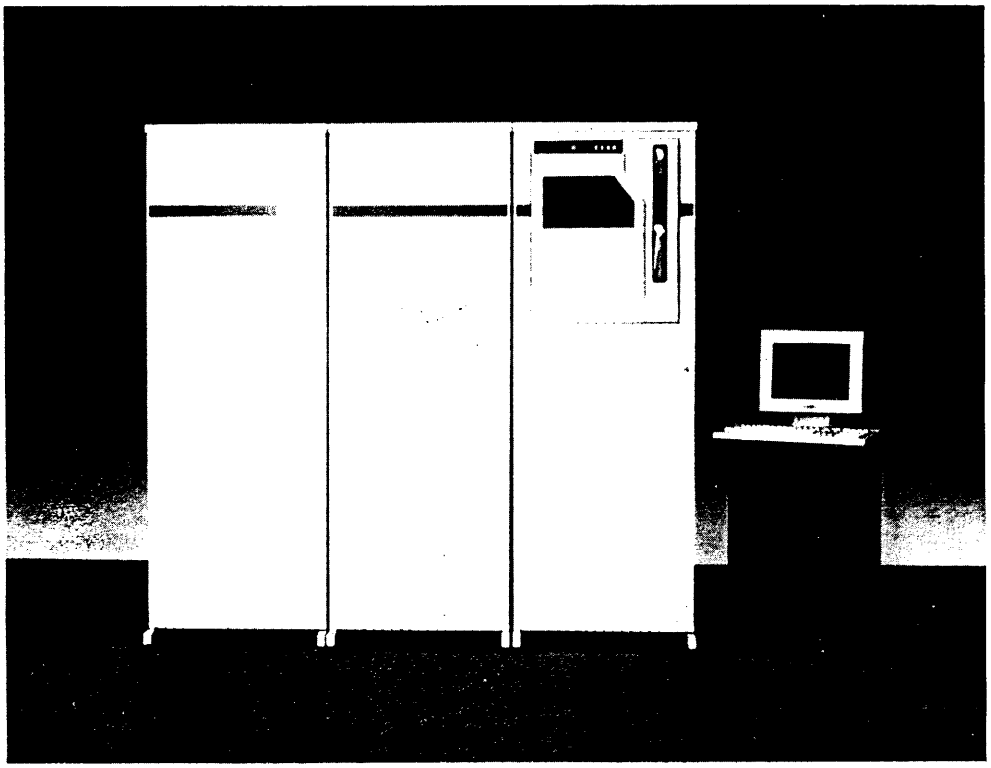


Figure 4–5. Micro3200 MPS

The Micro3200 MPS accepts either the MicroFive processor or the MicroThree. Up to six processors may be installed.

1045-47

System Snapshot – Micro3200 MPS

Processing System

Host processor – MicroFive or MicroThree processor
Attached processor – Same as host processor
Maximum attached processors – 5
Maximum total processors – 6
Cache system – instruction & data caches
Data cache – 16kB, 2-way set associative
Instruction cache – 16kB, direct mapped
Floating point hardware – On-board (standard)
Cache access – 100ns
Memory read (cache miss) – 700ns (min)
Control store (user accessible) – none
Interrupt method – 4-level priority

System Bus

Type used – S-bus
Timing – synchronous (100ns cycles)
Path bandwidth – 64MB/s (usable read/write aggregate)

Memory System

Memory system – CMM, 1-, 2- or 4-way interleaving
Number of modules – 8 (max)
Memory capacity – 256MB (max)

I/O System

Type of interface – DMI
Number of DMA channels – 2 to 4
Number of DMA ports – 32 (max)
DMA throughput – 40MB/s (max)

Software

Operating system – OS/32
Data communications – RTnet Family, IBM Gateways
Interactive processing – MTM or Reliance PLUS

System Organization

Similar to the other Micro3200 systems, the Micro3200 MPS is based entirely on an S–bus architecture. All memory requests and I/O signaling travels by way of the S–bus. In addition, the S–bus also supports interprocessor messages and broadcasts.

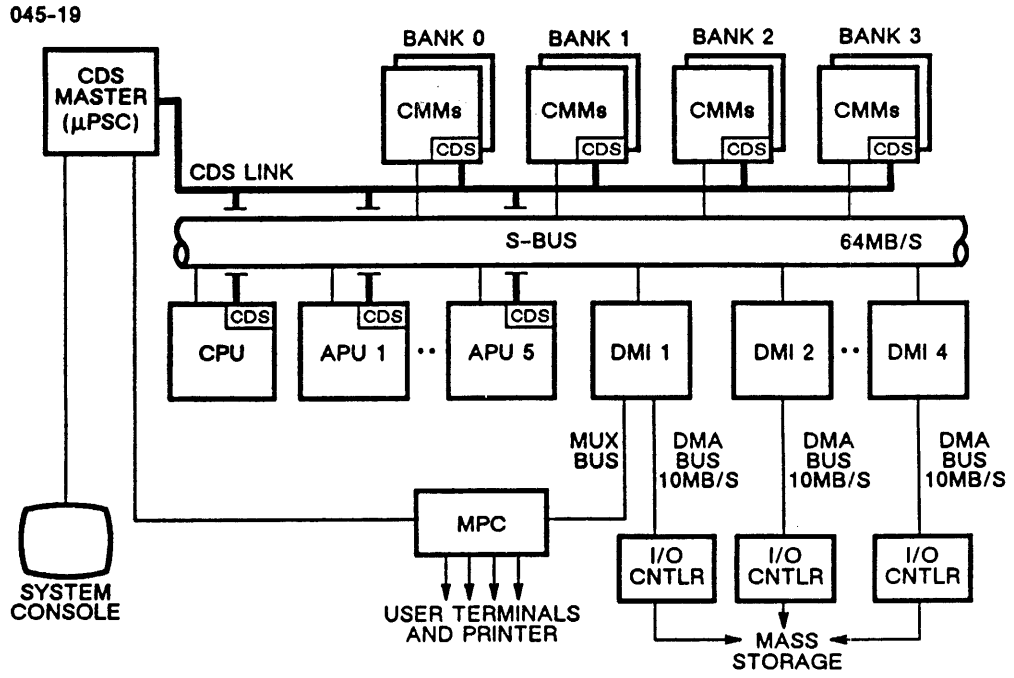


Figure 4–6. Micro3200 MPS Block Diagram

A maximum of six processors, including the CPU, perform the processing function of the system.

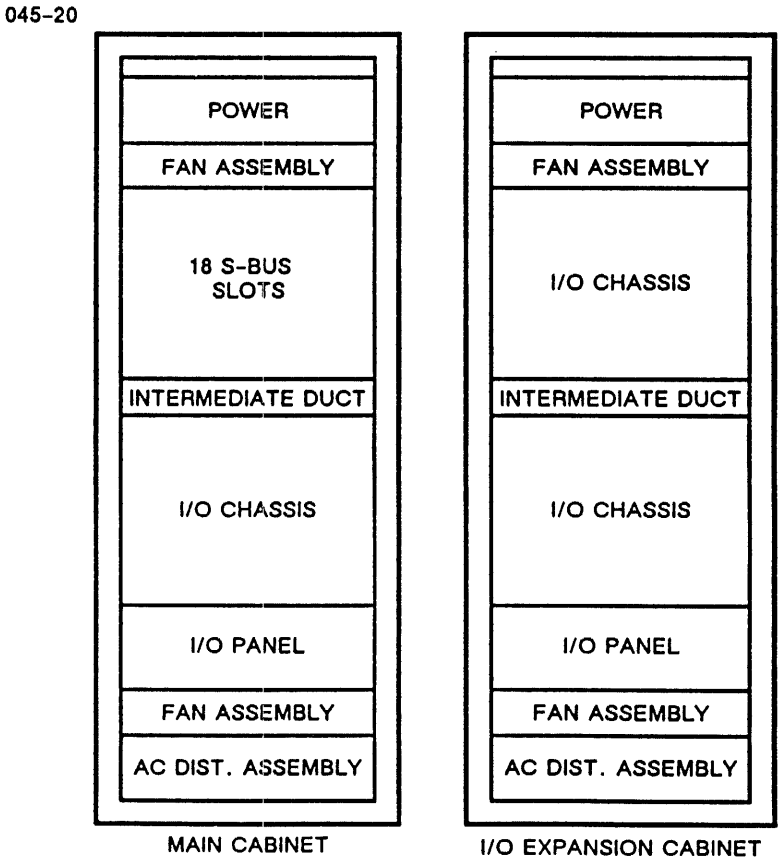
The memory system can range up to 256MB and can be noninterleaved, two–, or four–way interleaved for maximum memory throughput.

Like all other 3280 and Micro3200 machines, the Micro3200 MPS uses the DMI module as its I/O interface. By installing up to four DMIs, your system can attain a sustained 40MB/s I/O bandwidth.

The CDS master polls its slave modules for status checks and control during system operation. In Micro3200 MPSs, the CDS master is an integrated function of the micro–based power subsystem controller (μPSC). Slave modules are comprised of the Micro3200 processor(s) and the memory modules.

Physical Profile

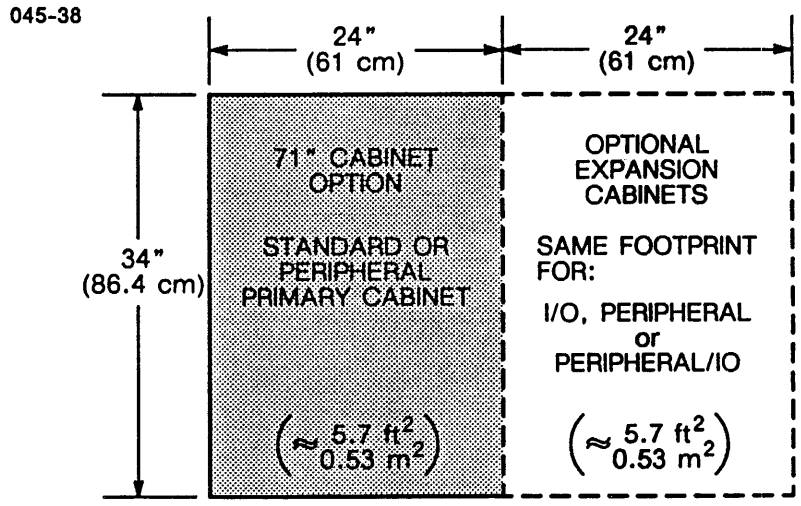
Excluding peripherals, the entire Micro3200 MPS processing system resides within a single 71" tall cabinet. Additional expansion cabinets are installed only for larger I/O configurations.



The CPU cabinet contains individual S–bus and I/O chassis. Power and cooling assemblies are included as well. The S–bus chassis provides 18 S–bus slots required by the processor(s), memory modules (CMM) and the I/O interface (DMI).

The I/O chassis comes equipped with a 20–slot backpanel that carries the MUX and DMA buses to the I/O boards. A maximum of 20 I/O controllers and interfaces can be installed. If additional I/O modules are required, the Micro3200 MPS can connect to I/O expansion cabinets, that each hold up to two 20–slot I/O chassis.

The Micro3200 MPS is an extremely space–conscious multiprocessing system. Excluding peripherals, an entire system can be confined to a single cabinet footprint (5.7 ft^2).



The 3280MPS Multiprocessor

The 3280MPS offers a slightly higher performance than the Micro3200 MPS.

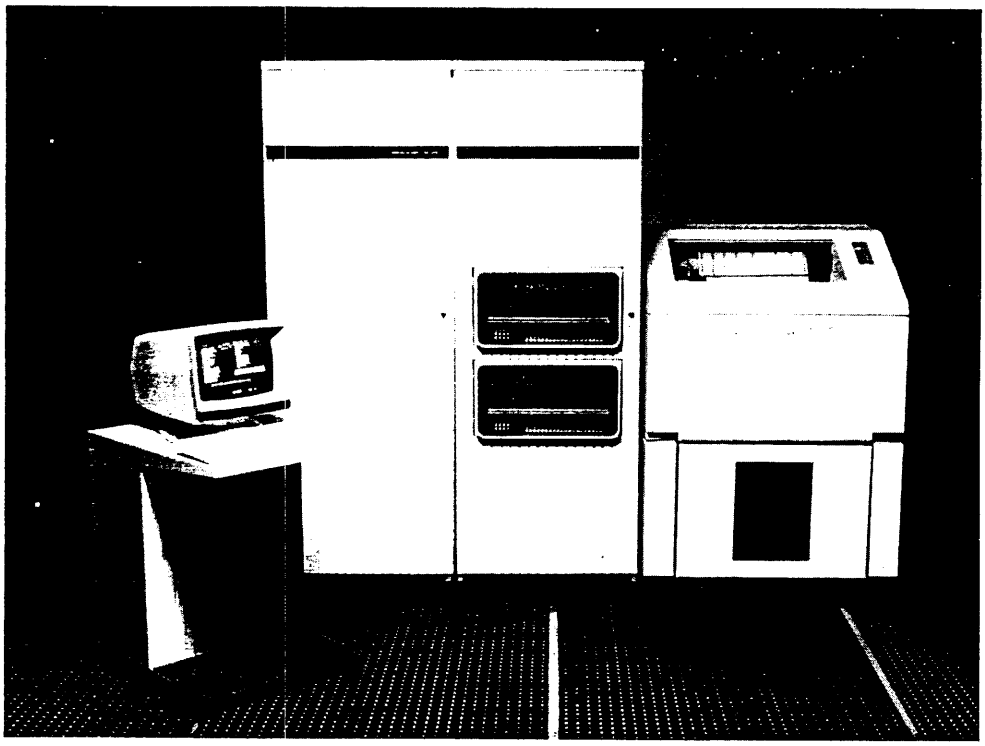


Figure 4–7. 3280MPS

Based on the 3280 processor, this system accommodates from one to six processors and provides a processing throughput of 6 to 36 MIPS.

I045-45

System Snapshot – 3280MPS

Processing System
- Host processor – 3280
- Attached processor – 3280
- Maximum attached processors – 5
- Maximum total processors – 6
- Cache system – instruction & data caches
- Data cache – 8kB, 2-way set associative
- Instruction cache – 8kB, 2-way set associative
- Floating point hardware – On-board (standard)
- Cache access – 100ns
- Memory read (Cache miss) – 700ns (min)
- Control store (user accessible) – 8K microinstructions
- Interrupt method – 4-level priority

System Bus
- Type used – S-bus
- Timing – synchronous (100ns cycles)
- Path bandwidth – 64MB/s (usable read/write aggregate)

Memory System
- Memory system – CMM, non- or 2-way interleaving
- Number of modules – 8 (max)
- Memory capacity – 256MB (max)

I/O System
- Type of interface – DMI
- Number of DMA channels – 4
- Number of DMA (SELCH) ports – 32 (max)
- DMA throughput – 40MB/s (max)

Software
- Operating system – OS/32
- Data communications – RTnet Family, IBM Gateways
- Interactive processing – MTM or Reliance PLUS

System Organization

Components of the 3280MPS are centered around the S–bus. With a 100ns clock cycle, the S–bus conveys all memory requests, I/O, and interprocessor messages.

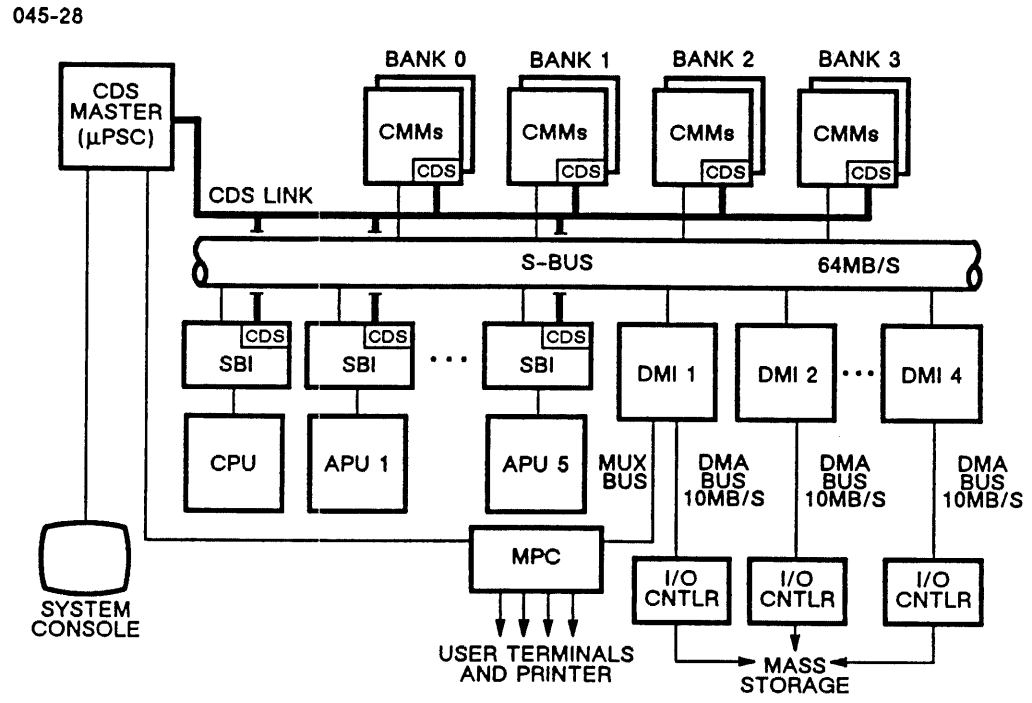


Figure 4–8. 3280MPS Block Diagram

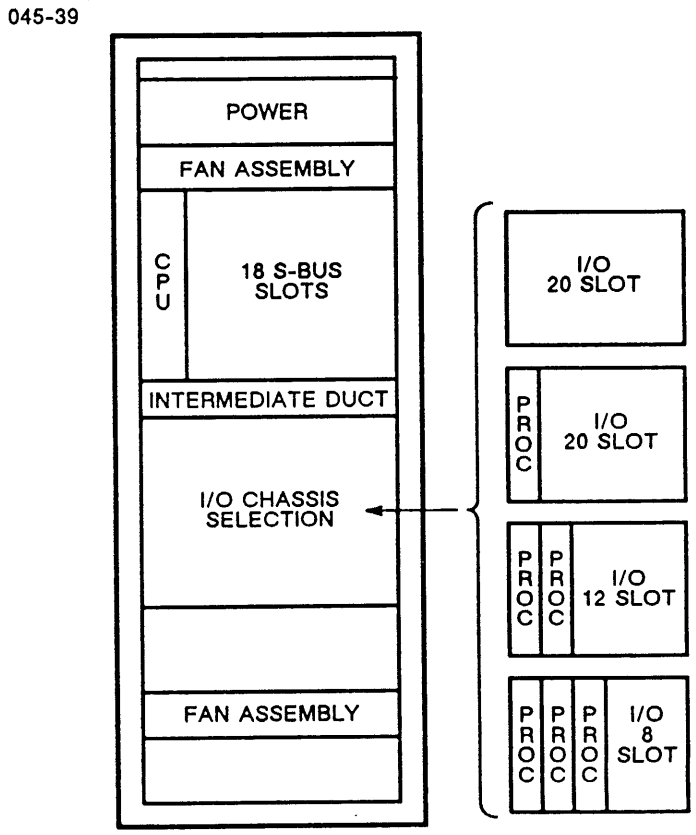
The 3280MPS accommodates up to six 3280 class processors. Each processor interfaces the system bus via the S–bus interface board.

System memory contains from one to eight CMMs of equal or varying capacities, and can range up to 256MB. To maximize throughput, memory can be two– or four–way interleaved.

The I/O subsystem interfaces the system through the DMI module. A maximum of four DMIs can be installed to provide 40MB/s of I/O bandwidth. The first DMI generates the CPU's MUX bus and a DMA bus. Subsequent DMIs generate additional DMA buses. The MUX bus is used for low–speed device interfaces such as the MPC which provides terminal, printer, and timer support. The DMA bus provides the data artery for the system's high speed devices.

Physical Profile

Regardless of an individual configuration, the 3280MPS always starts with a processor/S–bus cabinet. This cabinet is 71" tall and houses power and cooling assemblies and upper and lower PC board chassis.



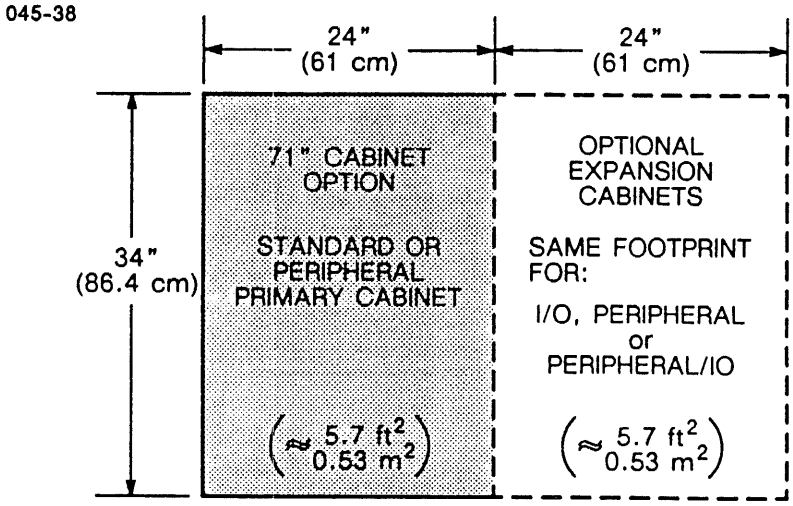
The upper chassis contains the CPU boards and 18 S–bus slots that hold the processor interface(s) (SBI), direct memory interfaces and memory modules. The lower chassis is some form of I/O chassis that may have between one and three CPU board sections depending on your systems requirements.

For larger configurations, the 3280MPS can accommodate a number of expansion cabinets, such as:

- processor expansion,
- I/O expansion, and
- peripheral cabinets.

Each of these cabinets is 71" tall. The processor expansion cabinet houses up to four processors in its upper chassis. The lower portion of the cabinet has room for a 20–slot I/O chassis. The I/O expansion cabinet can hold up to two 20–slot I/O chassis and the peripheral cabinet can accept a variety of mass storage devices in all form factors.

Generally, the 3280MPS is slightly larger than its Micro3200 counterpart; however, smaller configurations (not counting peripherals) can still be confined to the footprint of a single–cabinet.



The 3280E MPS Multiprocessor

The 3280E MPS processing system is the most powerful and versatile system we offer. Depending on the number of expansion modules installed, the 3280E MPS offers from 12 to 72 MIPS of processing performance and yields an I/O throughput of up to 120MB/s.

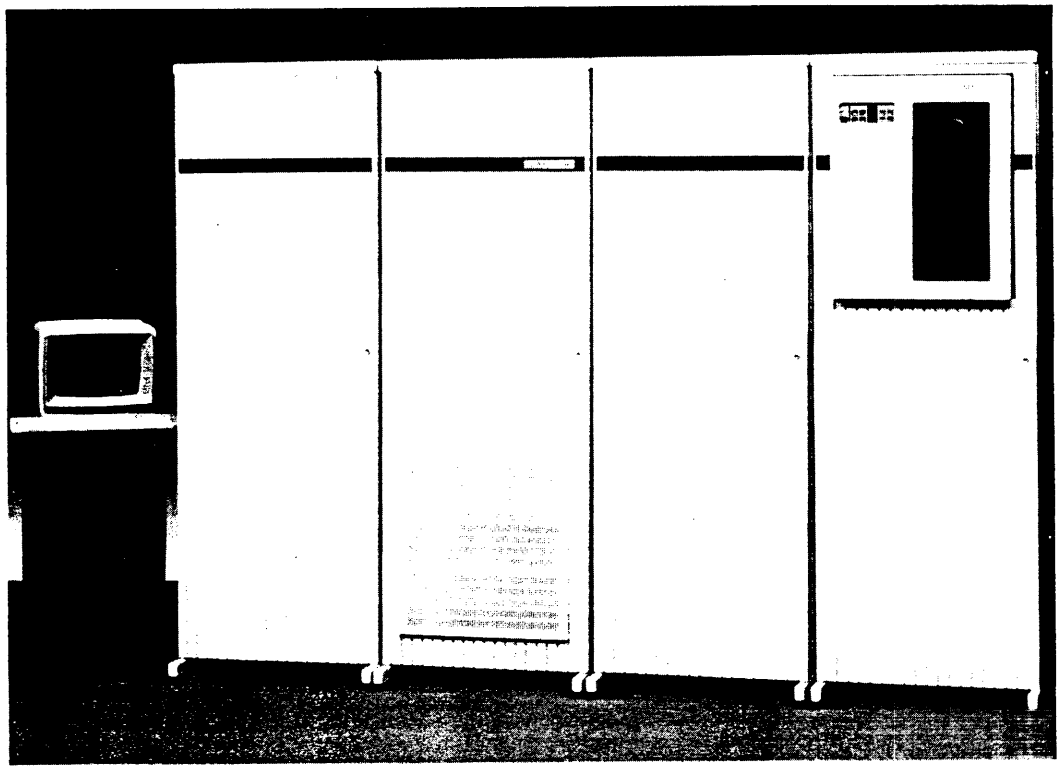


Figure 4–9. 3280E MPS

1045-46

System Snapshot – 3280E MPS

Processing System

Host processor – 3280
Attached processor – 3280
Maximum attached processors – 11
Maximum total processors – 12
Cache system – instruction & data caches
Data cache – 8kB, 2-way set associative
Instruction cache – 8kB, 2-way set associative
Floating point hardware – On-board (standard)
Cache access – 100ns
Memory read (cache miss) – 800ns (min)
Control store (user accessible) – 8K microinstructions
Interrupt method – 4-level priority

System Bus

Type used – E-bus and S-bus
E-Bus timing – synchronous (25ns cycles)
S-bus timing – synchronous (100ns cycles
E-Bus bandwidth – 256MB/s (usable read/write aggregate)
S-Bus bandwidth – 64MB/s (usable read/write aggregate)

Memory System

Memory system – EMM, 2- or 4-way interleaving
Number of modules – 8 (max)
Memory capacity – 128MB (max)

I/O System

Type of interface – DMI
Number of DMA channels – 2 to 12
Number of DMA (SELCH) ports – 36 (max)
DMA throughput – 120MB/s (max)

Software

Operating system – OS/32
Data communications – RTnet Family, IBM Gateways
Interactive processing – MTM or Reliance PLUS

System Organization

Unlike the other systems, the 3280E MPS is based on a hybrid E–bus/S–bus architecture.

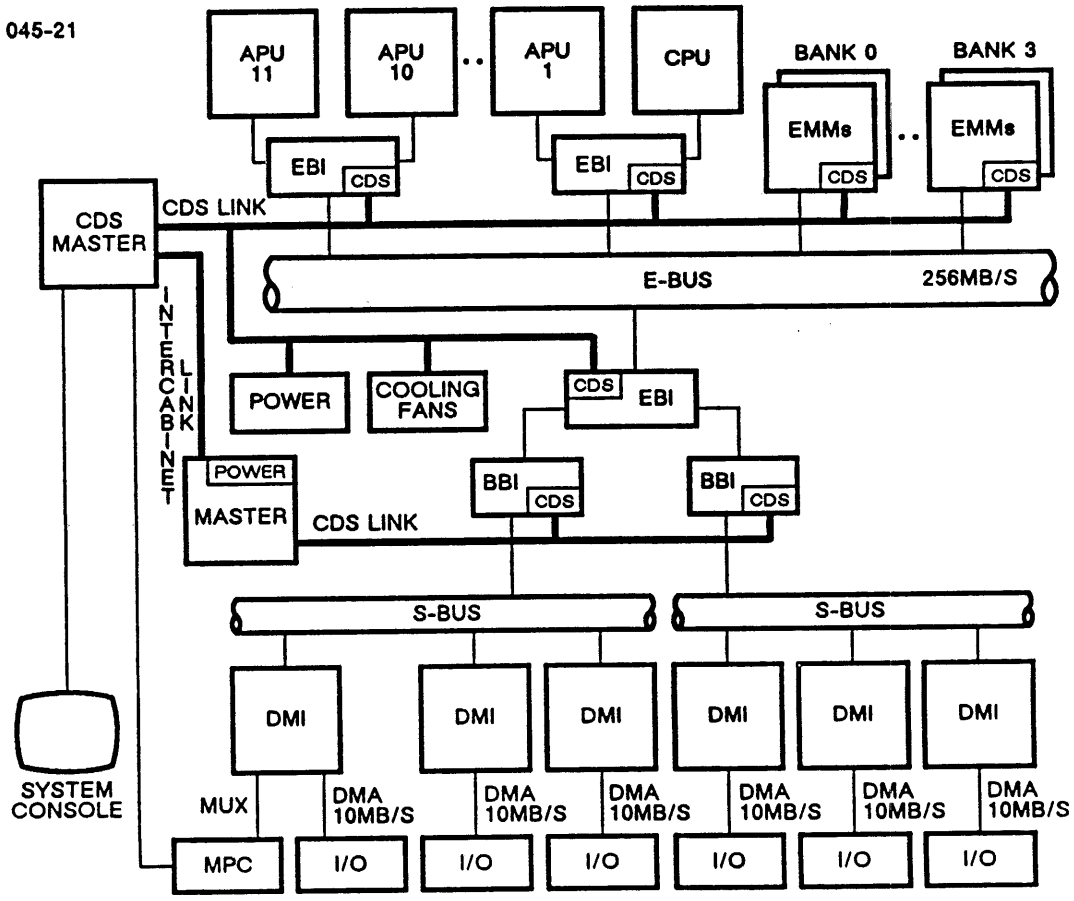


Figure 4–10. 3280E MPS Block Diagram

The E–bus supports all system operations including, memory requests, interprocessor messages, broadcasts, and I/O. The S–bus, in this system, acts only as a conduit for I/O operations.

The main functional difference between the 3280E MPS and the previously described systems is the E–bus and the attached memory system. Using emitter–coupled logic (ECL), the memory modules (EMMs), driven by a 40MHz clock, provide twice the memory throughput of the TTL–based CMMs. When memory is four–way interleaved and the system is at its maximum configuration, the E–bus' usable aggregate bandwidth of 256MB/s is realized.

Processing is performed by the 3280 processor. This processor connects to the E–bus through the E–bus interface (EBI) module. The EBI provides two B–bus ports, one for each processor. By driving the ports 180° out of phase, processor/EBI contention is eliminated.

The minimal 3280E MPS processor configuration consists of two 3280–type processors (CPU and APU) operating from a single EBI. Additional APUs and their EBIs can be installed to augment the system's processing capabilities.

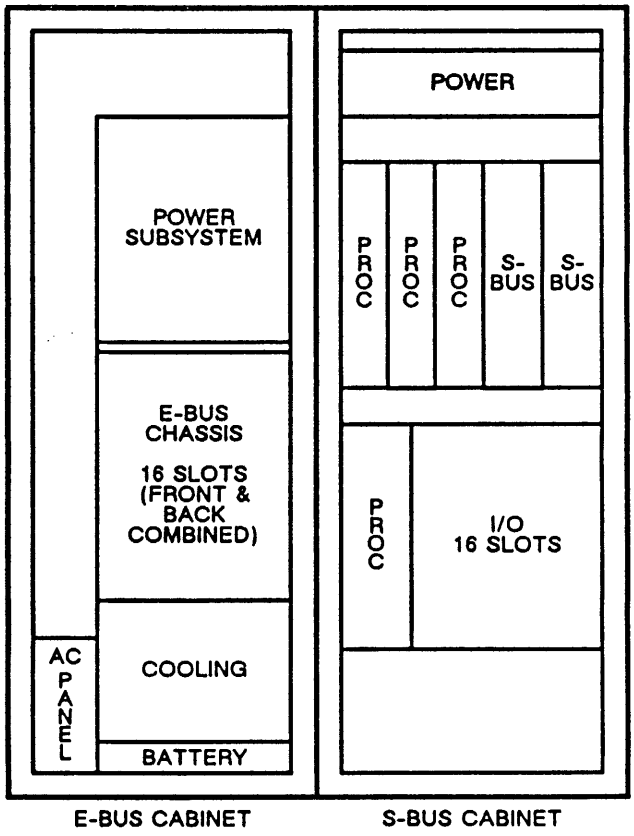
In the previously described systems, the DMI was the sole I/O interface to the system. However, in the 3280E MPS system, memory and processing resides on the E–bus; therefore, I/O must travel the E–bus as well as the S–bus. Bridging these buses are two modules: the EBI mentioned previously and the B–bus interface (BBI). The BBI translates S–bus signals into the B–bus signals required by the EBI. In turn, the EBI translates B–bus data into E–bus data. Again, because the EBI provides two B–bus ports, up to two BBIs, consequently two different S–buses can be attached to an EBI.

Using two EBIs, the 3280E MPS can accommodate up to four S–buses. With three DMIs on each of the four S–buses, the 3280E MPS can sustain a 120 MB/s I/O bandwidth.

Physical Profile

The minimum 3280E MPS configuration requires two cabinets, not including peripherals.

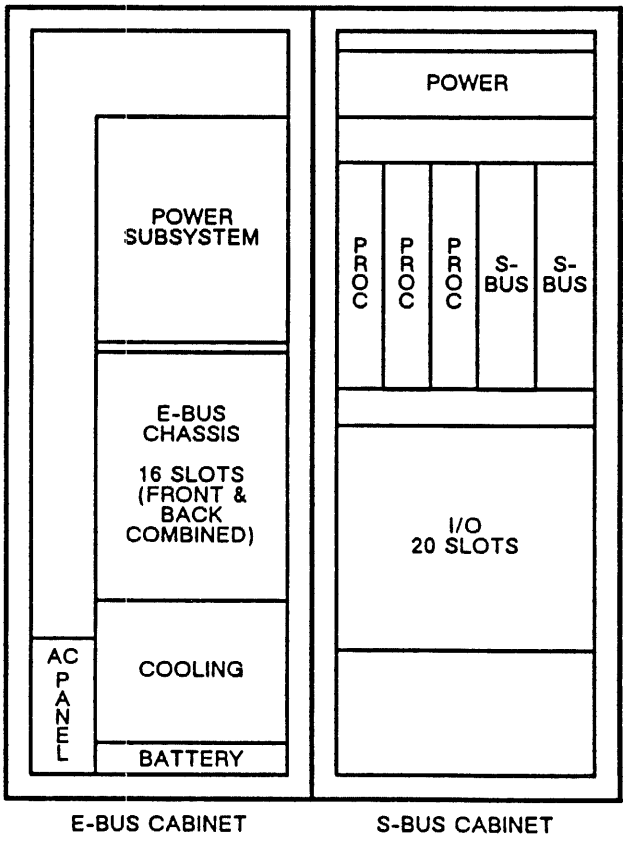
045-22



The E–bus cabinet houses the E–bus, the memory modules (EMMs), and the E–bus interfaces (EBIs) used by the processors and system I/O. Power and cooling are included as well. The E–bus resides on a printed circuit board that acts as a centerplane in the cabinet. The EMMs and the EBIs are inserted from the front and rear of the cabinet.

The adjacent S–bus cabinet houses a processor/S–bus chassis and a lower chassis which, depending on the application, is either another processor/S–bus chassis or an I/O chassis. The S–bus chassis provides three sets of processor slots and dual 5–slot S–buses. The processor slots hold the system's processor boards which cable to their respective EBIs in the E–bus cabinet. Each 5–slot–S–bus accommodates one BBI and from one to three DMI modules. If the application demands three or four S–buses for extremely high I/O capacity, a second dual S–bus chassis can be installed in the lower portion of the cabinet. Naturally, this second S–bus chassis can accept three more processors in addition to the dual S–buses.

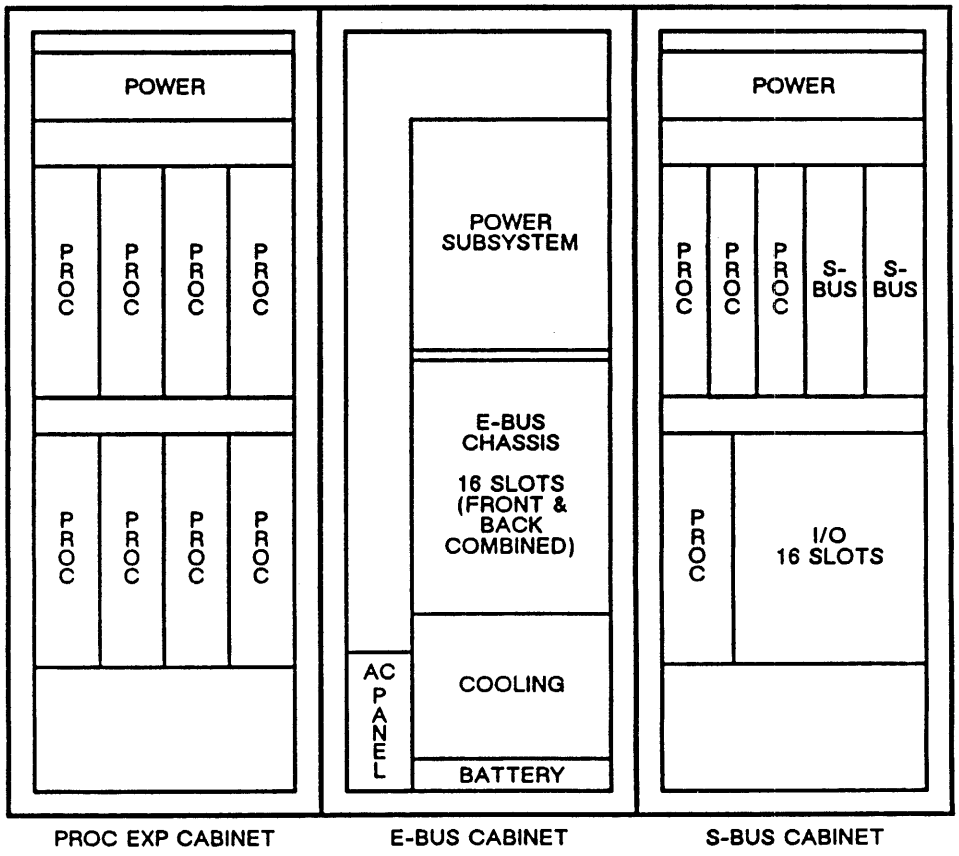
045-23



If you elect to place an I/O chassis in the lower chassis position you can choose one of two I/O chassis options: a 1–processor/16–slot I/O chassis or a 20–slot I/O chassis. The selection you make is application–dependent.

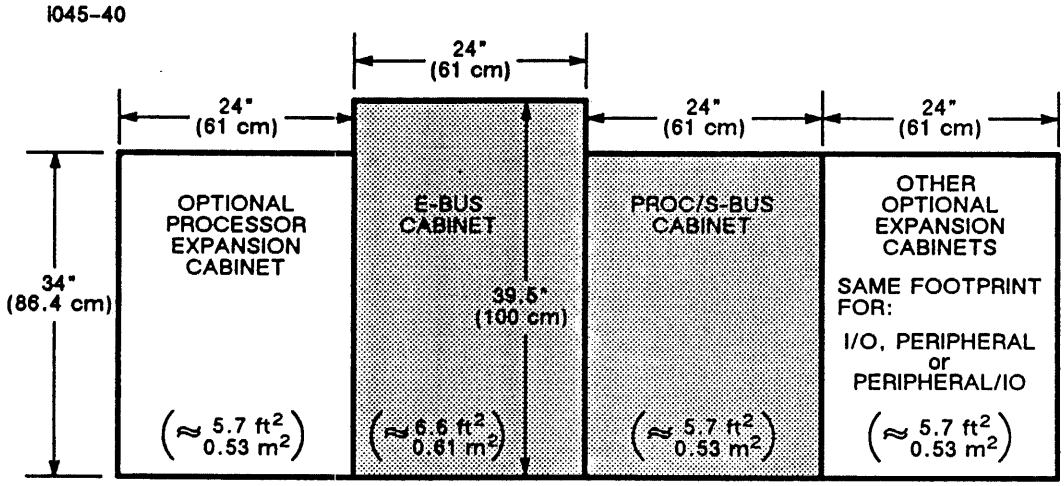
Depending on the processing requirements of your application an expansion processor cabinet may be necessary.

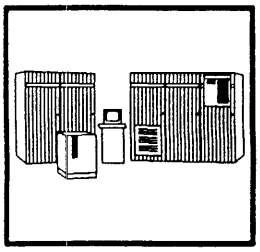
045-24



The expansion processor cabinet can house up to two chassis plus power and cooling. Again, all processors connect to their EBIs via intercabinet cabling.

The 3280E MPS, though our most powerful system, maintains a space–conscious footprint. Excluding peripherals, a four–processor, dual S–bus system requires only two cabinets or 12.3 ft.^{2} of floor space. A maximum configuration, comprising 12 processors and four S–buses can be housed in four cabinets (excluding peripherals). The four–cabinet configuration has a 23.7 ft^2 footprint.





5

Concurrent Services

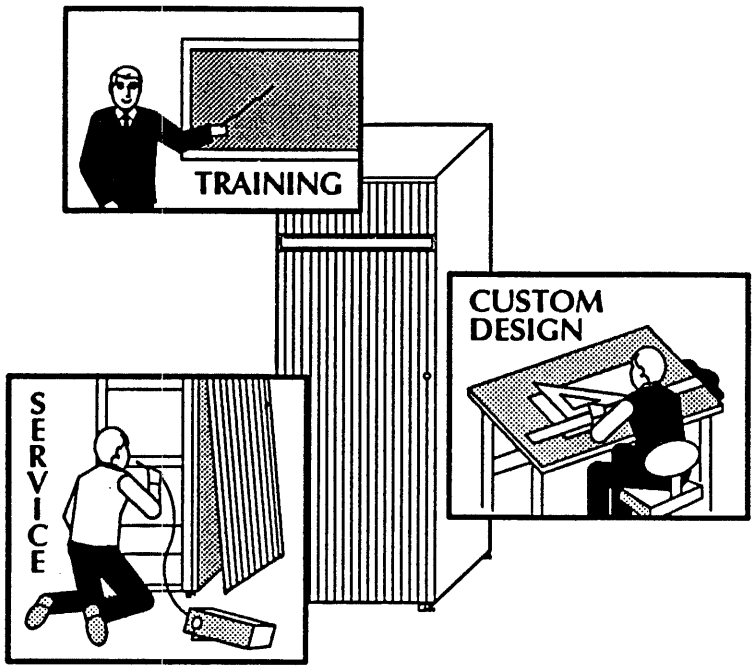
In this chapter

We profile three organizations; each offering a range of services that can assist you in getting the most from your computer system. If you want to contact any of these organizations, please see your sales representative.

Topics include:

- the services available from our Customer Service Organization,
- the types of courses offered by our Training Center, and
- the custom and semi–custom products of the System Products Division.

045-41



Customer Service

Our Customer Service organization provides customers with comprehensive technical support. Customer Service safeguards your computer hardware with maintenance contracts ranging from limited service, for those customers requiring minimal protection, to extensive coverage for those installations requiring extremely high availability and immediate response times. For complex installations requiring full–time coverage and immediate response, Customer Service can station a service engineer at your site through its Resident Service Program.

To answer software issues, Customer Service offers a selection of software maintenance contracts plus the Technical Assistance Center (TAC). If software problems arise, our TAC center assists you over the telephone. If the problem or the recommended solution is too complex for telephone procedures, TAC can send a software specialist to your site. You can also elect the Remote Diagnostic Service option through which TAC, from their facilities, can take immediate measures without the delays of a service visit.

Understanding Your System

Concurrent offers a wide selection of training courses to customers who want to familiarize themselves or their employees with specific tasks. Through classroom lectures and hands–on lab experience, students gain a comprehensive understanding of topics such as:

- OS/32 programming and operations,
- Data communications,
- Reliance PLUS, and
- Assembly and all supported high–level languages.

At the conclusion of the course, students return to work with documentation and printed classroom tutorials to assist them with future inquiries.

For those of you with specific training requirements, our training organization can design a customized course targeted specifically to your needs.

Classes are usually held at one of Concurrent's training centers; however, our instructors can travel to your facility to deliver on–site courses.

Custom Engineering

Sometimes the implementation of standard products alone cannot satisfy the requirements of an application. This is typical of systems purchased by value–added resellers (VARs) who usually add custom software and/or hardware to the systems they purchase.

Designed to complement the standard product offering, the System Products Division (SPD) offers a product line of its own. These products typically provide unique solutions for special applications. For example, the solid state disk that has neither the seek time of standard disk heads nor the rotational latency of disk media provides the ideal answer for frequently accessed files such as run–time libraries and database directories. SPD also offers an impressive array of industry standard bus interfaces such as:

- VME/VERSA,
- MIL–STD–1553,
- IEEE 488,
- ARINC 429–7, and
- NTDS

These interfaces resolve compatibility issues between specialized hardware and Series 3200 computer systems.

In addition to its standard products, SPD extends its talents to customers whose applications require exclusive attention. If technical issues arise, SPD can send a sales support group to evaluate the concern and define the solution.

Their custom design capabilities span several disciplines, namely: hardware, software, documentation, and mechanical engineering. The hardware and software engineering groups ensure a satisfactory solution, whether it requires a modification to a standard product or a complete custom design. Their documentation group offers custom documentation which, if requested, fulfills military specification requirements. If a standard product is functionally agreeable, but requires some physical changes, mechanical engineering can alter its dimensions and redefine product packaging. Furthermore, if the operating environment is prone to excessive vibration, they can ruggedize the equipment, enabling it to operate in abusive environments.

All custom designed products are fully backed and supported by our SPD and Customer Service organizations. In addition, SPD completely documents every custom product. In short, purchasers of custom products receive the same quality, support, and assurance that they expect of standard products.

A

System Documentation

In this Appendix

We discuss system documentation. Accompanying your 3280 or Micro3200 system is a set of documentation that gives you a comprehensive understanding of your system. These manuals address all levels of responsibilities, for facility planners, system administrators, system operators, and system level and application level programmers. For site service personnel, installation and configuration manuals are included as well. These manuals assist you in not only learning about your system, but how to tap its resources to achieve optimal performance. Your sales representative can help you order any documentation.

Many of the manuals listed in the following pages are available in packages. A documentation package is a collection of individual documents, which describe another aspect of a common topic or product. Some manuals serve as references for syntax and command usage, and others provide tutorials for accomplishing application related tasks. By ordering documention packages, you are provided with all of the relevent information specific to your inquiry.

OS/32 Documentation

The following sections identify the documentation that is available to users of OS/32.

OS/32 – System Operations

The manuals in this category cover topics such as generating your operating system, using the system console, establishing and maintaining user account structure, and administering disk storage.

Operator Reference

- *Using CDS to Power Up, Bootload, and Configure Your System*
- *CDS Command and Error Message Descriptions*
- *Multi–Terminal Monitor (MTM) Primer*
- *OS/32 MTM System Planning and Operator Reference Manual*
- *OS/32 Operator Reference Manual*
- *Environment Control Monitor (ECM/32) Reference Manual*
- *OS/32 Operator Pocket Guide*
- *OS/32 Operations Primer*

System Utilities

- *Guide to Using the System Performance Monitor (SPM)*
- *System Generation/32 (SYSGEN/32) Reference Manual*
- *OS/32 Fastback Reference Manual*
- *OS/32 System Support Utilities Reference Manual*
- *OS/32 Fastcheck Reference Manual*
- *OS/32 Fastcopy Reference Manual*

General Utilities

- *OS/32 Copy Reference Manual*
- *Source Updater User Guide*

- *OS/32 Patch Reference Manual*

- *OS/32 Library Loader Reference Manual*

OS/32 – System Level Programming

These manuals provide system level programmers with information on instruction sets, operating system services, task linkage, virtual task management, microprogramming, and data communications. These manuals along with the appropriate language documents provide you with a comprehensive systems programming resource.

- *Writable Control Store (WCS) Reference Manual*

- *OS/32 Link Reference Manual*

- *System Level Programmer Reference Manual*

- *OS/32 Supervisor Call (SVC) Reference Manual*

- *3280 and Micro3200 Quick Reference*

- *A Summary of OS/32 SVCs and Macro Calls*

- *Guide to Writing OS/32 Drivers Manual*

- *3280 and Micro3200 Instruction Set Reference Manual*

- *Microprogramming Reference Manual*

- *Microprogramming User Guide*

OS/32 – Application Level Programming

- *OS/32 Supervisor Call (SVC) Reference Manual*
- *OS/32 Link Reference Manual*
- *Multi–Terminal Monitor (MTM) Reference Manual*
- *3280 and Micro3200 Quick Reference*
- *OS/32 Edit User Guide*
- *Screen Editor User Guide*
- *OS/32 Application Level Programmer Reference Manual*

OS/32 – Data Communications

- *OS/32 Asynchronous Communications Reference Manual*
- *OS/32 Data Communications Reference Manual*
- *OS/32 Character Synchronous Communications Reference Manual*
- *OS/32 Network Drivers Program Reference Manual*

OS/32 Languages – FORTRAN (Package)

- *FORTRAN VII Reference Manual*
- *Symbolic Debug User Guide*
- *OS/32 System Support Run–Time Library (RTL) Reference Manual*
- *FORTRAN VII User Guide*

– *System Mathematical Run–Time Library Reference Manual*

OS/32 Languages – C³Ada

Two documentation packages are available for the C³Ada Language Environment.

C³Ada Debugging and Tuning Tools

– *Guide to Debugging Programs in the C³Ada Environment*

– *Guide to Monitoring Programs in the C³Ada Environment*

– *Guide to Using Tools for Analysis and Debugging in the C³Ada Environment*

– *C³Ada Debugging Environment Reference Manual*

C³Ada Compiler and APSE Tools

– *Introduction to the C³Ada Environment*

– *Guide to Developing Programs in the C³Ada Environment*

– *Guide to Using Program Development Tools in the C³Ada Environment*

– *C³Ada Environment Summary*

– *Run–Time Support Reference for the C³Ada Environment*

– *System Mathematical Run–Time Library (RTL) Reference*

– *OS/32 System Support Run–Time Library (RTL) Reference Manual*

– *Reference Manual for the Ada Programming Language*

– *Software Engineering with Ada*

- *Understanding Ada*

OS/32 Languages – C (Package)

- *OS/32 C Programming Manual*
- *System Mathematical Run–Time Library (RTL) Reference Manual*
- *OS/32 C Programming Encryption Manual*

OS/32 Languages – CAL

- *Common Assembly Language (CAL/32) Reference Manual*
- *OS/32 AIDS User's Guide*
- *CAL Macro/32 Processor & Macro Library Utility Reference Manual*
- *System Macro Library Reference Manual*
- *3280 and Micro3200 Instruction Set Reference Manual*

OS/32 Languages – Other High Level

- *BASIC Level II Reference Manual*
- *RPG II Reference Manual*
- *Pascal Plus User Guide for OS/32*
- *COBOL for OS/32 Reference Manual*
- *CORAL 66 Reference Manual*

Reliance Plus Documentation

The following manual titles identify the documentation set for Reliance PLUS.

- *Reliance in Action*
- *Reliance Master Index*
- *Reliance Design Guide*
- *Introduction to Reliance*
- *Installing Reliance*
- *Reliance Administrator's Guide*
- *Using a Reliance Terminal*
- *Developing Reliance Programs*
- *Developing Reliance Screen Forms*
- *Data Dictionary Reference Manual*
- *Reliance Productivity Aid (RELPA) Reference Manual*
- *ITC For Programmers*
- *ITC Technical Reference Manual*
- *Source Librarian User Guide*
- *Reliance/COBOL Programmers Pocket Guide*
- *Reliance/FORTRAN Programmers Pocket Guide*

- *Reliance/C Programmers Pocket Guide*
- *Reliance Reply Codes Pocket Guide*
- *DMS/32 for Programmers*
- *Data Management System/32 (DMS/32) CAL Programmers Reference Manual*
- *DMS/32 Technical Reference Manual*
- *Using DSS*
- *DSS for Programmers*
- *DSS Reference Manual*
- *Reliance Messages*

Reliance Builder Documentation

The following manual titles identify the documentation set for Reliance BUILDER.

- *Reliance Builder Adminstrator's Guide*
- *Getting Started with Menu Builder*
- *Getting Started with Transaction Sequencer*
- *Getting Started with Application Builder*
- *Using Menu Builder*
- *Menu Builder Reference Manual*
- *Using Transaction Sequencer*

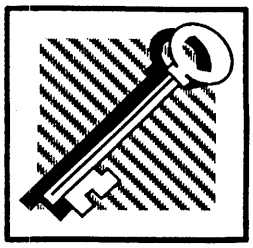
- *Transaction Sequencer Reference Manual*
- *Using Application Builder*
- *Using Application Builder*
- *Application Builder Reference Manual*
- *Writing Functions for Application Builder*
- *Reliance Builder Interfacing Guide*

Hardware Documentation

The following document titles list the manuals that describe the hardware for your 3280 or Micro3200 system. These manuals provide installation details and configuration ranges for your system.

- *Preinstallation Requirements Guide*
- *Using CDS to Power Up, Bootload, and Configure Your System*
- *CDS Command and Error Message Descriptions*
- *Micro3200 Compact Systems Installation and Configuration Manual*
- *Micro3200 Expanded System Installation and Configuration Manual*
- *Micro3200 MPS Installation and Configuration Manual*
- *3280MPS Installation and Configuration Manual*
- *3280E MPS Installation and Configuration Manual*
- *Installing and Configuring Distributed S–Bus Systems*
- *Shared Memory System Installation and Configuration*

- *Multiperipheral Controller (Multi–Layer MPC) Installation and Programming Manual*
- *Selector Channel (SELCH) (35–732M02) Maintenance Manual*
- *High Performance Selector Channel (HP SELCH) (35–AAJ) Installation and Programming Manual*



Index

C³Ada 15
Cache 32
CMMs 41
COBOL 16
Compatibility 5
Composite memory module. *See* CMMs.
Custom engineering. *See* System products division.
Customer service 84

Data communications 3
Data protection 6
DBL 45
Development environment 3
Distributed bus link. *See* DBL.
Distributed system bus architectures
  shared and shadowed memory 47

Documentation
  for C 92
  for C³Ada 91
  for CAL 92
  for FORTRAN VII 90
  for OS/32 87
  for other languages 92
  for reliance builder 18
  for reliance PLUS 93
  for system hardware 95

E-bus 37
ECL memory module. *See* EMMs.
EMMs 41

FORTRAN VII
  F7D compiler 14
  F7O optimizing compiler 14
  F7Z optimizing compiler 14

Hardware system
  control/diagnostic system 48
  distributed bus architecture 45
  documentation 95
  Input/output system 43
  power subsystem 39
  processing 29
  system buses 37
  memory 41

Input/output (I/O) system
  configurations 44
  DMA bus 43
  mass storage 44
  MUX bus 43

Instruction set 36

Languages 12
  C³Ada 15
  COBOL 16
  documentation 89
  FORTRAN VII 14
  other languages 17
  system support RTLs 13

Memory system
  expanded configurations 46
  memory modules 41
  shared and shadowed memory 47
Micro3200 compact systems
  peripherals 58
  system organization 54
Micro3200 expanded system
  physical profile 62
  system organization 61
Micro3200 MPS multiprocessor
  physical profile 68
  system organization 67

OLTP 3
  database integrity 23
  database security 23
  multiprocessing applications 23
  software 22
On-line transaction processing. *See* OLTP.
OS/32 8
  documentation 87
  program development 9
  task scheduling 10

Power subsystem 39
fail soft considerations 40
Processing system
3280 29
APU 29
CPU 29
MicroFive 29
MicroThree 29

Real–time processing 2
Register allocation 34
Reliance office
LEX 25
NEM/32 25
Reliance PLUS 22
availability 5
documentation 93
modularity 4
Reliance STAR 23

S–bus 37
Services
customer service 84
system products division 85
training 85
System buses
E–bus 37
S–bus 37
System products division 85
System software 7
compatibility 5
OS/32 8
programming languages 12
reliance PLUS 22
System support run–time libraries (RTLs) 13

Task scheduling
consolidation model 20
multitask model 18
Training Center 85

3280E MPS multiprocessor
physical profile 79
system organization 77
3280MPS, physical profile 73

Document Comment Form

In reference to...

3280 and Micro3200 Product Overview — 50-045 R00

We try to make our documentation easy to use, easy to understand, and free from errors. We invite your comments and suggestions to assist us in improving our documentation to suit your needs.

Please send us comments, corrections, suggestions, etc. Use the SCR system to report software documentation or software problems.

I think this manual...

	Strongly Agree	Agree	Disagree	Strongly Disagree
is easy to read	☐	☐	☐	☐
is easily understood	☐	☐	☐	☐
is concise & to the point	☐	☐	☐	☐
covers the subject	☐	☐	☐	☐
has enough detail	☐	☐	☐	☐
is well organized	☐	☐	☐	☐
provides easy-to-locate information	☐	☐	☐	☐
is aesthetically pleasing	☐	☐	☐	☐
has clear illustrations	☐	☐	☐	☐
has enough illustrations	☐	☐	☐	☐
has meaningful examples	☐	☐	☐	☐
has a helpful index	☐	☐	☐	☐

My other comments...

Please make any additional specific comments. (Include chapter, page, table or figure number.)

_____

_____

_____

_____

About myself...

Job Function: ☐ Dev. Engineer ☐ Sys. Analyst ☐ Sys./App. Prog.
☐ Technician ☐ Administrator ☐ Casual user
☐ Service Eng. ☐ Operator ☐ Other _____

What hardware system are you using? _____

_____

What revision level of system software are you using? _____

_____

Name/Title: _____

Company/Organization: _____

Address: _____

May we contact you? ☐ Yes ☐ No

Telephone: _____ Date: _____

FOLD

FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

Concurrent Computer Corporation
2 Crescent Place
Oceanport, NJ 07757

ATTN:
TECHNICAL SYSTEMS PUBLICATIONS DEPT.

FOLD

FOLD

STAPLE

STAPLE

9410