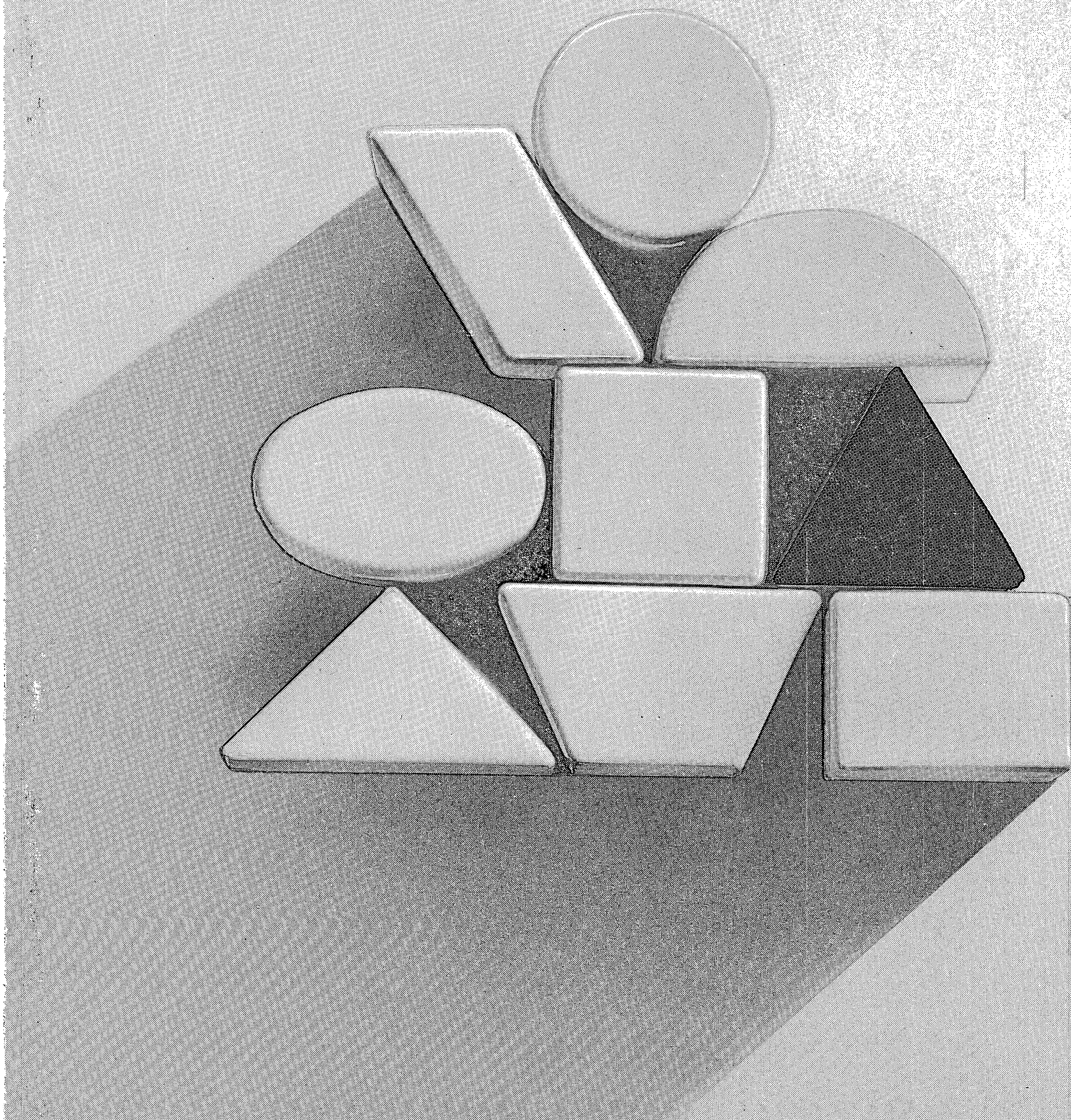
An Introduction to Microproducts and the Micron Operating System



Data General

An Introduction to Microproducts and the Micron Operating System

Data General

Data General Corporation, Westboro, Massachusetts 01581

Notice

Product Notice

References in this manual to 1.2 Mbyte diskette drives are made for informational purposes only. At the time of printing, such products were under development but not available or publicly announced. These references do not constitute an announcement of availability; DGC makes no representation that such products will at anytime be made available for purchase by its customers.

FIRST EDITION

(First Printing June, 1979)

Ordering No. 069-400000

Rev. 00

Contents

Preface

The purpose of *An Introduction to DGC Microproducts* is to give you an understanding of not only what DGC Microproducts are, but how you might use them. Microproducts consist of a number of hardware modules, and the Micron operating system and its associated utilities. They allow many different combinations, and can be used for a variety of applications.

Related Manuals

The list that follows gives a brief description of each of the other manuals which describe Microproducts and the Micron system.

- *An Introduction to Microproducts and Micron* describes the hardware and software in general terms, to give an overview of your MP/Computer and its capabilities.
- *Microproducts Hardware Systems Reference* gives a detailed functional description of the Microproducts line of Microcomputers and related peripherals, board by board.
- *Learning to Use the Micron Operating System* should be read by anyone who has never used an MP/Computer. It introduces the Micron file system and the Command Line Interpreter. A console session gives you step by step hands-on experience with your new MP/Computer.
- *Assembly Language Programmer's Reference* is the main source of information for the assembly language programmer. It describes the instruction sets of MP/Computers in detail, and gives details of the Macroassembler and operating system.
- *Micron Utilities Reference* describes the utility programs available with the Micron system.
- *MP/Fortran IV Programmers' Reference* covers all of the features of this powerful high level language.
- *MP/Pascal Programmers' Reference* describes Data General's extended version of PASCAL.

Organization of this Manual

We have organized *An Introduction to DGC Microproducts* so that general information is provided in the text, while details are given in tables and diagrams. The book has three parts: Part 1 covers the hardware; Part 2 describes the Micron operating system; and Part 3 provides examples of typical applications.

A more detailed breakdown of the book is as follows.

Part 1 Microproducts

Chapter 1 What are Microcomputers?
Provides some historical background to give you an idea of microprocessors and their usefulness. Briefly introduces Data General's microcomputers.

Chapter 2 Basic Microproducts Systems
Describes the hardware modules which constitute a basic Microproducts system.

Chapter 3 Expanding a Basic Microproducts System
Describes the peripheral devices and I/O interfaces which enable you to expand a basic Microproducts system.

Part 2 The Micron Operating System

Chapter 1 What is an Operating System?
Provides general information about operating systems and summarizes the features of the Micron operating system.

Chapter 2 Facilities of the Micron System
Describes file management, input/output, multi-tasking, program management, memory management, and library routines.

Chapter 3 Micron Utilities
Provides a description of each of the utilities available with the Micron operating system and gives you an idea of what you would use them for.

Chapter 4 High-level Languages
Describes the high-level languages available with the Micron operating system and suggests possible applications.

Part 3 Typical Applications

Provides examples of typical applications using DGC Microproducts.

Part 1
Microproducts

Chapter 1
What are Microcomputers?

We've come a long way from the early vacuum tube computers which filled whole rooms. Today, all the processing functions of a computer can fit into a package not much bigger than a dime.

This reduction in size has been made possible by advances in integrated circuit technology. Integrated circuits were first used in computers in the 1960's, and resulted in a drastic decrease in the size and cost of computers. Since then the trend in integrated circuitry has been to put together more and more components in a single chip. MSI (medium scale integration) chips have given way to LSI (large scale integration) chips. Some microcomputers use a combination of LSI chips to form the central processing unit (CPU); others use just a single chip. In March 1976, Data General introduced the first 16-bit single chip microcomputer to be designed and manufactured by a minicomputer company.

The reduction in size of computers, however, has not been accompanied by a reduction in efficiency. Although small computers do not have the memory or word size of larger machines, they are fast, efficient, and ideally suited for many general purpose applications, as well as specialized tasks. Large computers will continue to be used for complex operations, but microcomputers can perform many of the functions of the larger systems at a fraction of the cost.

Data General's Microcomputers

Data General's MP/100 and MP/200 microcomputers provide the basis for powerful and flexible systems which can be used for many different kinds of applications. The MP/100 and MP/200 are compatible, with the result that peripherals, I/O interfaces, and software are interchangeable.

The MP/100 is an inexpensive, but powerful microcomputer. The MP/100 processor consists of a single chip: the mN602 MOS microprocessor. It has a 16-bit architecture and a multi-functional instruction set. It supports up to 128Kbytes of memory. The memory can be RAM (read/write memory) and/or PROM (programmable read only memory) or EPROM (eraseable PROM).

The MP/200 is a fast and powerful microcomputer. The processor is implemented with MSI (medium scale integration) and LSI (large scale integration) components. The MP/200 has a compatible 16-bit architecture and an instruction set that is a superset of the MP/100. It supports up to 64Kbytes of RAM.

Both the MP/100 and MP/200 are upwardly compatible with NOVA and ECLIPSE line computers, so that programs written for them can be moved to a larger machine (e.g., a NOVA or an ECLIPSE) with a minimum of reprogramming.

In the next two chapters we will describe MP/computers and Microproducts devices and interfaces to give you an idea of how they can be combined, and what their usefulness is.

Chapter 2
Basic Microproducts Systems

Introduction

DGC Microproducts include a number of hardware modules which can be combined in a variety of ways to provide the type of system which best fits your needs. Microproducts systems are versatile, and can be used for many different kinds of applications, including scientific, business, process control, instrument control, data communications, and industrial automation.

A basic Microproducts system consists of:

- MP/100 or MP/200 computer
- RAM and/or ROM memory
- Power supply and chassis
- DASHER display terminal or printer
- Disk or diskette subsystem

The power supply, computer, and memory are modules which fit into the chassis. The chassis and disk may be mounted in a standard 19-inch rack or a 19-inch cabinet suitable for an office environment.

The basic Microproducts system can be expanded, as we will show you in Chapter 3, with communications devices, sensor I/O devices, a line printer, a paper tape reader, a programmable real-time clock, and a remote restart interface. All these options make the Microproducts line not only very versatile, but also very powerful. You have a great deal of flexibility in putting together the type of system best suited to your applications.

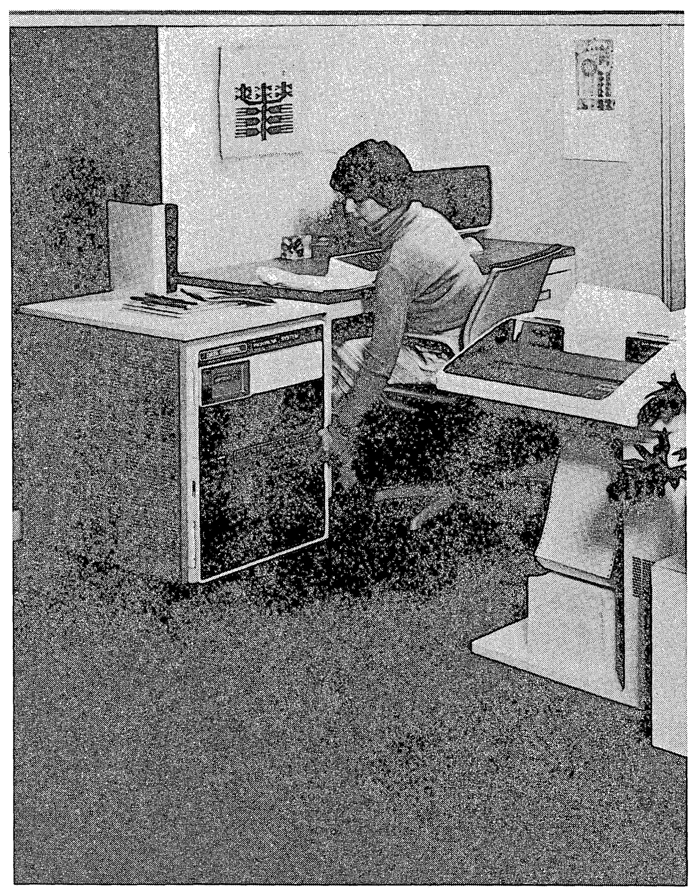


Figure 2.1 A basic MP/Computer system

Levels of Integration

The MP/computer hardware is available at four levels of integration:

- Packaged System
- Box
- Board
- Chip

The packaged system with a cabinet requires no knowledge of hardware packaging or interfacing on your part. It is ideal for development systems.

At the box level, all you have to do is plug in the peripherals; no further electronic interfacing is necessary. Should you wish to develop your own specialized interfaces however, the box is flexible enough to allow it.

The board level requires some hardware and mechanical design effort, but getting the final product completed is a fairly inexpensive and quick process.

The chip requires a significant investment in electronic and mechanical design, but can be cost-effective in large scale production or in small scale specialized production. It is most suited for applications where the computer is invisible to the end-user and is simply the least expensive way to provide the services the user requires.

MP/100 Systems

The MP/100 computer provides the basis for a powerful microcomputer system. Its high functionality and low cost make it suitable for a wide range of applications, such as process control, instrumentation and control, data acquisition, and small data systems.

The MP/100 computer is based on the mN602 MOS microprocessor. (MOS means that it is constructed using metal oxide semi-conductor technology.) This one chip comprises the central processing unit (CPU) of the MP/100. It includes:

- A full 16-bit architecture and multi-functional instruction set
- A real-time clock
- Multiply/divide logic
- Hardware stack
- Standard and high speed data channels

The mN602 chip, however, is like a car engine: essential to operating the car but not self-sufficient. Before you can run any applications you must combine the mN602 with memory and peripheral interfaces. You must also provide a power supply, attach peripherals, and perhaps load an operating system: then can you start executing an application program.

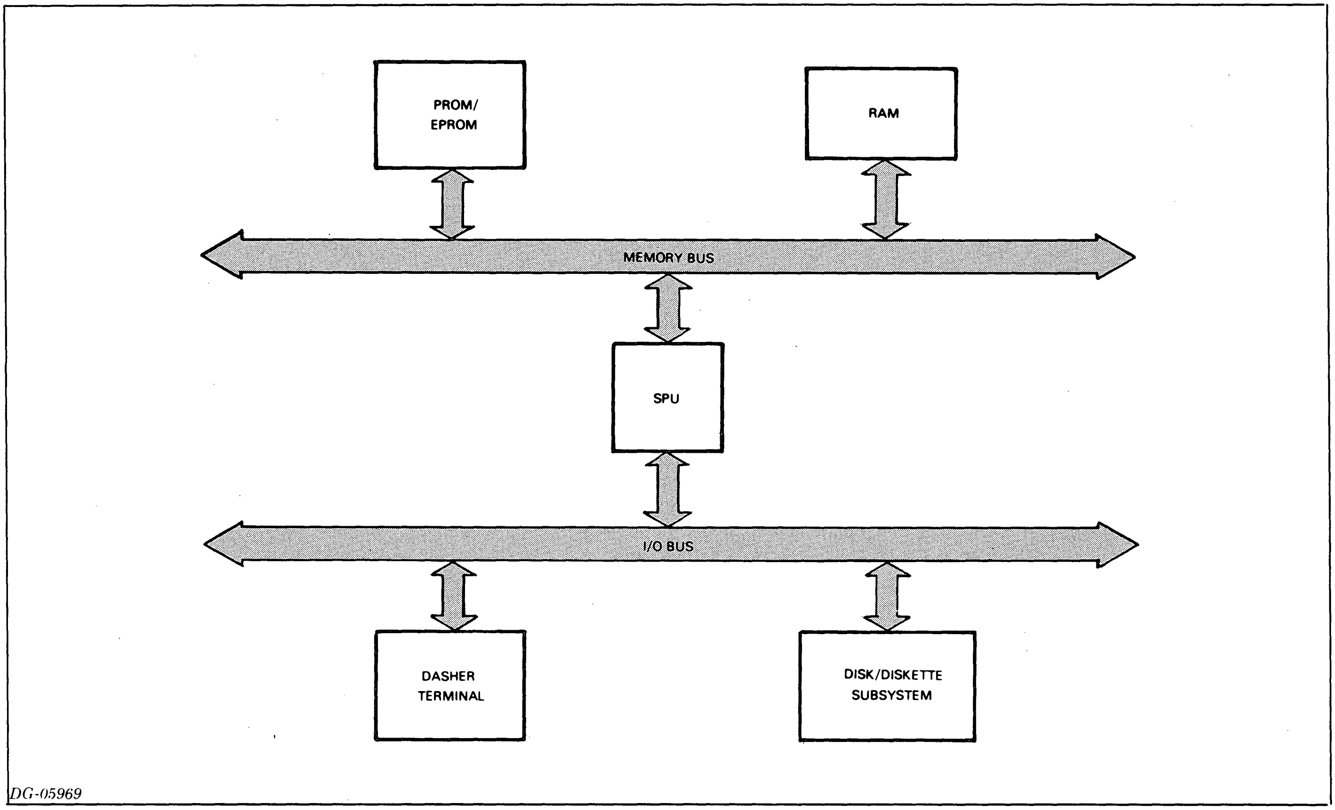


Figure 2.2 A basic MP/100 system

Thus, a basic MP/100 system actually consists of the following building modules:

- System Processing Unit (SPU)
- Dynamic Random Access Memory (RAM) in modules of 8/16Kb or 32/64Kb
- 16Kb Programmable Read Only Memory (PROM)
- RAM/EPROM (eraseable programmable read-only memory) in modules up to 32Kb RAM and 32Kb EPROM

Each module resides on a single 7 inch by 9 inch printed circuit board. In addition to these boards, a basic MP/100 system includes an 8-slot chassis, a power supply, a mounting box with fans and front panel, and an I/O bus terminator. The boards can also be used with OEM packages (the SPU is available as a separate unit, as is the mN602 chip itself).

System Processing Unit (SPU)

The System Processing Unit is a single board which contains the Central Processing Unit, the Asynchronous Interface and the Soft Control Panel.

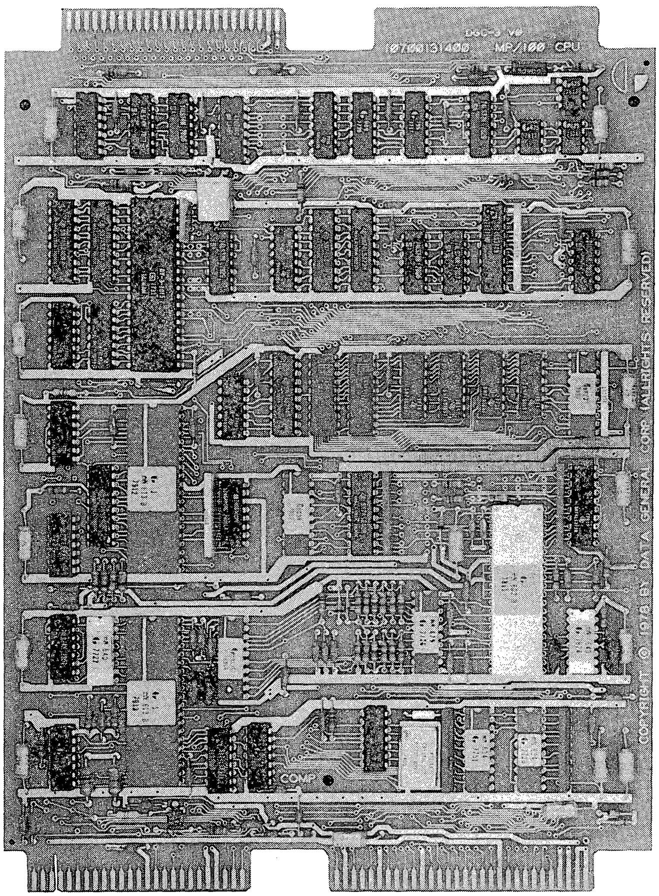


Figure 2.3 MP/100 SPU board

Central Processing Unit (CPU)

The mN602 chip supports:
Full NOVA 16-bit architecture and multi-functional instruction set.
Multiply/divide function
Hardware stack and frame pointer
16-level programmed priority interrupt
Multiple addressing modes
Integral real-time clock
Standard and high speed data channel facilities
Dynamic RAM refresh
Separate memory and I/O buses
Up to 64 Kbytes of RAM/PROM/EPROM memory. Up to 128 Kbytes of memory are available for special purpose applications which require additional memory.

Table 2.1 MP/100 CPU features

As described above, the CPU consists of the mN602 MOS microprocessor chip and is available as a separate unit. Table 2.1 describes the features of the CPU and Table 2.2 shows some sample instruction times.

Instruction	Time in μ
ADD (Register to Register Add)	2.4
SUB (Subtract)	2.4
DIA (I/O Load)	7.2
LDA (Load Register)	2.9

Table 2.2 MP/100 sample instruction execution times

NOTE: *We show actual instruction times rather than "cycle" times since some instructions have multiple cycles.*

Asynchronous Communications Interface

The asynchronous communications interface is a programmed I/O controller which allows communications between the CPU and a terminal over an asynchronous communications line. Modem control with auto answer is available, and a subset of EIA modem control functions is supported, providing an interface to telephone lines.

Line type	Asynchronous
Number of lines	1
Transmission type	Full-duplex
Jumper selectable features:	
Line speed	50 to 19,200 baud
Line interface	EIA RS232-C or 20mA current loop
Character length	5-8 data bits
Parity	Odd, even, or none
Modem control signals	Clear to send
	Request to send
	Data terminal ready
	Carrier detect
	Data set ready
	Ring indicator
Modem control	Bell 103, 202 or equivalent
Voltages required	+5V, -5V, +12V

Table 2.3 MP/100 asynchronous communications interface features

Soft Control Panel

The Soft Control Panel (SCP) is a program which allows you to interact with the MP/100 via a terminal connected to the asynchronous interface. By using the terminal, you can inspect and modify the state of the system. Basically, you can perform the kinds of operations which are usually performed by setting the front panel switches on a "traditional" computer. So, for example, the SCP lets you initiate program load sequences and also helps you debug programs by allowing you to stop, start, and continue program execution, and examine and/or alter registers and memory locations.

Memory

Random Access Memory (RAM)

Random access memory (RAM) is a local storage medium whose contents can be read and modified, one word at a time. The MP/100 uses dynamic MOS RAM which must be refreshed at intervals so that its contents are maintained. This operation is performed by the mN602 chip.

There are two MP/100 MOS RAM boards: an 8/16 Kbyte board and a 32/64 Kbyte board. The MP/100 can directly address up to 128 Kbytes of memory, all of which may be RAM.

Memory type	Dynamic MOS N-channel RAM
Board capacity	8/16Kb: 8,192 or 16,384 bytes 32/64Kb: 32,768 or 65,536 bytes
Cycle time	Read: 960 ns Write: 960 ns Refresh: 960 ns
Board dimensions	19 x 24.9 cm (7.5 x 9.9 inches)

Table 2.4 MP/100 RAM board specifications

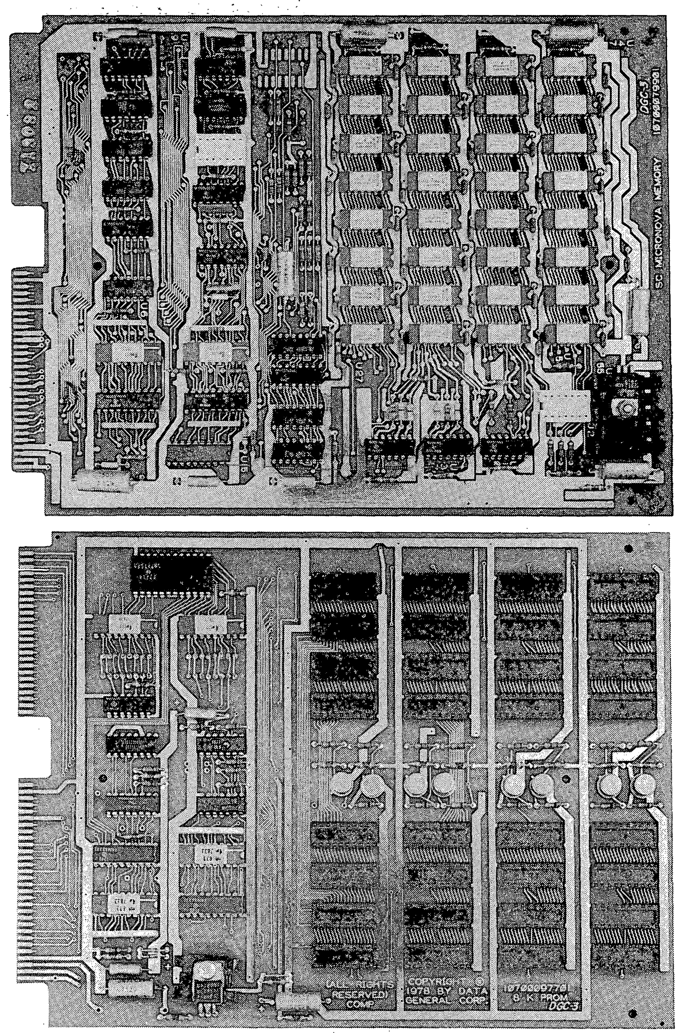


Figure 2.4 MP/100 memory boards

Programmable Read Only Memory (PROM)

The MP/100 has available a 16 Kbyte Programmable Read Only Memory (PROM) module and a RAM/EPROM module with up to 32Kbytes of RAM and 32Kbytes of PROM. Either of these modules allow you to tailor an MP/100 system to fit your needs.

The PROM is for applications that require programs or data to be fixed within the system. Systems configured with PROM memory often do not have disks or other "general purpose" computer interfaces. Instead, they tend to be inside "black boxes" so they are invisible to the end user. The applications for this kind of system might include medical analysis, laboratory control, security control, and power monitoring.

MP/200 Systems

The MP/200 computer provides the basis for a powerful, high-speed microcomputer system. It is suitable for a variety of applications, such as business, scientific, and data communications. The MP/200 is compatible with the MP/100. Its instruction set is a superset of the MP/100. This compatibility means that all the peripherals, I/O interfaces, and software are interchangeable.

The MP/200 computer consists of the following high-speed modules:

- System Processing Unit (SPU)
- Multi-Function Controller
- Dynamic Random Access Memory (RAM) in modules of 16 Kbytes or 32/64 Kbytes

Each module resides on a single 7 inch by 9 inch printed circuit board. In addition to these boards, a basic MP/200 system includes an 8-slot chassis, a power supply, a mounting box with fans and front panel, and an I/O bus terminator. (The same chassis is used by the MP/100.) The boards can also be used with OEM packages (the SPU is available as a separate unit).

System Processing Unit (SPU)

The SPU is a micro-programmed processor. It has exceptional throughput and response characteristics (240 nanosecond CPU cycle time) and supports both standard and high speed data channel facilities. Transfer rates via the standard data channel are typically 300 Kbytes/second and via the high speed data channel are 3.7 Mbytes/second.

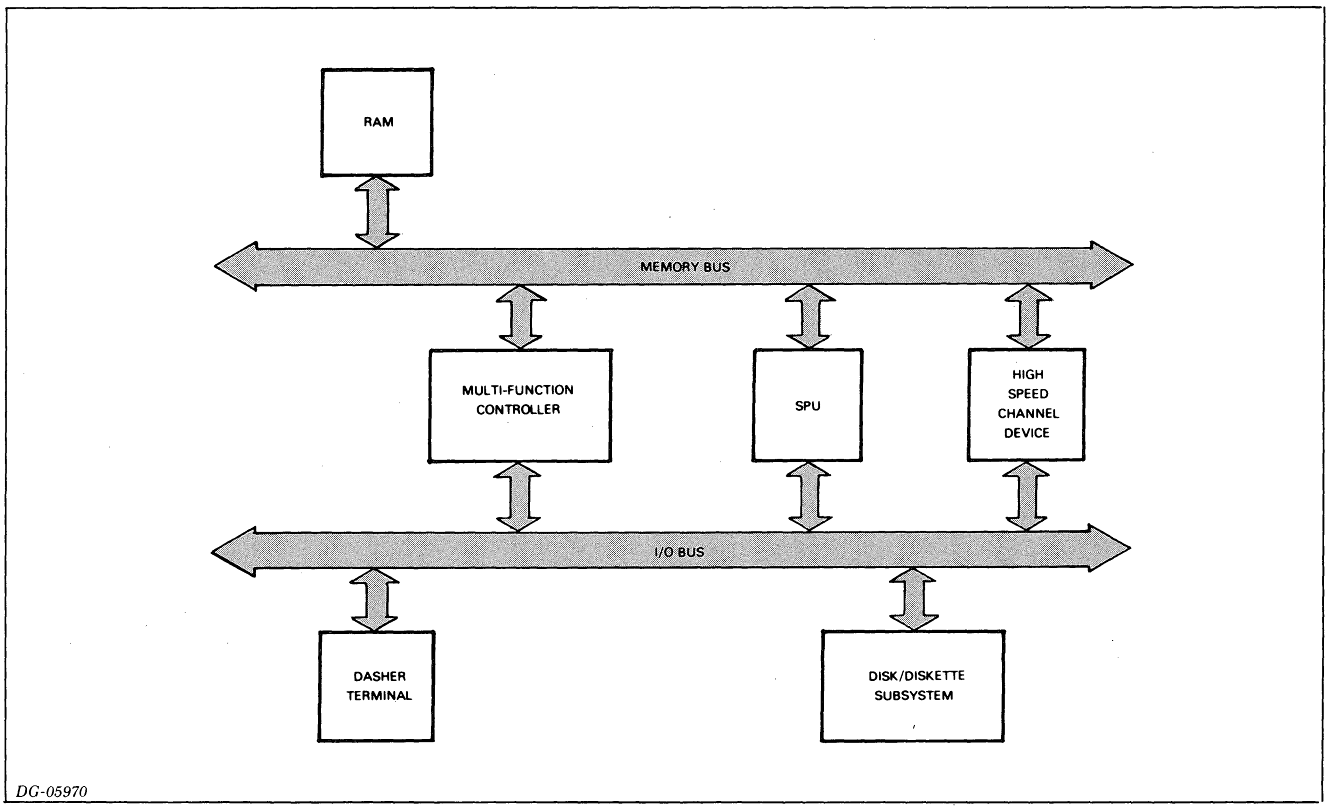


Figure 2.5 Basic MP/200 system

MP/200 SPU supports:

- Full NOVA 16-bit architecture and extended multi-functional instruction set. Additional instructions include signed multiply/divide and load/store byte.
- Multiple addressing modes
- Hardware stack and frame pointers
- 16-level programmed priority interrupts
- Separate memory and I/O buses
- Standard and high speed data channel facilities
- Up to 64 Kbytes RAM memory

Table 2.5 MP/200 SPU features

Instruction	Time in μ
ADD (Register to Register Add)	0.84
DIV (Unsigned Divide)	6.6
MUL (Unsigned Multiply)	4.92
LDA (Load Register)	1.44
LDB (Load Byte)	1.92
STB (Store Byte)	1.44

Table 2.6 MP/200 sample instruction execution times

NOTE: *We show actual instruction times rather than "cycle" times since some instructions have multiple cycles.*

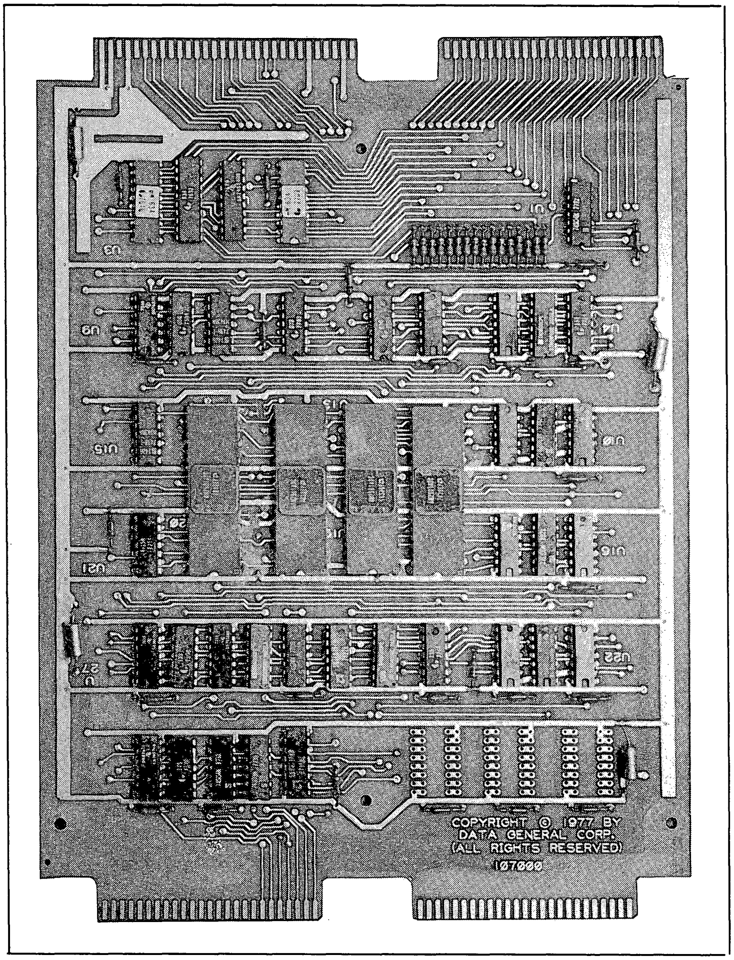


Figure 2.6 MP/200 SPU board

Multi-Function Controller

The multi-function controller is a companion board for the SPU board and consists of the following:

- Asynchronous Communications Interface
- Soft Control Panel
- Real-Time Clock Interface
- Control Panel Interface

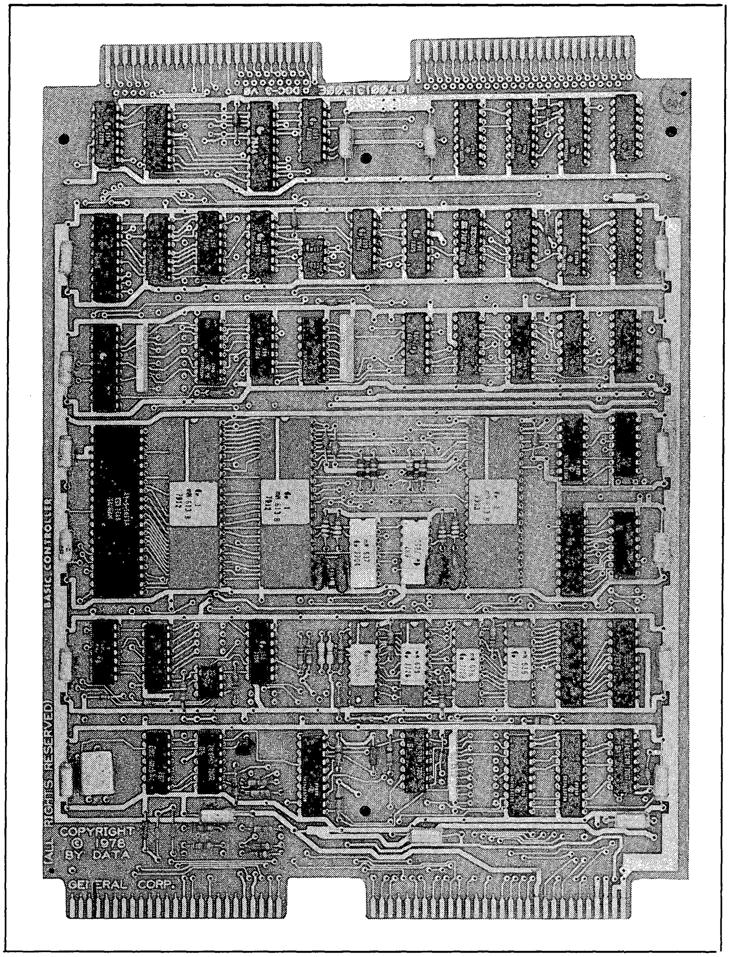


Figure 2.7 MP/200 multi-function controller

Asynchronous Communications Interface

The asynchronous communications interface is a programmed I/O controller which allows communications between the CPU and a terminal over an asynchronous communications line. Modem control with auto answer is available, and a subset of EIA modem control functions is supported, providing an interface to telephone lines. The interface is identical in function and control to the asynchronous interface on the MP/100 and the Model 4207S asynchronous controller (see Chapter 3).

Line type	Asynchronous
Number of lines	1
Transmission type	Full-duplex
Jumper selectable features:	
Line speed	50 TO 19,200 baud
Line interface	EIA RS232-C or 20mA current loop
Character length	5-8 data bits
Parity	Odd, even, or none
Stop bits	1 or 2
Modem control signals	Clear to send
	Request to send
	Data terminal ready
	Carrier detect
	Data set ready
	Ring indicator
Modem control	Bell 103, 202 or equivalent
Voltages required	+5v, -5v, +12v

Table 2.7 MP/200 asynchronous communications interface features

Soft Control Panel

The Soft Control Panel (SCP) is a program which allows you to interact with the MP/200 via a terminal connected to the asynchronous interface. By using the terminal, you can inspect and modify the state of the system. Basically, you can perform the kinds of operations which are usually performed by setting the front panel switches on a "traditional" computer. So, for example, the SCP lets you initiate program load sequences and also helps you debug programs by allowing you to stop, start, and continue program execution, and examine and/or alter registers and memory locations. The SCP resides in 512 words of ROM (read only memory).

Real-Time Clock Interface

The Real-Time Clock Interface can supply program interrupt requests at one of four program-selectable frequencies: 10 Hz, 100 Hz, 1,000 Hz, or line frequency. The interface is functionally and programmably compatible with the Model 4220S real-time clock (see "Expanding a Basic Microproducts System").

Control Panel Interface
The Control Panel Interface monitors the power status of the system and supplies power fail/auto restart information to the CPU. It also monitors the MP/200 panel switches.

Memory

Random Access Memory (RAM) is a local storage medium whose contents can be read and modified, one word at a time. The MP/200 has dynamic MOS RAM, which means it is constructed using metal oxide semi-conductor (MOS) technology. Dynamic MOS RAM must be refreshed at intervals so that its contents are maintained. This operation is performed by the CPU.

There are two MP/200 MOS RAM boards: an 16 Kbyte board and a 32/64 Kbyte board.

Memory type	Dynamic MOS N-channel RAM
Board capacity	16K: 16,384 16-bit words 32K/64k: 32,768 or 65,536 16-bit words
Cycle time	Read: 600ns Write: 600ns
Board dimensions	19 x 24.9 cm (7.5 x 9.9 inches)

Table 2.8 MP/200 RAM board specifications

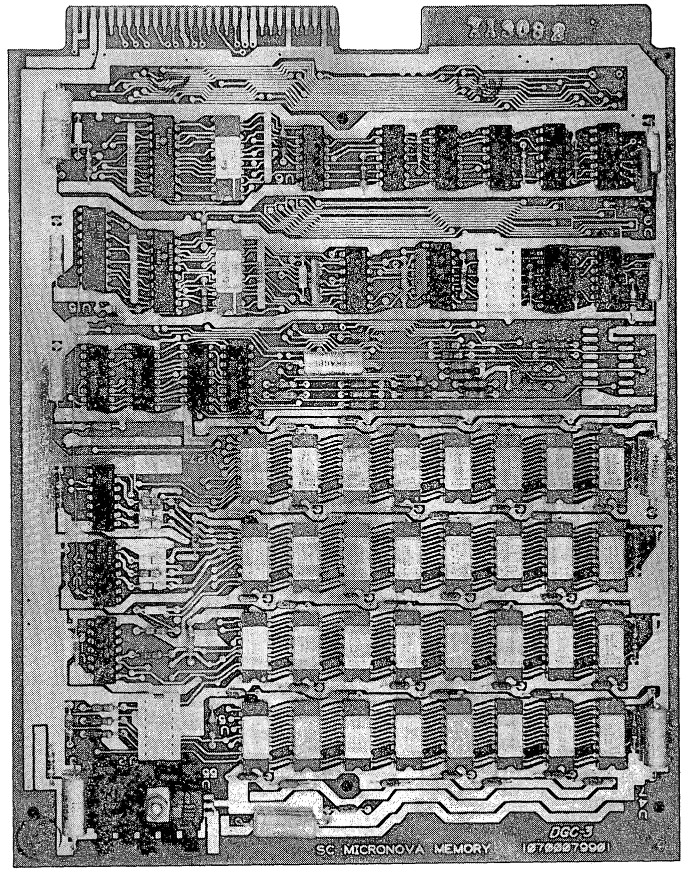


Figure 2.8 MP/200 RAM board

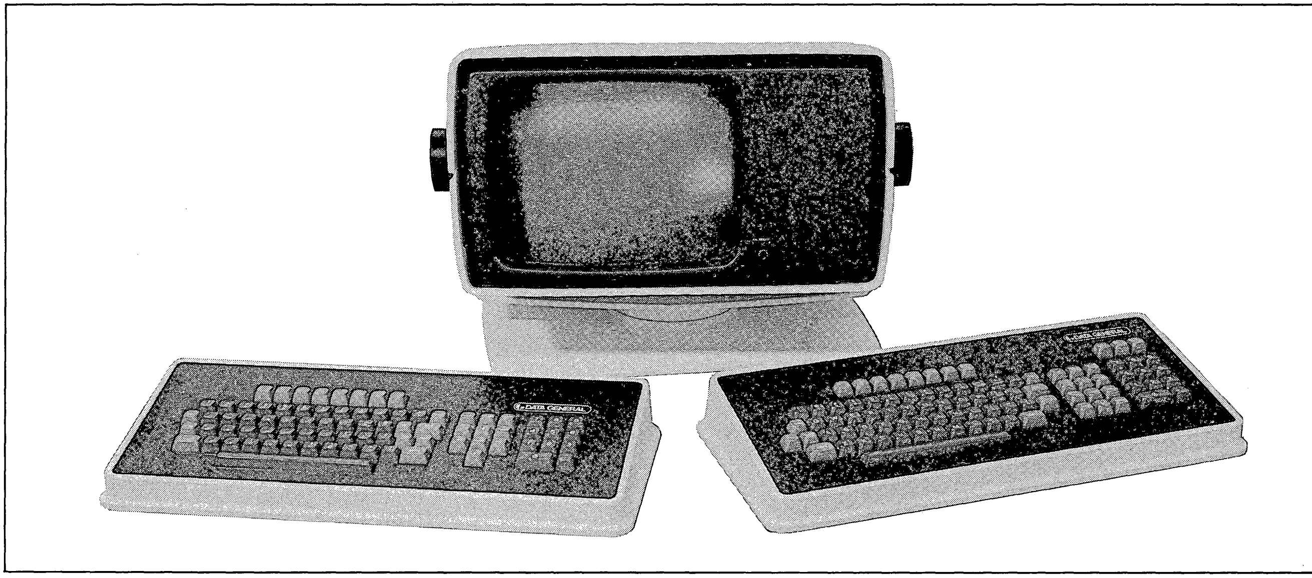


Figure 2.9 DASHER display terminals

Figure 2.10 DASHER lineprinters

DASHER Terminals

A basic Microproducts system includes either a DASHER display terminal or DASHER printer. Both allow communication with an MP/computer via an asynchronous communications interface. Thus a terminal provides you with an interface to your application programs. Also, as you may recall, a terminal allows you to use the Soft Control Panel on an MP/computer to check the state of the system and to debug programs.

DASHER Display Terminal

The 6052 and 6053 DASHER display terminals are CRT input/output devices. The 6052 is the basic terminal, using the 64-character upper case ASCII set. It has 8 user-definable keys and an 11-key numerical keypad, in addition to the main keypad. The 6053 is the enhanced terminal, using the 96-character upper/lower case ASCII set. The 6053 has 11 user-definable keys and an 11-key numerical keypad, in addition to the main keypad.

The 6052/3 terminals can be used in any interactive alphanumeric applications, but they are especially suited to applications such as editing or order entry where the ability to quickly display a large amount of information is useful.

DASHER TP1 and TP2 Printers

The DASHER TP1 and TP2 printers are hard-copy input/output devices. The printers are available with one of 3 jumper-selectable ASCII character sets: 64-character upper case; 96-character upper/lower case; and the 124-character extended set. They print at rates of 30 cps, 60 cps, or 180 cps and accomodate 1 to 6 part forms ranging from 4 to 15 inches in width. Other features include a view mode and a numeric keypad.

The 6040 printers can also be used in any interactive alphanumeric applications, especially where printed output is required.

Mass Storage

Disk drives are mass storage devices which enable you to transfer data quickly. The recording medium consists of magnetic material coated on a platter called a disk. Different types of disks are available: a disk cartridge is a single platter encased in a plastic cartridge; a diskette is a small flexible disk enclosed in a protective packet.

Some drives use diskettes or disk cartridges which you can change; some drives have fixed disks; still others combine fixed disks with removable disk cartridges. Cartridge disk and fixed disk drives record data on both sides of the disk, but some diskette drives record data on the top surface only.

6101/6102 and 6096 Disk Subsystems

The Model 6101 subsystem consists of:

- 12.5 Mbyte Fixed Disk Drive
- 1.2 Mbyte Diskette Drive*
- Controller

The diskette drive mounts on top of the fixed disk drive and is powered by it.

The Model 6102 subsystem consists of:

- 12.5 Mbyte Fixed Disk Drive
- Controller

The Model 6096 subsystem consists of:

- Up to 4 1.2 Mbyte Diskette Drives*
- Controller

Fixed Disk Drive

The fixed disk drive contains a single disk which you cannot remove. The disk has a storage capacity of 12,582,912 bytes. The drive provides two heads per surface to improve access time. It uses Winchester technology to assure maximum reliability, high storage density, and fast data transfers. The modified frequency modulation (MFM) encoding method maximizes the drive's storage capacity. The drive is self-contained in a 19-inch rack mountable chassis, and includes a power supply.

The large storage capacity and fast access time of the fixed disk drive makes it ideal for program development since it cuts the time and cost of development substantially. Its high speed makes it perfect for any applications where fast access is an important factor. The fixed disk drive is also very suitable for a data base management system.

**Please refer to the Product Notice on the reverse side of the title page of this manual.*

1.2 Mbyte Diskette Drive*

A 1.2 Mbyte diskette drive holds one removable diskette. The drive uses technology which allows you to record data on both sides of the diskette. This ability, together with the MFM recording technique, allows you to store 4 times the amount of data you can store on a standard diskette. The 1.2 Mbyte diskette has a capacity of 1,261,568 bytes.

The 1.2 Mbyte diskette drive is normally housed in a special 19-inch rack mountable chassis that also contains a power supply. Each chassis can hold one or two drives.

The 1.2 Mbyte diskette drives have a smaller storage capacity and slower access times than the fixed disk drives, but they are less expensive. Diskettes are very useful for making back-ups of important data files stored on fixed disks, and for diagnostic loading and software distribution. You can also use diskettes in small-scale program development systems.

**Please refer to the Product Notice on the reverse side of the title page of this manual.*

6095 Cartridge Disk Subsystem

The 10 Mbyte cartridge disk subsystem consists of a controller, which directs all the activities of the subsystem, and a moving head disk drive with two disk platters. One of the platters is factory-installed, so you cannot remove it. The other is a removable disk cartridge. Each platter has a storage capacity of 5 Mbytes.

Due its large storage capacity (10 Mbytes) and fast access time, the cartridge disk subsystem is suitable for program development, a data base management system, or business data processing.

6038/9 Diskette Subsystem

The 0.3 Mbyte diskette subsystem consists of a controller, which directs all the activities of the subsystem, and one or two moving head diskette drives. Each drive can contain one diskette, and records data on the top surface only. The diskette has a storage capacity of 315,392 bytes.

The 0.3 Mbyte diskette subsystems have a small storage capacity and relatively slow access time, but are inexpensive. They are primarily useful for transferring data from one system to another or for making back-ups of important data files.

	12.5 Mb Disk Drive	1.2 Mb Diskette Drive*	10 Mb Disk Drive	0.3 Mb Diskette Drive
Capacity/ unit	12.5 Mb	1.2 Mb	10 Mb	0.3 Mb
Surfaces/ unit	2	2	4	1
Heads/ surface	2	1	1	1
Tracks/ head	192	77	408	77
Sectors/ track	32	16	12	8
Bytes/ sector	512	512	512	512
Effective data transfer rate	910,500 bytes/sec	62,500 bytes/sec	625,000 bytes/sec	31,250 bytes/sec
Data channel latency	No max.	50 usec max.	No max.	63.9 msec
Track access time	1 track: 15 msec 1/3 stroke: 60 msec	3 msec/track	10 msec/ track	10 msec/ track
	Full stroke: 120 msec	Full stroke: 70 msec		
Recalibrate time	540 msec max.	251 msec max.	-	-
Head load time	-	52 msec.	-	-
Rotational speed	49.4 rev/sec	6 rev/se	80 rev/sec	6 rev/sec

Table 2.9 Mass storage device characteristics

NOTE: *The average seek time and average rotational latency of a disk drive may be critical for applications which are heavily disk-bound. As you can see from Table 2.9, random disk access time is equal to many instruction cycles. When we discuss multi-tasking (see Part 2), we will point out how this time can be used. You should remember, however, that disk access speed is often the key to application performance.*

Power Supply

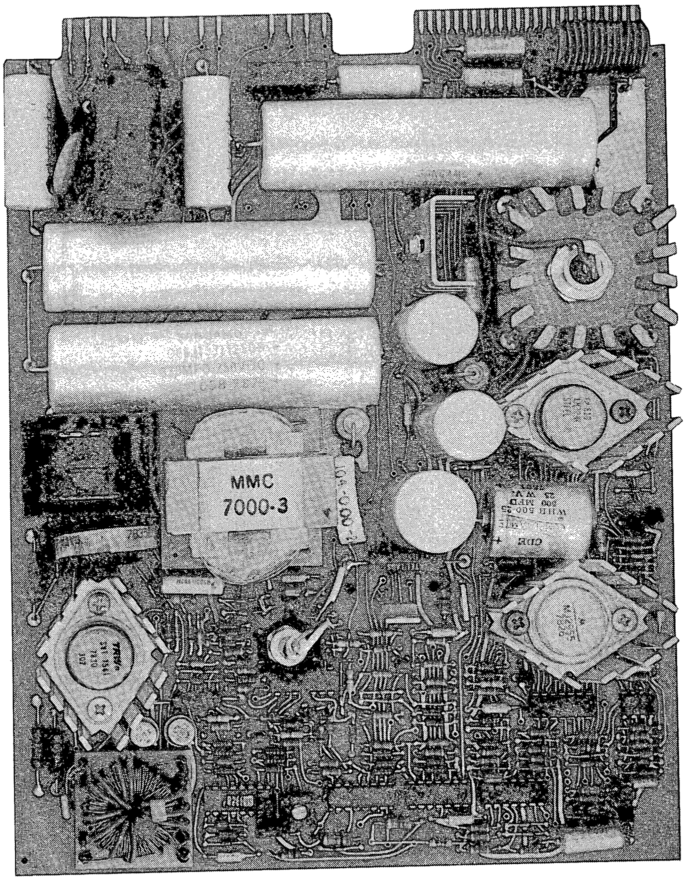


Figure 2.11 MP/Computer power supply

The power supply is on a single printed circuit board which increases its reliability and maintainability. It supplies all the voltages your MP/system requires:

- +5V @ 16A
- -5V @ 2A
- +12V @ 1.75A
- -12V @ 0.75A

Packaging

A basic Microproducts system includes packaging building blocks consisting of chassis, power supply, and cabinets. An 8-slot chassis is available. It can hold the SPU board, the memory board(s), and interface boards such as the MP/200 multi-function controller. It also holds the power supply, and has cooling fans. All I/O cabling is easily accessible on the unobstructed backplane. A front panel is optional. (For board-level users there is a low-cost 4-slot chassis available.)

Your MP/system, together with a disk subsystem, can be mounted in an inexpensive half-bay (less than 30 inches high) cabinet. The cabinet is portable, and has a formica top.

Chapter 3
Expanding a Basic Microproducts System

So far we have described the building blocks which might constitute a basic Microproducts system and have suggested possible applications. Several other building blocks exist, however, thereby increasing the versatility of Microproducts systems. For example, a range of communications devices allows you to build communications systems and a range of sensor I/O devices permits a variety of process control applications.

These additional building blocks include the following:

- Asynchronous Communications Controller
- Communications Subsystem
- Line Printer Controller
- Paper Tape Reader Controller
- Real-Time Clock
- Remote Restart Interface and Down-Line Load Option
- Sensor I/O Devices

NOTE: *In addition to these standard interfaces, custom designed interfaces for special applications are available from Data General's Special Systems Group.*

Asynchronous Communications Controller

The Model 4207S Asynchronous Communications Controller provides the interface between an MP/computer and an asynchronous communications line. Modem control is available, providing an interface to telephone lines.

The 4207S is functionally identical to the asynchronous communications interface on the MP/100 and MP/200 computers. If, in your system, you want more than the one terminal which a basic MP/computer supports, you can use a 4207S to support an additional terminal.

The 4207S can support a wide variety of terminal devices for either local or remote applications.

Line type	Asynchronous
Number of lines	1
Transmission type	Full-duplex
Jumper selectable features:	
Line speed	50 to 19,200 baud
Line interface	20mA current loop or EIA RS232-C
Character length	5 - 8 data bits
Parity	Odd, even, or none
Stop bits	1 or 2
Modem control signals	Request to send
	Data terminal ready
	Data set ready
	Ring indicator
Modem control	Bell 103, 202 or equivalent
Voltages required	+5v, -5v, +12v, -12v

Table 3.1 Model 4207S asynchronous communication controller characteristics

Communications Subsystem

The communications subsystem consists of a number of modules which enable you to design a communications system to fit your particular needs. You can combine the communications modules in a variety of ways, forming small or large systems and mixing high and low speed, asynchronous and synchronous, local and remote lines. The modularity of the communications subsystem makes it very flexible; you can easily expand your communications system as needs change. The fact that you can program certain line characteristics makes it even easier to adapt your communications subsystem to changing requirements.

The communications subsystem consists of the following modules:

- Model 4225S Controller
- Model 4227S Asynchronous Line Multiplexor (ALM)
- Model 4226S Synchronous Line Controller (SLC)
- Model 4228S Cyclical Redundancy Check (CRC) Generator

Controller

The Model 4225S controller decodes all I/O instructions between the communications subsystem and the CPU. The controller can support a mixture of up to 4 asynchronous and synchronous boards (ALMs and SLCs).

Asynchronous Line Multiplexor

The Asynchronous Line Multiplexor (ALM) can support up to 4 asynchronous lines with full modem control. The lines support communications at rates of up to 19,200 baud over serial EIA or 20mA current loop lines. You can program the following line characteristics: character length, type of parity, number of stop bits, and line speed.

The ALM is a single board which fits into an MP/computer chassis slot. You can have up to four ALMs in a communications subsystem. You could therefore support as many as 17 terminals (remember you have an asynchronous interface on your MP/computer) in a Microproducts system.

The terminals could be used in applications such as order entry/verification or information display.

Line type	Asynchronous
Number of lines	4 (up to 16 in a subsystem)
Chassis	MP/100 or MP/200
Slots required	1
Prerequisite	4225 controller
Transmission type	Half or full duplex
Line speed	50 to 19,200 baud
Line interface	EIA RS232-C or 20mA current loop
I/O latency	1 character
Programmable characteristics	5-8 bits/character 1 or 2 stop bits even/odd/no parity

Table 3.2 Model 4227S ALM characteristics

Synchronous Line Controller

The Synchronous Line Controller (SLC) can support 1 synchronous communications line with full modem control. The SLC is compatible with IBM's binary synchronous communications protocols (BSC and BISYNC) and supports transparent and non-transparent communications at speeds of up to 19,200 baud over a serial EIA line. You can program the following line characteristics: character length, type of parity, and SYN and DLE characters. Line speed is jumper selectable, or may be provided by a modem.

The SLC is a single board which fits into a slot of any MP/computer chassis. You can have up to four SLCs in a communications subsystem. The SLC might be used as the interface between a remote job entry MP/computer system and a host computer.

Line type	Synchronous
Number of lines	1 (up to 4 in a subsystem)
Chassis	MP/100 or MP/200
Slots required	1
Prerequisite	4225 controller
Transmission type	Half or full duplex
Line speed	50 to 19,200 baud
Line interface	EIA RS232-C
I/O latency	1 character
Programmable characteristics	6-8 bits/character Even/odd/no parity SYN and DLE characters

Table 3.3 Model 4226S SLC characteristics

Cyclical Redundancy Check Generator

Cyclical redundancy checking (CRC) is a method of checking blocks of data being transmitted on a synchronous communications line. The data stream is divided by a CRC polynomial with the remainder serving as the cylic redundancy check character. The receiving station compares the remainder transmitted with the data block (calculated at the transmitting end) with the one it has computed on the data block it received, and signals a data error if they do not match. You can perform this type of error checking with software instead of a CRC generator, but performance is degraded.

The CRC generator is a single board which fits into a slot of any MP/computer chassis. One CRC unit will support multiple synchronous interfaces. You can program the CRC to use one of two polynomials shown in Table 3.4.

CRC-16 ($X^{16} + X^{15} + X^2 + 1$)
CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$)

Table 3.4 CRC polynomials available

Line Printer Controller

The line printer controller is the interface between an MP/computer and a line printer. The line printer controller is programmable, and resides in a slot in the main chassis.

Line printers provide high speed alphanumerical hard-copy output. They are useful in any applications where you require printed copies, such as program listings and reports. The line printer controller interfaces with the following DGC line printers: 4034G/H and 6073/4.

Model 4034G and 4034H Line Printers

4034G/H line printers are character drum impact printers. They accomodate 1 to 6 part forms, ranging from 4 to 15 inches in width.

DASHER 6073/4 LP2 Printers

DASHER 6073/4 LP2 printers are serial dot matrix impact printers. They also accomodate 1 to 6 part forms, ranging from 4 to 15 inches in width. Other features include horizontal and vertical tabbing, plotting, underscoring, elongated characters, a switch-selectable view mode, and a self-testing diagnostic program. Compressed printing and automatic formfeeds are optional.

Model	Printing Characters	Maximum Characters/ Line	Vertical Line Spacing	Print Rate
4034G	ASCII 64	136	6 OR 8 lines/inch	300 lpm
4034H	ASCII 96	136	6 OR 8 lines/inch	240 lpm
DASHER 6073/4 LP2's	ASCII 96	132	6 OR 8 lines/inch	180 cps

Table 3.5 Line printer characteristics

Paper Tape Reader Controller

The paper tape reader controller is the interface between an MP/computer and a Model 6013 paper tape reader. The paper tape reader controller is available on the same board as the Model 4220S programmable real-time clock.

A paper tape reader is an input device which reads information stored on 8-channel paper or mylar tape. Data is recorded in the form of holes in the tape.

Programmable Real-Time Clock

The Model 4220S programmable real-time clock resides on the same board as the paper tape reader controller. The real-time clock provides a choice of precise time bases for your entire Microproducts system. Four frequencies are available: 10Hz, 100Hz, 1000Hz, or line frequency.

Notice that the real-time clock is functionally and programmably compatible with the one on the MP/200 multi-functional controller. You would be most likely, therefore, to use the 4220S real-time clock with an MP/100 for applications (such as process control) where a programmable clock is preferable to the CPU's built-in clock.

Remote Restart Interface and Down-Line Load Option

The Remote Restart Interface provides a Microproducts system with:

- Remote reset and restart
- A secondary asynchronous communications port

In addition, the down-line load option offers:

- Remote program loading
- Remote diagnostic checkout

All the above operations can be performed from a host computer which is connected to a Microproducts system by an asynchronous communications line. The communications line may be local or remote with either 20mA or EIA RS-232-C characteristics.

> **NOTE:** *The down-line load option is not supported by the MP/200.*

The Remote Restart Interface and Down-Line Load option make it possible for a Microproducts system to be used in unattended operations, such as automated factory, laboratory, and data communications applications. The master CPU can be in constant touch with the slave MP/computer system and can reload it in the event of a power failure. The Remote Restart Interface can also allow a remote system to abort the current operation, reset all interfaces, and start at a predetermined location. With the Down-Line Load option, the remote processor can down-line load a program into the local MP/Computer, as well as check out the interface connection.

Sensor I/O Interfaces

Sensor I/O interfaces allow a Microproducts system to interface directly with a wide range of data acquisition and control equipment. Using either a single interface or a combination of interfaces, you can tailor a system to your particular application. Applications might include medical analysis, radio navigation, pollution monitoring, and sample analysis.

The following sensor I/O interfaces are available:

- Digital I/O Interface (Model 4222S)
- Analog-to-Digital Interface (Model 4223S)
- Digital-to-Analog Interface (Model 4224S)

Digital I/O Interface

The Digital I/O Interface provides:

- 16 digital inputs
- 16 digital outputs
- 1 input strobe
- 2 output strobes
- 1 status output

Using the output lines, a program can supply control information and/or data to user equipment. Using information appearing on the input lines, the interface can supply the program with three types of data: external, stored, and comparison data.

The digital I/O interface would be used with any process control device with a parallel interface.

Analog-to-Digital Interface

The Analog-to-Digital Interface allows an MP/computer to read digitized data from an external analog measuring device. The interface supplies digitized data with a resolution of 12 bits, from 16 single-ended or from 8 differential analog input channels.

The Analog-to-Digital Interface has two operating modes: programmed I/O and data channel. In programmed I/O mode, the interface performs single conversion. In data channel mode, the interface can perform either single conversion or automatic conversion.

Resolution	12 bits
Maximum conversion time	25 μ
Maximum conversion rate	30 kHz
Maximum allowable data baud channel latency @ 30kHz	33.33 μ
System accuracy:	
Linearity	+-1/2 LSB
Differential linearity	+-1/2 LSB
Relative accuracy @ 25°c	+-0.025% of FSR
Jumper selectable voltage ranges	0 to 5V 0 to 10V +/-5V +/-10V

Table 3.6 A-D interface features

Digital-to-Analog Interface

The Digital-to-Analog Interface allows an MP/computer to interface with any standard storing or non-storing oscilloscope, among other devices. The interface has two analog output channels, each of which supplies voltage generated by its associated digital-to-analog converter and drives 5mA loads.

The two digital-to-analog converters operate with voltage ranges which are independently jumper-selectable. For each voltage range, each converter can supply up to 4096 discrete analog values from digital data with a resolution of 12 bits.

The Digital-to-Analog Interface has two operating modes: programmed I/O and data channel. In programmed I/O mode, the interface performs a single conversion. In data channel mode, the conversion can be single or automatic.

Resolution	12 bits
Settling time to 0.01% of final value	7 μ
Relative accuracy @ 25°c	0.025% of FSR
Jumper selectable voltage ranges	0 to 5V 0 to 10V +/-5V +/-10V

Table 3.7 D-A interface features

Part 2
The Micron Operating System

Chapter 1
What is an Operating System?

If your computer system consisted of hardware devices alone, you would find it very difficult to write programs for your applications. Not only would you have to write them in machine language, but you would have to worry about the unique characteristics of all the devices in your system. An operating system solves these problems. Basically, an operating system is a program or group of programs which acts as an interface between you and the hardware, and facilitates the creation and execution of your programs. An operating system allows you to use the hardware without having to know all the details of its operation.

The operating system manages access to processor time, memory, and input/output devices. By allowing you (the user) and your programs to communicate with it, an operating system makes it possible for your programs to be serviced. In each case, simple requests may result in complex operations on the part of the operating system. You can communicate with the operating system by means of commands and messages typed at a console. A program can communicate with the operating system by means of instructions called system calls which reference routines built into the operating system.

Operating systems usually contain a supervisor program which, as its name suggests, supervises various operations performed by the computer system. The supervisor consists of routines which are referenced with system calls. It resides in main memory and manages functions such as:

- Input/output
- File management
- Multi-tasking
- Program management

These functions will be explained in subsequent chapters.

In addition to the supervisor program, an operating system may include utility programs which perform basic services such as assembling, binding, and debugging programs; text editing; file system management. These services allow you to make full use of the resources of the computer system. The operating system may also include high-level languages for use in program development.

A variety of operating systems exist. For example, operating systems may be single-user or multi-user. The Micron operating system is single-user. In a single-user system only one person has access to the full development facilities of the operating system. Data General's Advanced Operating System (AOS) is a multi-user system for the ECLIPSE computer line. In a multi-user system, many people can use the system simultaneously, and the system must decide how the resources are to be shared.

Operating systems can also be differentiated by the number of programs or processes they allow to run at the same time. Some (such as the Micron system) allow only one program; others (such as Data General's Real-Time Disk Operating System) allow foreground/background programming which means you can run two programs at the same time; and still others (such as AOS) allow multi-programming so that many programs run with apparent simultaneity.

In addition, operating systems can be differentiated by the number of operations they allow an application to control at one time. The Micron system, AOS, and RDOS all allow multi-tasking so that multiple operations can be controlled and coordinated by a single-user program. This means that an application (even on a single-process system like the Micron system) can support many "users". See "Multi-Tasking" for a more detailed description.

The Micron Operating System

The Micron operating system is a single-process general purpose system. It can perform:

- File Management
- Input/Output
- Multi-Tasking
- Program Management
- Memory Management

Micron supports several utilities which allow operations such as program development, text editing, and file management. The operating system also supports high-level languages (MP/Pascal and MP/Fortran IV) to aid you in program development.

The Micron system is "upwards compatible" with the Advanced Operating System (AOS). This means that you can write, assemble, bind, and debug programs on an AOS system and run them on a Micron system, and vice versa. You can also manipulate Micron disks and files on an AOS system.

Organization

We have organized this part of the manual similarly to the previous section. The text contains general descriptions of Micron features and the tables contain more detailed information, such as lists of commands or system calls. The tables of system calls are there to give you an idea of the kinds of easily-referenced routines built into the Micron system.

Chapter 2
Facilities of the Micron System

File Management

The File System

As you may recall, data in a disk-based Microproducts system is stored on disk and/or diskette. One of the purposes of the operating system is to allocate and de-allocate storage space for the data and to organize it so that it is easily accessible. The data is organized according to a file system which minimizes input/output operations and separates the logical characteristics of files from the physical characteristics of a device. A file is simply a collection of data. The data might be text, binary numbers, instructions, etc.

Disk devices are random access devices, which means they read or write data blocks in any order. This makes it possible for the Micron system to have an indexed file system. The basic unit of storage in this file system is the file element, consisting of one or more contiguous disk blocks (512 bytes).

The size of the file elements for each file in your system can be controlled. If the element size is large, reading and writing of data is done quickly since the disk heads don't have to move around the disk as much to find the data. If, on the other hand, the element size is small, it is easier for the system to allocate disk blocks and therefore use space efficiently.

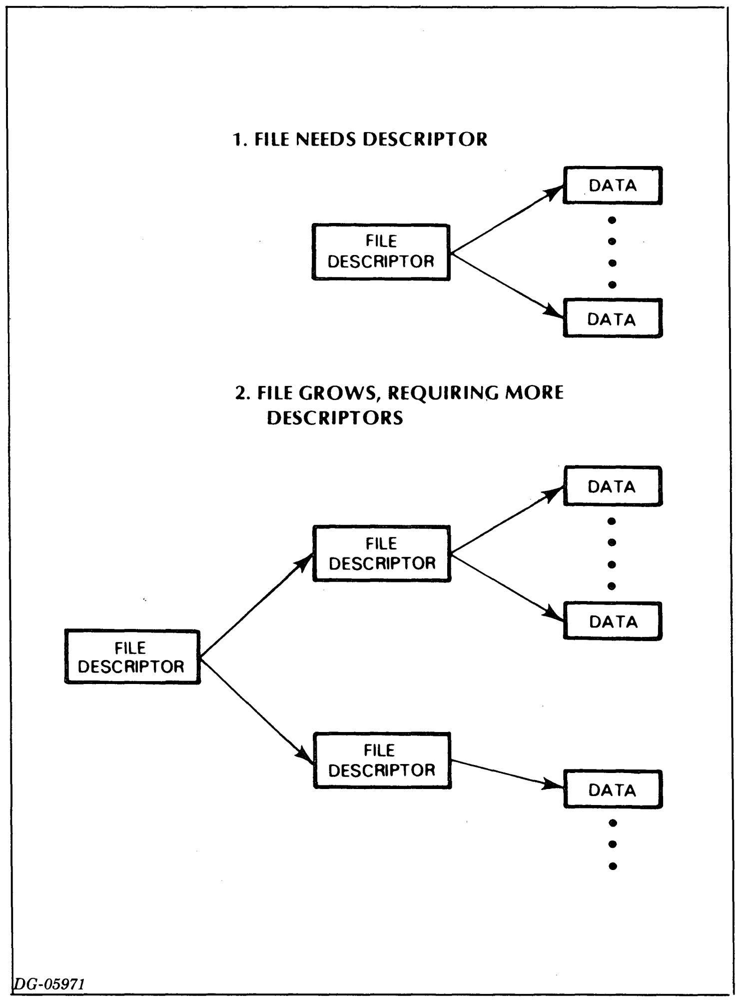


Figure 2.1 How a disk file grows

When a file grows beyond 36 file elements, the Micron system allocates a full 256-word (512 byte) disk block to serve as a file descriptor to both the old and new data elements. If the file continues to grow, and the index (the file descriptor block) is used up, the Micron system creates another level of indices. The advantage of such indexing is that files can expand without requiring physically contiguous storage.

The files in a Micron system are grouped into directories, which are simply files that can contain other files. They can also contain directories, which, in turn, can contain other data files and directories. Directories can nest to any level, and the result is a hierarchical or tree-structured file system.

Each disk device has a root directory which contains other files and directories in a tree structure. Since it is up to you how you structure the tree, you have a great deal of flexibility in organizing your data in a way that seems most logical and convenient. You can, for instance, have one directory containing many other directories; several levels of directories; or a combination of the two.

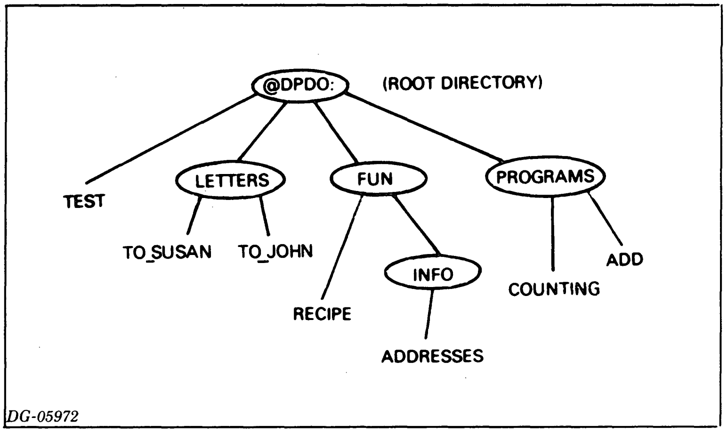


Figure 2.2 A hierarchical file system under the Micron system

System Call	Action
?ALIST	Add a name to the searchlist.
?CREATE	Create a file.
?DELETE	Delete a file.
?DIR	Select a working directory.
?FSTAT	Get a file's status information.
?GLIST	Get the searchlist.
?GNAME	Get the fully qualified pathname.
?GTATR	Get file attributes.
?RENAME	Rename a file.
?STATR	Set file attributes.

Table 2.1 File management system calls

Device-Independence

So far we have discussed files which consist of data and reside on a disk device. A file, however, can also be a device (i.e., a disk, terminal, line printer, etc.). All files have filenames to identify them. The fact that devices are referenced as files simplifies the writing of programs because you don't have to worry about the unique characteristics of a particular device. This is known as device-independence.

Input/Output

A program running under the Micron system can communicate with up to 16 files or devices at a time. It communicates with them by means of *I/O channels* which are software-defined data paths.

Types of I/O

There are two types of I/O available under the Micron system:

- Dynamic
- Data Sensitive

Your choice of I/O mode depends on your application.

Dynamic I/O

Dynamic I/O lets you transfer any number of bytes at a time, including entire disk blocks (512 bytes), between the file and main memory. Transferring entire disk blocks increases speed significantly.

Data Sensitive I/O

In data sensitive I/O you can transfer an arbitrary number of bytes between an I/O channel and main memory until a delimiter is reached. Delimiters are characters such as New-line or Form Feed, and they signal the end of a line of characters. This mode of I/O is especially convenient for text or other line-oriented data.

Non-Pended I/O

Many system calls take a long time to execute, and they suspend your program while they are executing. As a result, potentially useful processor time is lost. Non-pended calls avoid this waste. When the NP option is added to a system call, the program continues its operation while the system call is executing.

System Call	Action
?AWAIT	Await completion of a non-pended system call
?CLOSE	Close an I/O channel.
?GPOS	Get the current file position.
?OPEN	Open an I/O channel.
?READ	Read data from device or file.
?RESET	Close multiple I/O channels.
?SPOS	Set current file position.
?WRITE	Write data to device or file.

Table 2.2 I/O system calls

I/O Device Management

The Micron system supports a variety of input and output devices, including disk drives, terminals, line printers, and paper tape readers. The devices fall into two categories: those with directory structures (disks) and those without directory structures (character devices).

Disk Devices

System calls are available for mounting and dismounting disks. When you mount a disk, you "introduce" it to the system, and the supervisor program checks the disk to make sure it has a valid Micron directory structure (see *Disk Initializer*). When you dismount a disk, you ensure that any data in memory is written out before the disk is removed.

Console Devices

The Micron system supports many different types of consoles. Consoles have a number of characteristics (such as tab simulation or character echoing) which you can set with system calls. In addition, the Micron system provides a number of control characters which allow you to change or stop what is happening at the console. For example, you can erase a line of input or terminate the currently running program.

Special Device Support

If you have non-standard devices in your system or if you need extremely fast response to interrupts, the Micron system allows you to define an interrupt handler. An interrupt handler is a routine within your program which services a specific I/O device. If you have defined an interrupt handler, and the supervisor detects an interrupt from the specified device, it transfers control to your routine. This feature is particularly useful in real-time applications.

System Call	Action
?BOOT	Restart the system.
?DISMOUNT	Remove a disk from the system.
?DSTAT	Get a disk's status information.
?GCHAR	Get device characteristics.
?IDEF	Define an interrupt handling routine.
?IRMV	Remove an interrupt handling routine.
?IUNPEND	Unpend a task from interrupt handling routine.
?IXIT	Exit from an interrupt handling routine.
?MOUNT	Introduce a disk to the system.
?SCHAR	Set device characteristics.

Table 2.3 Device management system calls

Multi-Tasking

Multi-tasking is a powerful technique which greatly simplifies writing code for a program which must "do several things at once." The system allows you to divide your program into a number of sub-programs called tasks. A routine called the task scheduler switches control among the various tasks. Switching between tasks is very rapid, thereby creating the illusion of parallel processing. Also, as we will explain below, multi-tasking can essentially transform a single-user system to a "multi-user" system.

Multi-tasking is particularly useful in something like a multi-user editing or data entry system which supports a number of people working at consoles. Without multi-tasking the program would have to contain some sort of scanning routine that checked all users to see who needed service. Under the Micron system you simply assign a separate task to each user. The system takes charge of deciding which user to service, freeing you from the necessity of writing a long and complex scanning routine.

You create and delete (kill) tasks with system calls. As many as 255 tasks can be active at once. When you create a task, you specify its priority. The system always runs high priority tasks first. Lower priority tasks run only when all higher priority tasks are blocked from running. You can change the priority of a task at any time with a system call.

System Calls	Action
?CTASK	Create a task.
?DRSCH	Disable task re-scheduling.
?ERSCH	Enable task re-scheduling.
?KTASK	Kill a task.
?MYID	Get task identity
?PEND	Suspend a task.
?PRI	Change task priority.
?UNPEND	Resume execution of a task.

Table 2.4 Multi-tasking system calls

There are times when multi-tasking should be suspended. For example, if you require a task to read and modify a critical memory location, you don't want some other task doing so at the same time. You therefore have the option of disabling the task scheduler so that no other tasks can run.

Tasks may use system calls to communicate with one another and synchronize their activities, by suspending and resuming operation.

Program Management

You can do the following to manage programs under the Micron system:

- Transfer control from one program to another.
- Create a re-startable program file.
- Pass a message between programs.

The Stack

Program management under the Micron system is largely based on the concept of program level. The system maintains information about programs in an 8-level stack that your program can manipulate with system calls. A program can call another program nested in the stack.

Program Swap

When the system is first brought up, the Command Line Interpreter (CLI) is invoked (see Chapter 7, Micron Utilities). The CLI is the first program to enter the stack and is at program level 1. You can use the CLI to call another program, in which case the CLI is blocked, and all information about its operating state is saved in the stack. Your program executes at level 2. This operation is known as program swap. When your program finishes running, the system removes it from the stack and returns control to the calling program (CLI).

Program Chain

Alternatively, your program can transfer control to a companion program, causing it to execute at the same level. This operation is called a program chain. The difference between a swap and a chain is that, in a chain, the state of the calling program is lost.

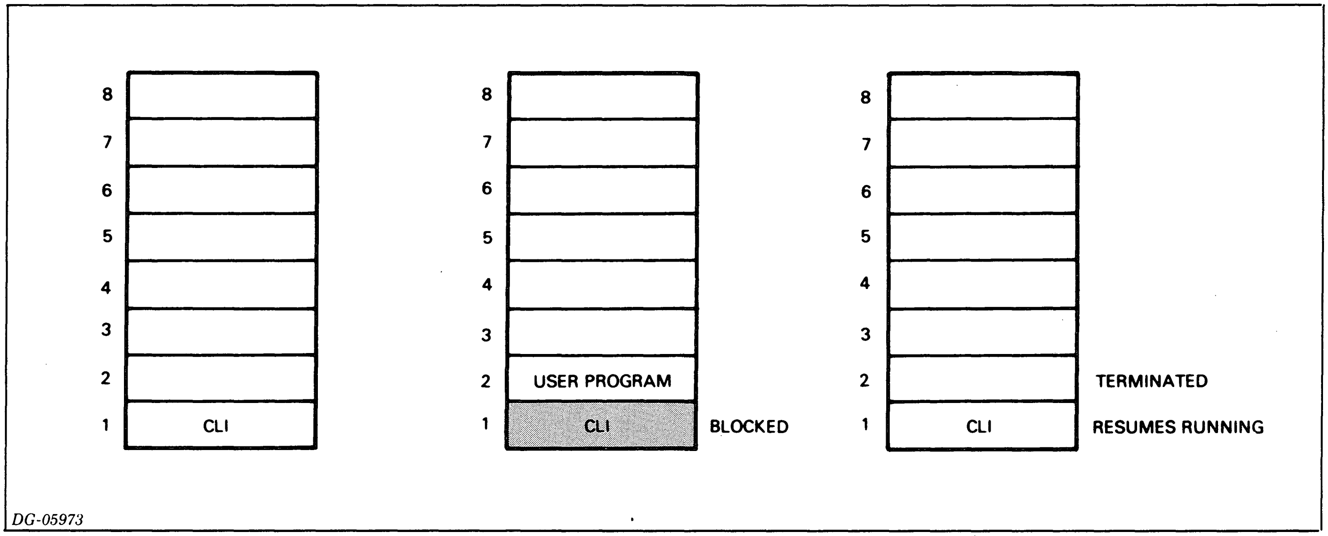


Figure 2.3 Program swap

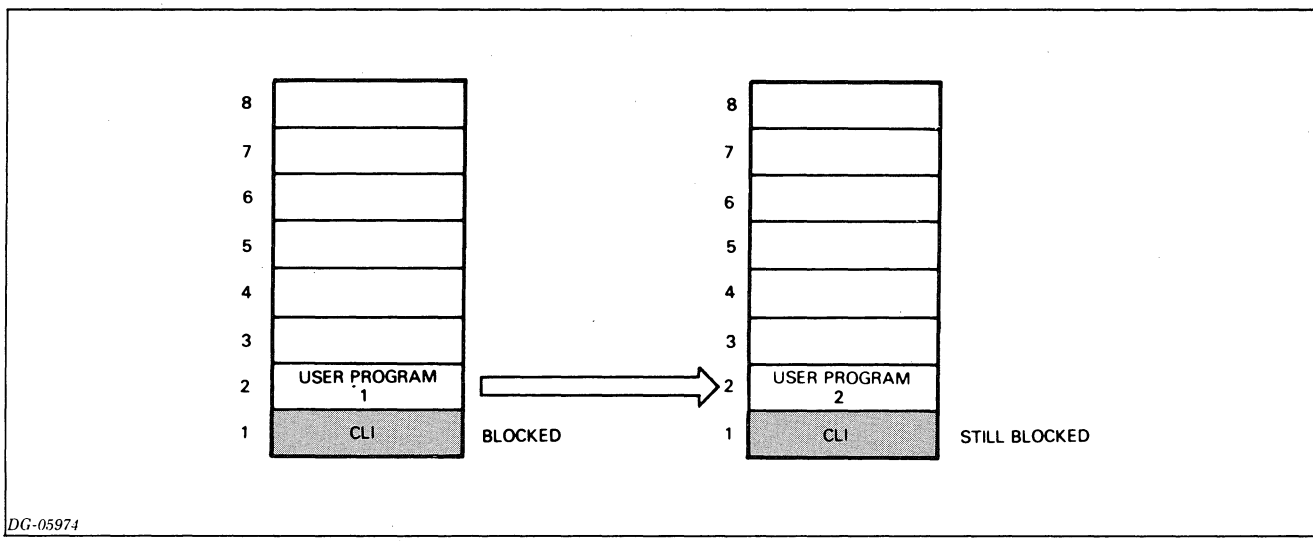


Figure 2.4 Program chain

Break Function

When it terminates, your program can ask the system to preserve its operating state so it can resume execution at a later time. By using the break function you create a break file which is stored till you need it. You can also cause program breaks at any time from the console.

Inter-Program Communication

The Micron system allows programs to send messages to one another. A message can be up to 2047 bytes long. Its format is entirely up to the user.

The operating system also passes I/O channels between programs. Active I/O channels are passed to a new program so that it doesn't have to re-open them. The operating system always opens two channels for console I/O and passes them to other programs since almost all other programs use them.

The CLI, however, always closes all other I/O channels before calling another program.

System Calls	Action
?EXEC	Execute a program.
?GTMSG	Get an inter-program message.
?RETURN	Return to next lowest program level.

Table 2.5 Program management system calls

Memory Management

Micron programs have two main data areas: pure and impure. The pure area consists of code and data that is never modified during a program's execution.

The impure area consists of code and data which is modified. Having a pure area increases the efficiency of the system since the pure area does not have to be saved when the program is swapped. When the program resumes execution, the pure area can be recovered from the program file. In dedicated applications you would store your pure code and data in ROM (read-only) memory. A program development system, on the other hand, would have both pure and impure areas stored in RAM (read/write) memory.

The Micron supervisor and your program both occupy the same address space so there is the potential danger of your program accidentally overwriting part of the system. The system therefore provides system calls which allow a program to acquire and release memory in an orderly manner and which inform programs about their memory usage.

Overlays

In a typical program, many routines are not used very much. Overlaying is a technique which allows you to reduce the memory requirements of a program by having infrequently used routines reside on disk until actually needed.

When writing a program using overlays, you form one or more overlay nodes. An overlay node is a block of memory which is associated with a set of overlays. The node can hold one of the overlays at a time. The overlays are simply the routines or groups of routines which you don't execute frequently.

The system keeps track of which overlays are currently loaded. If a program is multi-tasked, several tasks may share a node if they request the same overlay.

System Calls	Action
?INFO	Get program information.
?MEMI	Change impure memory allocation.

Table 2.6 Memory management system calls

Library Routines

The Micron system provides a number of convenient routines which some users are willing to forego for the sake of speed. These functions are implemented as library routines, instead of system calls. You call a library routine in the same manner as a system call, but the code which implements the function is part of your program rather than residing in system memory.

Routine	Action
?CDAY	Convert system time/date to date.
?CTOD	Convert system time/date to time of day.
?DELAY	Delay execution of a task.
?ERMSG	Retrieve a system error message.
?GNFN	Get the next filename in the working directory.
?MSEC	Convert a time to milliseconds.
?OVLOD	Load an overlay.
?OVREL	Release an overlay file.
?SLIST	Set the searchlist.
?TMSG	Translate a CLI-format message.

Table 2.7 Micron library routines

Chapter 3
Micron Utilities

Micron includes several utility programs:

- Command Line Interpreter (CLI)
- Speed Text Editor
- Macroassembler
- Binder
- Library Editor
- Debugger
- Move Utility
- Disk Initializer
- Disk Fixup
- File Editor
- PROM Burning Utility
- System Generation
- AOS File Transfer
- File Display and Comparison Utility

These utilities perform basic services, as described below.

Command Line Interpreter (CLI)

The Command Line Interpreter (CLI) is a program which allows you to interact with the Micron system. By typing in commands at the console, you can communicate with the operating system. The CLI allows you to maintain the Micron file system and to display or set certain system variables.

You can also use the CLI to invoke the other Micron utilities and your own programs. Since the CLI starts running as soon as the Micron system is brought up, you can start your applications right away.

Basically, the CLI is like lots of little utilities. It is equivalent to a CLI on big systems, but has far more power than is typical on a microcomputer system.

An important feature of the Micron CLI is its macro facility. This facility allows you to create your own certain written commands. These macro commands are included during arguments which allow you to write a generalized macro, and use it for a variety of specific purposes later. The macro facility can save you a great deal of time and effort.

Another feature of the Micron CLI is the **HELP** facility which you can use to get information about CLI commands and general topics.

The Micron CLI is functionally identical to the AOS CLI.

Command	Action
ATTRIBUTES	Set or display a file's attributes.
BOOT	Read bootstrap program from disk and execute it, after shutting down the currently executing system.
BYE	Terminate the CLI.
CHAIN	Overwrite CLI with program named in pathname.
CHARACTERISTICS	Set or display device characteristics for specified device.
COPY	Copy one or more files to a destination file.
CREATE	Create a file.
DATE	Set or display current system date.
DEBUG	Execute specified program and enter Debugger.
DELETE	Delete one or more files.
DIRECTORY	Set or display current working directory.
DISKSTATUS	Return information about space available on specified disk and about disk's error history.
DISMOUNT	Prepare specified disk for removal.
EXECUTE	Execute the specified program.
FILESTATUS	List file status information.
HELP	Explain a CLI command or general topic.
INFORMATION	Display program information.
MESSAGE	Display text message corresponding to error code arguments.
MOUNT	Make disk accessible for I/O.
PATHNAME	Display a complete pathname.
RENAME	Change a file's name.
REVISION	Set or display a program's revision number.
SEARCHLIST	Set or display the searchlist.
TIME	Set or display the current system time.
TYPE	Type the contents of one or more files.
WRITE	Display argument string.
XEQ	Execute the specified program.

Table 3.1 CLI commands

Speed Text Editor

Speed is a character-oriented text editor. It allows you to do such things as create, modify, and merge text files; insert, delete, and change individual characters, lines, or blocks of text; and search for specific words, strings or patterns. You have the option of using display mode for a constant image of your most recent updates and actions.

Obviously, Speed is useful whenever you are working with text. In program development you can use Speed to write and correct program source files.

Commands that open and close files.
File input commands.
Text type-out commands.
Character pointer commands.
Search commands.
Insertion commands.
Deletion commands.
Buffer commands.
File output commands.
Exit command.
Input mode commands.
Case control commands.
Iteration commands.
Flow control commands.
Program execution commands.
Commands that manipulate numeric arguments.
Commands that manipulate numeric variables.

Table 3.2 Categories of Speed commands

Macroassembler

The Macroassembler (MASM)

- Assembles source files into machine language object files.
- Allows the use of pseudo-ops in your programs to control assembly.
- Has a macro facility to simplify the process of assembly language programming.

In program development you would assemble the assembly language source files, which you wrote using Speed, to produce object files.

Binder

The Binder (BIND) produces program files which can be executed. It

- Binds together object files and libraries to form program files.
- Builds symbol tables.
- Has an overlay facility which lets you execute more code than fits into your user address space.

You can use the Binder to bind object files or libraries which were produced either under the Micron system or under the Advanced Operating System (AOS). This means that you can write and assemble programs under AOS and run them under the Micron system. You may also generate stand-alone and PROMable programs under AOS and run them under the Micron system.

Library Editor

The Library Editor (LED) allows you to build libraries of object files to be used by the Binder when producing program files. It provides a convenient way to store a group of related subroutines, since the Binder can scan it quickly and retrieve only those modules which are referenced by your programs.

Symbolic Debugger

The Symbolic Debugger (DEBUG) lets you see how your program is executing and helps you detect and remove mistakes. It allows you to do such things as:

- Set breakpoints and conditional breakpoints control the flow of execution
- Examine the contents of memory locations and special registers
- Display the contents in different formats such as octal numbers or assembly language

Memory monitoring commands.
Breakpoint and conditional breakpoints commands.
Search commands.
Accumulator monitoring commands.
Special register monitoring commands.
Disable symbol recognition command.
Frame manipulation command.
Single value display formats.
Output device command.
Program execution commands.

Table 3.3 Categories of Debugger commands

Move Utility

The Move utility allows you to move copies of files from one directory to another. This utility is particularly useful in creating back-up files. A back-up file is a copy of a file put on a medium which can be removed from the system and stored in a safe place. If anything happens to the working copy on the system, you have the back-up file.

The command which invokes the Move utility accepts a number of "switches" which alter the command. These switches (shown in Table 3.4) allow you to move files according to preselected constraints.

Switch	Action
/AFTER/TLM= *DD-MMM-YY:HH:MM:SS*	Only move files modified after date *(DD-MMM-YY)* and time *(HH:MM:SS)*.
/BEFORE/TLM= *DD-MMM-YY:HH:MM:SS*	Only move files modified before date *(DD-MMM-YY)* and time *(HH:MM:SS)*.
/DELETE	Delete any file in destination directory which has same name as file being moved.
/FROM	Move files from specified directory to current working directory.
/L	Print name of all files moved.
/L=*filename*	List all files moved in specified file.
/RECENT	If there is file with same name in destination directory, only move file if it was created more recently.
/V	List name of each file on the console as the file is transferred.

Table 3.4 Move command switches

Disk Initializer

The Micron system stores data on disks in complex structures. These structures are invisible to you and are designed to allow efficient access to your data files. They are also designed to be easy to reconstruct in case of system failure. The Disk Initializer (DINIT) writes the initial formats of these structures onto a new disk. You must format a disk(ette) with the Disk Initializer before you can store any data on it. The Disk Initializer destroys any old information on the disk as it formats it.

The information in the disk format includes the number and location of bad blocks; the location of free space; and the names and locations of all the files on the disk. This information is stored at special locations on the disk.

The Disk Initializer also installs the operating system, its bootstrap, and fixup if it is requested to do so.

Disk Fixup

The Disk Fixup utility (FIXUP) verifies that the directory and file structures in a disk are correct and consistent, and it repairs them if necessary. It may be run automatically on system setup.

File Editor

The File Editor (FEDIT) allows you to examine and modify locations in any Micron file, using a format similar to the Debugger.

PROM Burning Utility

The PROM Burning Utility produces programmed read-only memory from stand alone files, by either direct connection of a PROM burner to an asynchronous line or generation of a tape.

System Generation

The System Generation Utility (SYSGEN) generates a tailored version of Micron systems for either stand-alone run-time use, or for non-standard peripheral configurations.

AOS File Transfer

The AOS File Transfer utility allows you to manipulate Micron disks and files on an AOS system. You can move files to and from a Micron disk, as well as create sub-directories, rename files, and list the contents of the disk. The ability to perform these operations on an AOS system increases the flexibility of your Microproducts system.

Command	Action
BYE	Terminate AOS File Transfer session.
HELP	Display a list of all AOS File Transfer commands.
CREATE	Create a directory.
DELETE	Delete a file.
DUMP	Move file from Micron disk to AOS system.
LIST	List files in a directory of Micron disk.
LOAD	Move files onto a Micron disk.
RENAME	Change name of file on Micron disk.
SPACE	Display number of bytes on disk that are in use and number still available.

Table 3.5 AOS file transfer commands

File Display and Comparison Utility

The File Display and Comparison Utility (FDISP) will display files. It can also perform file comparison.

Program Development with the Utilities

The Micron utilities are designed to facilitate quick, interactive development of your own application programs. For example, you can use Speed to write assembly language source files and then assemble them with the Macroassembler to produce object files. Next you can use the Binder to merge the object files into a program file. Once you have a program file you can use the Debugger to see if your program executes as required. If at any point in this process you detect mistakes, you can go back and use Speed to correct them, then re-assemble and re-bind the files till the program is as you want it.

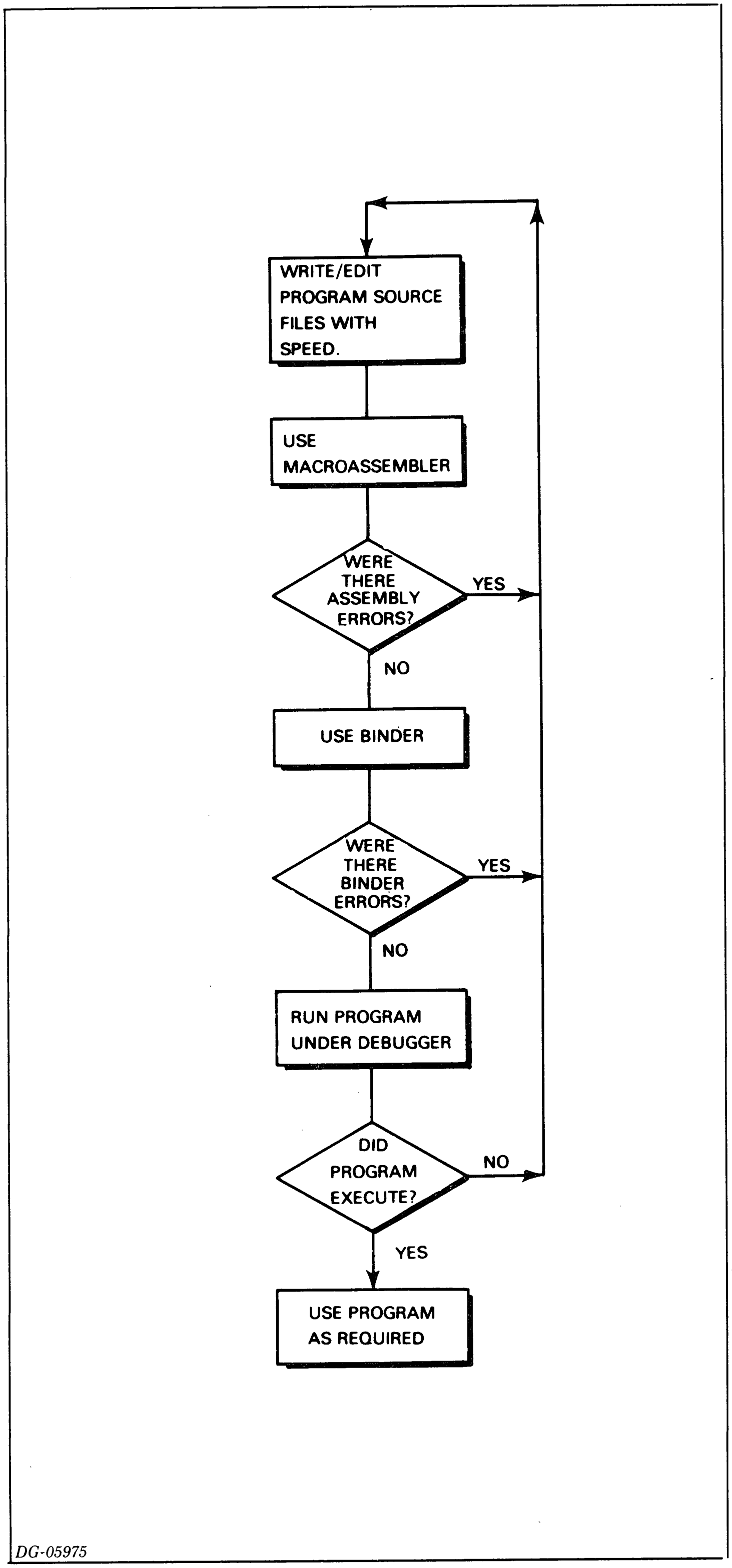


Figure 3.1 Program development under the Micron system

Cross-development On AOS

As we have mentioned before, the Advanced Operating System (AOS) is a powerful multi-programming system that runs on Data General's ECLIPSE line computers. Since Micron system calls are a functionally compatible subset of those on AOS systems, you can run a program on either system if it uses only Micron calls. The two systems, however, are different internally, so you need translating software to run Micron programs on AOS systems.

DGC provides translating software that consists of an object file which is bound along with your program. The result is an AOS-format program file. This means that you only have to re-bind a program if you want to transfer it from AOS to the Micron system (or vice versa).

Micron utilities are either functionally identical to there AOS counterparts (e.g., the CLI), or they are available under AOS (e.g., the Binder).

Chapter 4
High-Level Languages Under The Micron System

The Micron system supports two high-level languages: MP/Pascal and MP/Fortran IV. These languages can be used for program development in a large variety of applications.

MP/Pascal

Pascal was developed in the early 1970's by Niklaus Wirth. It has simple notation since it was originally designed to be a teaching language. MP/Pascal is based on Wirth's Pascal but has many useful extensions.

Pascal has the following advantages:

- A block structure that lends itself easily to structured programming techniques.
- A data-typing facility for creating new data types.
- A record structure composed of different data types to make structuring information easy.
- A case statement for multi-way branching.

Data General extensions include:

- A string type and string operations.
- PROMable programs
- Enhanced I/O.
- An include file facility
- Separate compilation units within a program
- Access to all Micron operating system facilities (including overlays.)
- Easy interface to external procedures written in assembly language.

```
CONST
   MINRADIX = 2;
   MAXRADIX = 10;
   1_DIGSTR = 17;      {room for the largest value of the min radix}

TYPE
   digstr = string 1_digstr;
   rdx = minradix .. maxradix;

{
   Convert the given integer to a number in the indicated radix.
   Return the result in the string parameter.
   If the input number is negative, then prefix the result with
   a minus sign.
}

PROCEDURE INT_TO_STRING ( num: integer;
                           radix : rdx;
                           var outstr : digstr );
   VAR
      temp_num : integer;
   BEGIN
      if num <0 then begin
        outstr :='-';
        temp_num := - num;
      end else begin
        outstr := '';
        temp_num := num;
      end;
      repeat
        append (outstr, chr ( (temp_num mod radix) + ord ('0') ) );
           {slightly more complex logic is needed for radices > 10}
        temp_num :=temp_num div radix;
      until temp_num = 0;
   end;
```

DG-06048

Figure 4.1 A sample MP/Pascal procedure

MP/Pascal can be used in applications such as:

- **Systems Programming** - MP/Pascal is a powerful language for systems programming; many of the Micron utilities are written in it.
- **Scientific** - MP/Pascal is an important scientific tool since it has real and integer data types and arrays. It also allows user-defined data types.
- **Business** - MP/Pascal is a versatile and useful business tool since it has a record structure, and integer and data types.

MP/Fortran IV

Fortran is one of the oldest high-level languages. It was designed by a committee of IBM computer scientists in the mid-1950's, and has been refined and developed by a large community of users.

```
        INTEGER FIB
        DIMENSION FIB(20)
c
        FIB(1) = 1
        FIB(2) = 1
c
c    Loop to calculate the next 18 numbers in the sequence
c
        DO 10 I = 3.20
10          FIB(1) = FIB(I-1) + FIB(I-2)
c
c    Print the results
c
        DO 20 I = 1,20,10
        J = I+9
20      WRITE(12,100) (FIB(K),K=I,J)
100     FORMAT (1X,10I6)
c
        STOP
        END

       1    1    2    3    5    8    13    21    34    55
      89  144  233  377  610  987  1597  2584  4181  6765
```

DG-06047

Figure 4.2 A sample MP/Fortran IV program

Data General's MP/Fortran IV is a variation of ANSI Fortran standard X3.9-1966. It has the following features:

- A wide variety of data types including integer, real, complex, and double precision real and double precision complex variables.
- Many built-in, pre-defined functions: trigonometric, hyperbolic, absolute value, truncation, and logarithms.
- Formatted I/O.
- Random access I/O.

Data General enhancements include:

- PROMable programs.
- More flexible use of constants in parameter statements.
- Variable names up to 31 characters long.
- Expansion of the maximum number of dimensions for arrays from 7 to 128.
- A set of routines in the subroutine library for accessing the operating system's capabilities.
- Multi-tasking.

MP/Fortran IV can be used in applications such as:

- **Industrial** - MP/Fortran IV is ideal for real-time industrial control due to its multi-tasking capability and PROMability, which allow dedicated systems applications.
- **Mathematical and Statistical** - Many packaged programs for mathematical and statistical problem-solving are already available in Fortran. If you need to design your own programs, MP/Fortran IV's many built-in mathematical functions and extended array-handling capabilities will help expand your computational range.

Part 3
Typical Applications

Typical Applications

In this part of the manual we give examples of typical applications with the Microproducts line.

Industrial Automation

Possible applications in the area of industrial automation include:

- Process Control
- Building Management
- Test Facility Control
- Pollution Monitoring
- Quality Control
- Bulk Liquid Terminal Control
- Security Control

You can use Microproducts systems in a variety of industrial automation applications which require an economical computer system and controls at each unit of production. Microproducts systems have simple interfaces and the cabling required for configurations with a variety of sensor control elements.

Example: Alarm System

Modules:

- MP/100 with 16Kb PROM and 32Kb RAM
- Digital-to-Analog Converter
- Analog-to-Digital Converter
- 2 Digital I/O Interfaces
- 1 Relay Controller

This Microproducts system constantly monitors heat and smoke sensors to see if they have detected a fire. During off hours, the system automatically turns off heat and light to conserve energy. It also monitors door and window sensors to see if they have detected a burglar/intruder. You can perform all but the fire detection from a central control station.

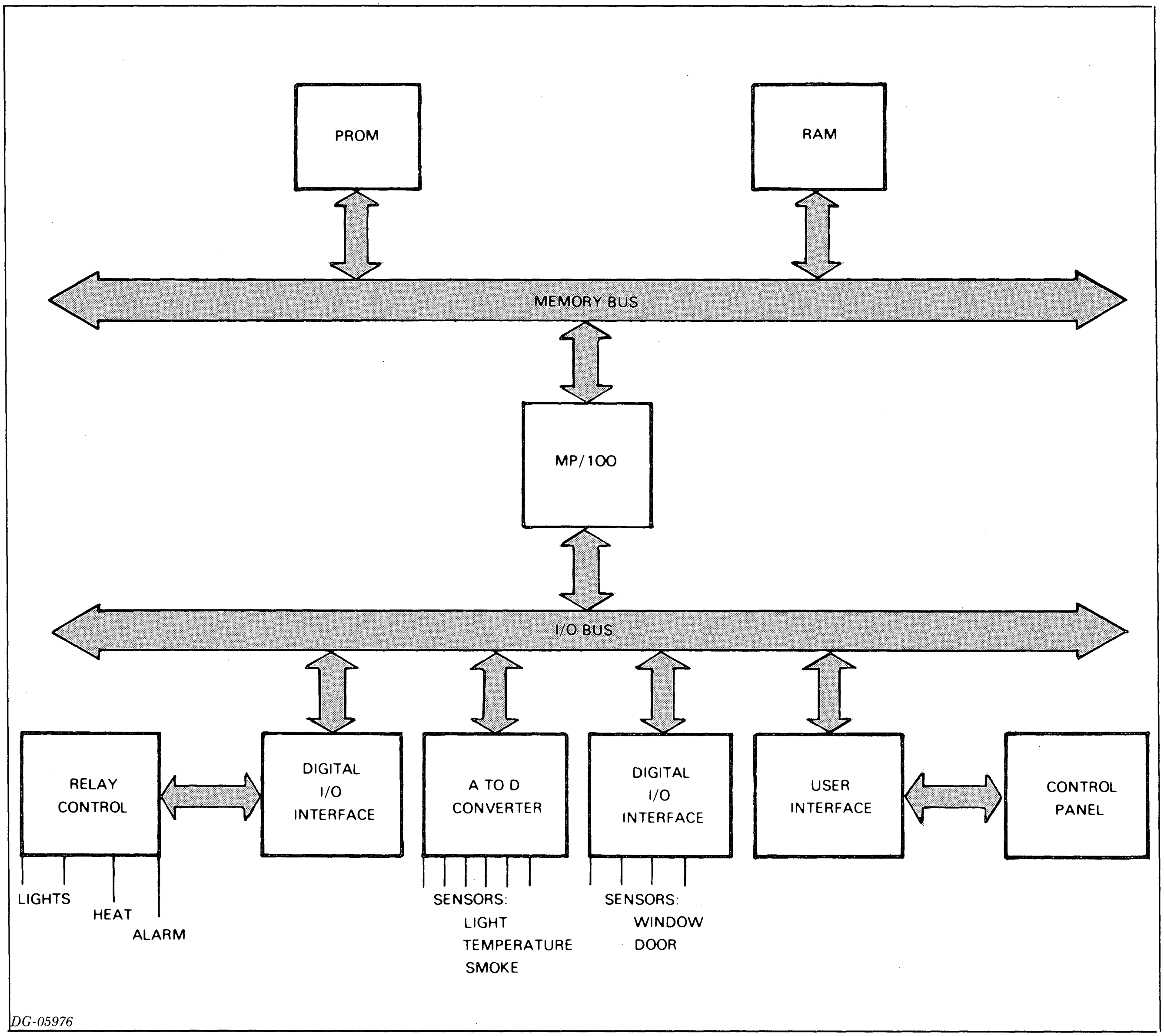


Figure 3.1 Alarm system

Communications Systems

Possible applications in the area of data communications include:

- Remote Terminals
- Data Concentrators
- Switching Systems
- Information Display
- Data Entry and Verification
- Intelligent Terminals
- Satellite Communications Controllers

Microproducts systems provide the high throughput and growth possibilities required by communications systems. MP/computers also have architectural compatibility with other Data General computers so they can be used with economy and versatility in large communications systems.

Example: Data Collection Concentrator

- MP/200 with 64Kb RAM
- 10 Mb Disk Subsystem
- Communications Controller
- Synchronous Line Controller (SLC) with CRC Generator
- 2 Asynchronous Line Multiplexors (ALMs)
- 8 DASHER Display Terminals

This data collection concentrator collects data from a number of terminals and validates and buffers data. It then transmits the data to a host computer.

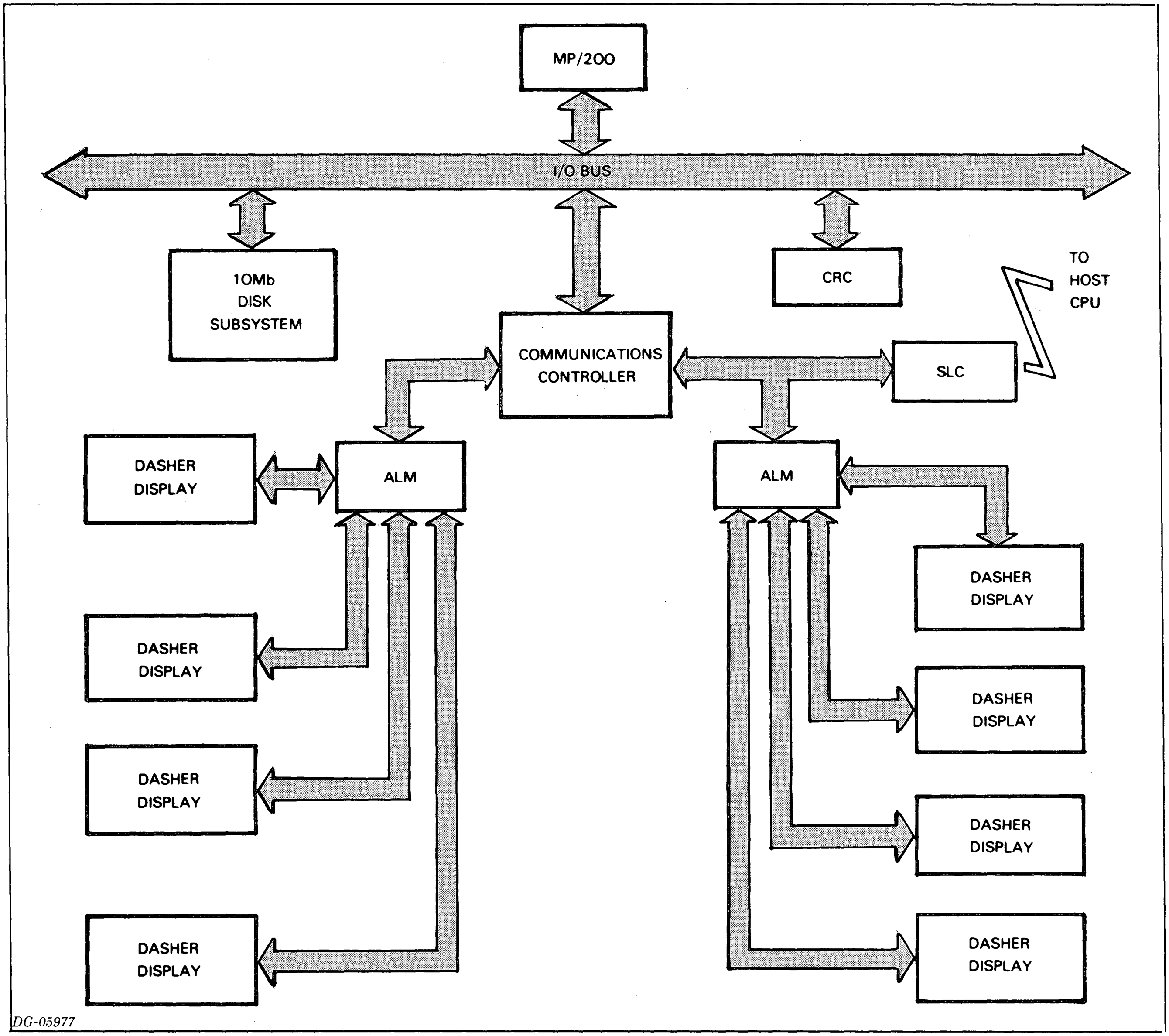


Figure 3.2 Data collection concentrator

Data Acquisition and Control

Possible applications in the area of data acquisition and control include:

- Laboratory Automation
- Sample Analysis
- Inventory Control
- Optical Measurement
- Railroad Control Systems
- Nuclear Cargo Monitoring

Example: Multi-site Manufacturing Control System

- MP/200 with 64Kb RAM
- 12.5Mb 6101 Disk Subsystem
- Asynchronous Line Multiplexor (ALM)
- Line Printer Controller
- DASHER LP2 Printer
- 2 DASHER Display Terminals
- 2 DASHER TP1 Printers
- Digital I/O Interface
- Synchronous Line Controller (SLC) with CRC Generator

In this multi-site manufacturing control system, the following operations take place at each site: any time a customer order comes in you enter it into the system, causing a scheduler to allocate time for the job on the basis of job priority and inventory. Once the job is scheduled, the system traces it through the manufacturing process. It subtracts raw goods from inventory and automatically generates bills and re-order forms. Every evening each site transmits manufacturing data to the main computer at company headquarters.

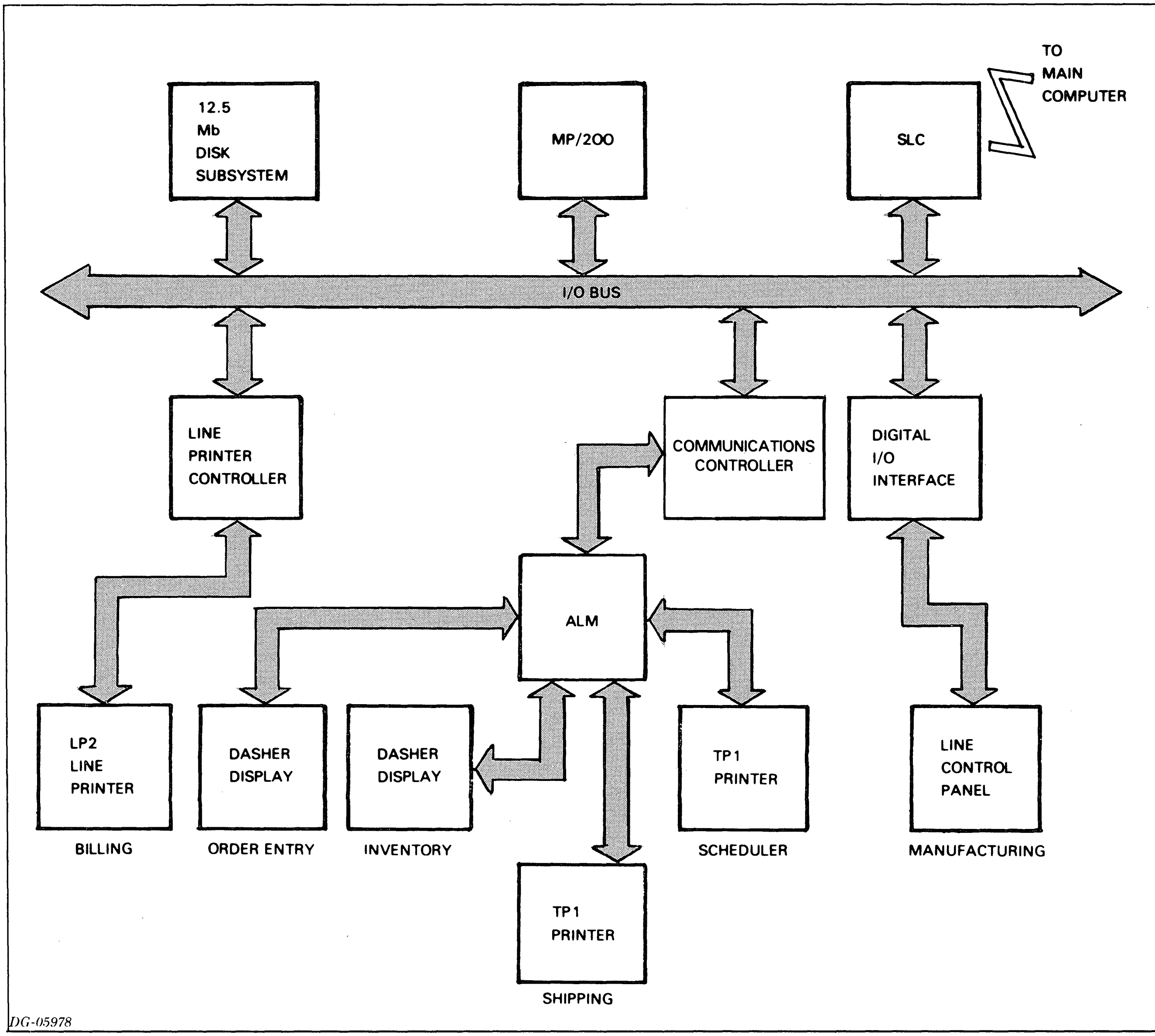


Figure 3.3 Multi-site manufacturing control system

Small Commercial Data Systems

Possible applications in the area of small commercial systems include:

- Inventory Control
- Point-of-Sales Terminals
- Credit Card Verification
- Accounting Systems
- Insurance Policy Record Management
- Automatic Banking

MP/computers are very suitable for small commercial applications since they provide quick access to data and 16-bit accuracy. In addition, MP/Pascal allows library support of double precision integers (32 bits). Both MP/Pascal and MP/Fortran IV support real numbers, and MP/Fortran IV supports double precision real variables.

Example: Bank Account Transaction Control System

- MP/200 computer with 64Kb RAM
- 2 Asynchronous Line Multiplexors (ALMs)
- Diskette Subsystem 6096
- Synchronous Line Controller (SLC) with CRC Generator
- 8 DASHER Terminals

In this transaction control system, an MP/200 system is used in each bank branch. The main data base is stored in a large system like an ECLIPSE or NOVA. The MP/computer systems enter data; detect and correct errors; and transmit the information to the central computer for account verification and transaction execution. The results of transactions can be transmitted back to the Microproducts systems for display and visual verification.

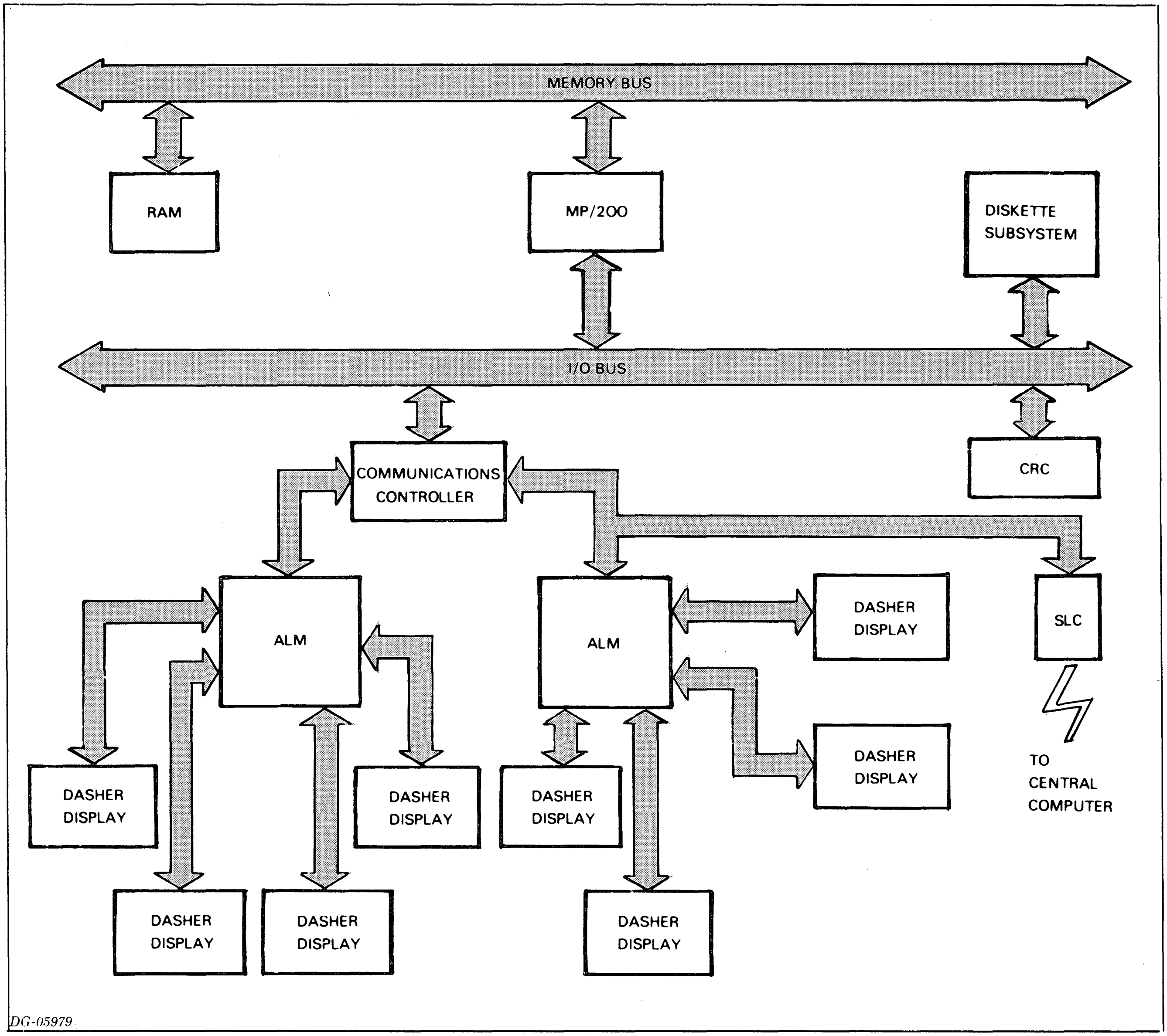


Figure 3.4 Bank account transaction control system

Instrumentation Systems

Possible applications in the area of instrumentation control include:

- Radio Navigation
- Medical Analysis
- Blood Analyzers
- Medical Screening

MP/computers provide the 16-bit accuracy required for superior precision in lengthy calculations. They also provide the high throughput required for complex numerical calculations required for complex calculations in real-time applications.

Example: Blood Gas Analyzer

Modules:

- MP/100 with 16Kb ROM and 32Kb RAM
- Analog-to-Digital Converter
- Asynchronous Line Multiplexor
- 4 DASHER terminals
- Solenoid Controller

In this blood analyzing system, an operator starts the blood sampling. The results of the analysis together with data identifying the patient are sent to the hospital's central computer. The results of the analysis are available at various terminals throughout the hospital.

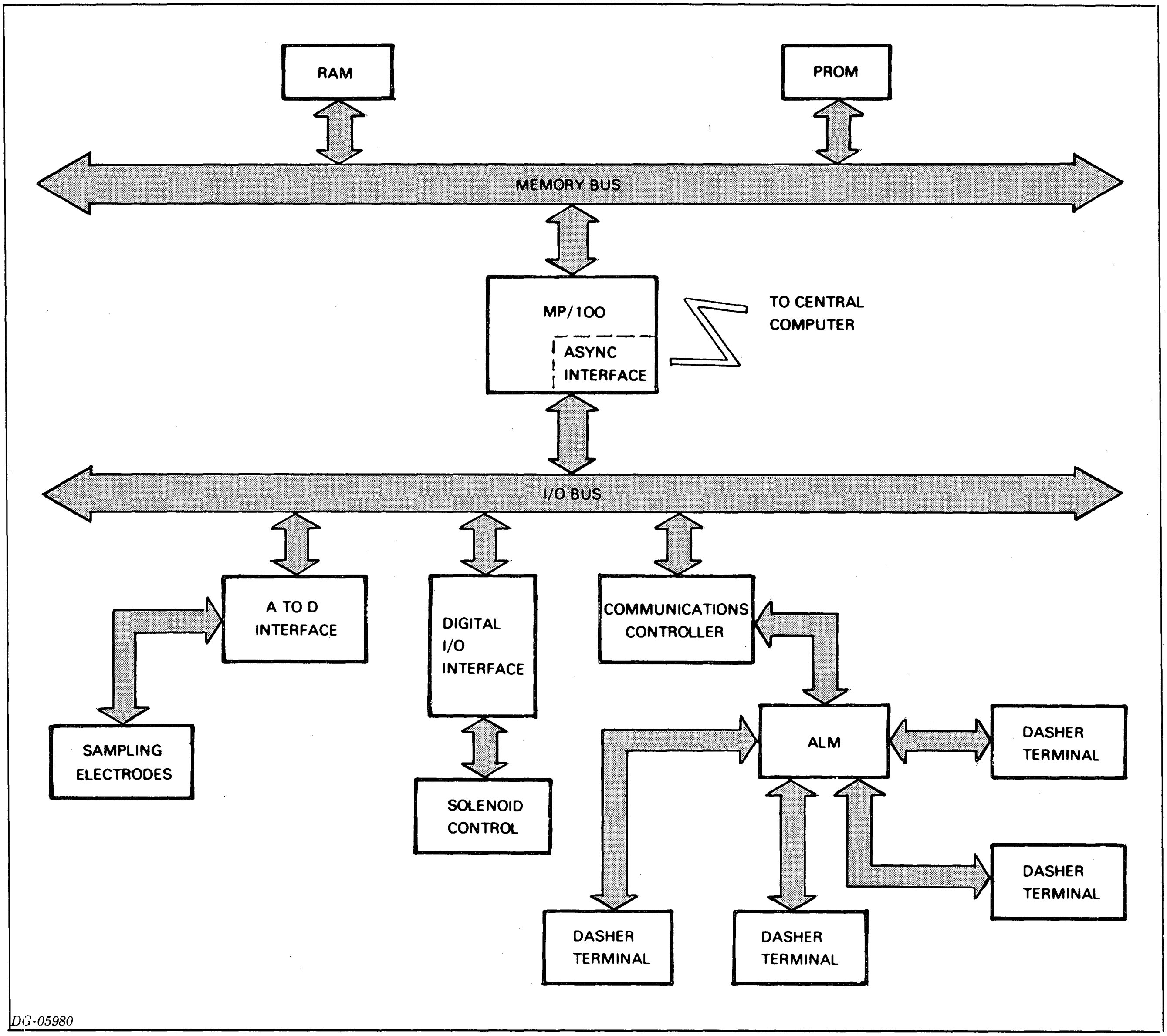


Figure 3.5 Blood gas analyzing

SALES AND SERVICE OFFICES

Alabama: Birmingham
Arizona: Phoenix, Tucson
Arkansas: Little Rock
California: El Segundo, Fresno, Palo Alto, Sacramento, San Diego, San Francisco, Santa Ana, Santa Barbara, Van Nuys
Colorado: Englewood
Connecticut: North Branford
Florida: Ft. Lauderdale, Orlando, Tampa
Georgia: Norcross
Idaho: Boise
Illinois: Peoria, Schaumburg
Indiana: Indianapolis
Kentucky: Louisville
Louisiana: Baton Rouge
Maryland: Baltimore
Massachusetts: Springfield, Wellesley, Worcester
Michigan: Southfield
Minnesota: Richfield
Missouri: Kansas City, St. Louis
Nevada: Las Vegas
New Hampshire: Nashua
New Jersey: Cherry Hill, Wayne
New Mexico: Albuquerque
New York: Buffalo, Latham, Melville, Newfield, New York, Rochester, Syracuse, White Plains
North Carolina: Charlotte, Greensboro
Ohio: Columbus, Dayton, Brooklyn Heights
Oklahoma: Oklahoma City, Tulsa
Oregon: Portland
Pennsylvania: Blue Bell, Carnegie
Rhode Island: Rumford
South Carolina: Columbia
Tennessee: Knoxville, Memphis
Texas: Austin, Dallas, El Paso, Ft. Worth, Houston
Utah: Salt Lake City
Virginia: McLean, Norfolk, Richmond, Salem
Washington: Kirkland
West Virginia: Charleston
Wisconsin: West Allis

Australia: Melbourne, Victoria
France: Le Plessis Robinson
Italy: Milan, Padua, Rome
The Netherlands: Rijswijk
New Zealand: Auckland, Wellington
Sweden: Gothenburg, Malmoe, Stockholm
Switzerland: Lausanne, Zurich
United Kingdom: Birmingham, Dublin, Glasgow, London, Manchester
West Germany: Filderstadt, Frankfurt, Hamburg, Munich, Ratingen, Rodelheim

MANUFACTURER'S REPRESENTATIVES & DISTRIBUTORS

Argentina: Buenos Aires
Costa Rica: San Jose
Ecuador: Quito
Egypt: Cairo
Finland: Helsinki
Greece: Athens
Hong Kong: Hong Kong
India: Bombay
Indonesia: Jakarta
Iran: Tehran
Israel: Givatayim
Japan: Tokyo
Jordan: Amman
Korea: Seoul
Kuwait: Kuwait
Lebanon: Beirut
Malaysia: Kuala Lumpur
Mexico: Mexico City
Nicaragua: Managua
Nigeria: Lagos, Ibadan
Norway: Oslo
Peru: Lima
Philippine Islands: Manila
Puerto Rico: Hato Rey
Saudi Arabia: Riyadh
Singapore: Singapore
South Africa: Johannesburg, Pretoria
Spain: Barcelona, Bilbao, Madrid, San Sebastian, Valencia
Taiwan: Taipei
Thailand: Bangkok
Uruguay: Montevideo
Venezuela: Maracaibo

ADMINISTRATION, MANUFACTURING RESEARCH AND DEVELOPMENT

Massachusetts: Cambridge, Framingham, Westboro, Southboro
Maine: Westbrook
New Hampshire: Portsmouth
California: Anaheim, Sunnyvale
North Carolina: Research Triangle Park, Johnston County

Hong Kong: Kowloon, Tai Po
Thailand: Bangkok

Data General

Data General Corporation, Westboro, Massachusetts 01581