The Big Difference: Sigma 9

XDS

Xerox Data Systems

5 November 1970

To Data Processing Sales:

With this announcement of the Sigma 9, XDS significantly broadens its computer power within the Sigma memory-oriented architecture. Increased speed, more advanced interrupt structure, larger memory, unique data manipulation techniques - all offer our customers a giant step in the Sigma line.
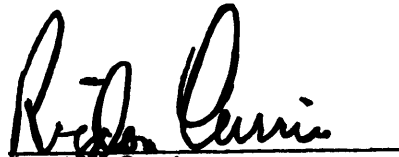
On the other side of the "Sigma 9" coin are two sophisticated operating systems - UTS and XOS. UTS, as you are aware, will be installed before the end of the year. With Sigma 9 it offers extensive multi-use facilities - interactive time-sharing services along with concurrent real-time, local and remote batch - and efficient use of core memory. An excellent system to expand our traditional marketplace.

XOS, an operating system of tremendous versatility, has been under development for over four years. The Sigma 9/XOS system extends the XDS arm strongly into new more commercially-oriented marketplaces, offering these customers a highly sophisticated File Management System, a true multiprogramming system, and a versatile and powerful communications management system. XOS is also available on the Sigma 6 and the Sigma 7.

This notebook, as well as the other material included in this announcement package, has been designed to support you in every phase of your selling effort - from the Sales Presentation Material for initial customer introduction, to a comprehensive Competitive Analysis for effective selling, to names of key support contacts and benchmark procedures.

Sigma 9 truly is "The Big Difference". Let it be the big difference in meeting your quota. Good Luck!

Lou Perillo

Rigdon Currie

ps

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

SECTION I

GENERAL INFORMATION

Xerox Data Systems introduces the Sigma 9, the latest and most powerful member of the Sigma family. This announcement is one of the most significant ever made by XDS. The power of the Sigma 9 opens entirely new areas of applications and, together with the successes in our traditional markets, puts XDS in _the_ challenging position for the 70's.

Sigma 9 is an extension of the Sigma line; its architecture retains the conceptual framework of the Sigma family. The Sigma 9 is memory oriented — with larger memories than previously available with Sigma. The Sigma 9 memory is multi-ported — with more ports than previously available. The Sigma 9 is the system designed to meet the requirements of an expanding multi-programmed, multi-application market, permitting more users per system with excellent response time and throughput.

The Sigma 9 is designed to be fully compatible with the other members of the Sigma family. User programs now running on a Sigma 5, Sigma 6, or Sigma 7 will run without alteration on a Sigma 9 of equivalent or larger configuration.

The Sigma 9 offers the user the greatest growth potential of any system in its class. Memory capacity can grow to 2 million bytes (512K words). The UTS operating system has the capacity to service 128 on-line users, even in a real time environment, and maximize throughput for concurrent batch operations. The XOS operating system incorporates all the ability of UTS with the significant addition of multi-programming. XOS permits an unlimited number of jobs to be core resident with each being serviced on a priority basis since XOS allocates available resources — I/O, memory, CPU time — automatically with minimum overhead.

The new memory with Sigma 9 has greater port capacity; Sigma 9 memory can accommodate up to 12 ports. There may be as many as 11 input/output processors on a system.

1-1

To complement the new CPU and memory of the Sigma 9, a new series of IOP's is being introduced. The new MIOP is designed with a 2-channel capacity. Channel A can interface with as many as 24 device controllers each serviced in a multiplexed mode. Channel B extends the multiplexing operation to an additional eight device controllers. The data rates attainable on the new MIOP are higher than ever before offered by XDS (470,000 bytes/channel). A second IOP being introduced with the Sigma 9 is the HIGH SPEED RAD IOP (HSRIOP). The new HSRIOP incorporates the functions of a selector IOP and a high speed controller in one unit. Up to four high speed RAD's can be connected to this unit, each operating in the burst mode.

Very significant innovations have also been made in the area of system availability, maintainability, and reliability. These improvements assist Field Engineers in keeping the system up and locating system faults. Among these improvements, which are mostly internal hardware, is the capability to perform on-line diagnostics without disturbing system operation.

UTS, available with Sigma 9, offers more time-sharing capability than any other operating system now available. Further, XOS expands the high throughput capability of Sigma 9 by introducing multiprogramming.

In summary, XDS is introducing a significantly new system. The Sigma 9 is a logical upward extension of the current Sigma family. Programs running on current systems are compatible with the Sigma 9, yet Sigma 9 offers wide capability for expansion. Importantly, the concept of system availability has influenced the Sigma 9 design. The customer's basic problem centers around availability and XDS has the solution. Lastly, XDS offers two extremely sophisticated operating systems to match the power of the hardware.

SECTION II
SPECIFICATIONS

2.0   INTRODUCTION

The Xerox Sigma 9 computer system is a versatile, multi-purpose, digital computer. The system
is primarily organized around a high speed central processing unit, fast magnetic core memories,
and flexible input/output processors controlling I/O device controllers and I/O devices. Each
of the major system elements performs asynchronously with respect to other elements in the sys-
tem. Modular design facilitates system expansion to readily accommodate a vast range of com-
puting and data processing requirements.

In the section that follows, a description of the features of the Sigma 9 Central Processing Unit
(CPU), Memory and Input/Output Processors (IOP's) is offered. Also included is a discussion of
the major differences between Sigma 9 and the Sigma 7, and a brief statement concerning the ex-
pected marketplace of the Sigma 9 computer.

2.1   CENTRAL PROCESSING UNIT

Sigma 9 is a logical extension of the successful Sigma family of computers. As an extension of
the Sigma family, Sigma 9 merges the unique architecture of Sigma 5/6/7 with the increased speed
and memory addressing capability of its CPU. The Sigma 9 PSD is shown in the figure on the fol-
lowing page.

2.1.1   MODES OF OPERATION

The Sigma 9 can operate in three different modes. Two of these are identical to the modes avail-
able on Sigma 7; they are the Master and Slave modes. Sigma 9 has a third mode called the
Master Protected mode. The modes of operation are determined by the condition of three bits in
the Program Status Doubleword.

PSW1

0 1 2 3	4	5	6	7	8	9	10 11	12	13 14	15	16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
CC	/////	F S	F Z	F N	M S	M M	D M	A M	A S	/////	E S	INSTRUCTION ADDRESS (IA)

EXTENSION SELECTOR (See paragraph 2.1.2.3)

ASCII MASK (See paragraph 2.1.6.2)

PSW2

32 33 34 35 36 37 38 39 40	41	42 43 44 45 46 47 48 49 50	51 52 53 54 55 56 57 58 59 60 61	62 63
///// W K 0 ///// W K 1 ///// C I I I E I M A /////		EXTENSION ADDRESS (EA)	TRAP STATUS FIELD	REGISTER POINTER (RP) R A /////

(See para 2.1.2.3) (See para 2.1.5.2)

MODE ALTERED
(See paragraph 2.1.1.1
and 2.1.2.3)

REGISTER ALTERED
(See paragraph 2.1.5.1)

Sigma 9 Program Status Doubleword

2.1.1.1 Master Mode (PSD 8=0, PSD 9=x, PSD 40=0)

In this mode of operation, the CPU can perform all of its control functions and can modify any part of the system. The only restriction placed upon the CPU's operation in this mode is that imposed by the write locks on certain protected parts of memory.

Master mode operation is indicated whenever bit 8 of the current PSD is zero. In addition, bit 40 (the mode altered FLAG) must also be zero for Sigma 7 compatibility in the Master mode.

2.1.1.2 Slave Mode (PSD 8=1, PSD 9=x, PSD 40=x)

This mode is identical to the Slave mode on Sigma 7. No privileged instruction may be executed and, if mapping is invoked, the access protection codes always apply to the Slave mode program.

The Slave mode of operation is indicated when the Master/Slave bit (PSD8) is equal to one.

2.1.1.3   Master Protected Mode (PSD 8=0, PSD 9=1, PSD 40=1)

The Master Protected mode is a new mode of operation for Sigma mainframes. A modification of the Master mode, it is designed to provide a certain amount of additional protection for programs which operate in the Master mode.

The Master Protected mode of operation can only occur when operating in the Master mode with virtual addressing. In this mode, if the program makes a reference to a virtual page to which access is prohibited by the current setting of the access protection codes, a trap will occur to the Memory Protection Violation Trap as it does in all Slave mapped programs.

The Master Protected mode of operation is in effect when the Master/Slave bit (PSD 8) is zero and when both the memory map bit (PSD 9) and Mode Altered bit (PSD 40) are set to one. The Mode Altered flag is new on Sigma 9.

2.1.2   TYPES OF ADDRESSING

There are three major types of addressing available on Sigma 9. Two of the major types are identical to the corresponding types of addressing on Sigma 7. The third is an extension required due to the fact that Sigma 9 is capable of having a very large memory.

2.1.2.1   Real Addressing

Real Addressing occurs whenever the memory map is not used and the extended type of addressing is not invoked. This is identical to the unmapped type of operation on Sigma 5/6/7. All addresses are 17-bit real word addresses. This type of addressing allows only the first 128K words of memory to be referenced in an instruction.

2.1.2.2   Virtual Addressing

Virtual Addressing is invoked whenever the memory map indicator is set to one. This type of addressing may be in effect during either the Master or Slave modes of operation. In operation, it is almost identical to the mapped mode of operation on Sigma 6 and Sigma 7. The mapping registers on Sigma 6 and 7 contained eight bits each. On Sigma 9, the mapping register has been extended to 13 bits each. The mapping feature works as follows: The high order eight bits of the original 17-bit reference address in an instruction are used as a pointer to one of the 256 mapping registers. The

13-bit contents of the mapping register replaces the eight high order bits in the original reference address. The 13 bits from the mapping register, combined with the remaining nine bits in the original reference address, give a 22-bit real memory address. This 22-bit address provides addressing of up to four million words of memory.

The compatibility with Sigma 7 operation takes place in the manner in which the map is loaded. When it is loaded by a Sigma 7 program, using only those instructions known to a Sigma 7 program, the lower eight bits of each 13-bit map register are loaded with significant information and the upper five bits will be set to zero. A Sigma 9 program that wishes to take advantage of the large memory may load the map with up to 13 significant bits instead of the eight bits used by Sigma 7.

The virtual memory for any given user at any time can be up to 256 pages (512 words per page). The 512K word memory available on the initial Sigma 9 could contain up to four of these maximum sized users.

2.1.2.3   Real Extended Addressing

Real Extended Addressing is invoked whenever the memory map indicator is zero and the Mode Altered flag is equal to one. Real Extended Addressing is provided on Sigma 9 in order to ease the burden of operating with a large memory for certain types of programs. Specifically, it is intended for the operating system when that system is manipulating users directly and is, in fact, working with real memory and not with the virtual memory of any given user. It also provides a method for writing programs, be they real time or other special purpose programs, which may operate any place within th the memory continuum and not be dependent upon the memory map for their execution.

Real Extended Addressing is discussed in three parts:

- Instruction Addresses and Reference Addresses in Instruction
- Other Addresses and Displacements
- Branching and Branch Addresses

Instruction Addresses and Reference Addresses in Instruction — The Instruction Address field of the PSD and the Reference Address field of each instruction is only 17 bits in length. This is enough to address only 128K words of memory. To extend the addressing capability, the 17-bit address fields in the PSD and instructions are divided into a 1-bit flag field and a 16-bit displacement field (reference the figure on the following page).

2-4

Memory Map

9

PSW1

0

6-Bit Extension Address

Reference or Instruction Address

40 42 43 44 45 46 47

15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

PSW2

1

E S

Mode Altered

0 > Sigma 7 Mode
1 > Sigma 9 Mode

Extension Selector

0 > 0–64K-1
1 > 0– 4M-1
(only if PSD 40 1
and PSD 9 0)

B15 1————————B15 0——

10   15 16   31

22-Bit Extended Address

Conditions:
PSD 9 = 0
PSD 40 = 1
RA/IA, B15 = 1

Address Range:

0–4M-1

16-Bit Address

Conditions:
PSD 9 = 0
PSD 40 = 1
RA/IA, B15 = 0

Address Range:

0–64K-1

Real Extended Addressing

2-5

The flag field (bit 15) is called the Extension Selector. Several new fields have been added to the PSD. Two of these are the Extension Selector (PSD 15) and the Extension Address (PSD 42-47). If the Extension Selector is a zero, then the 16-bit displacement address points to a word within the first 64K of memory.

If the Extension Selector is a one, then the 16-bit displacement addresses a word in the 64K region that is identified by the contents of the Extension Address. In other words, when bit position 15 is equal to 1, a full memory address is formed by concatenating PSD 42-47 with bits 16-31 of the address field. This gives a 22-bit word address to allow the instructions to address all of memory.

Other Addresses and Displacements — Outside of reference address fields and the instruction address field of the PSD, all address and displacement fields are extended into adjacent previously undefined fields in order to address all of memory directly. The areas affected are as follows:

- Indirect address location contains either a full word address of 22 bits, or a 16-bit address which is linked to the extension address bits of the PSD (see figure on the following page)
- Index registers contain a full displacement of from 21 to 24 bits for doubleword, word, half-word, and byte references
- The stack pointer for PUSH-PULL instructions contains a full 22-bit word address
- The registers which describe the source and destination byte address for byte string instructions are 24-bit byte addresses

Branching and Branch Addresses — The Extension Address field of the PSD is modified automatically by branch instructions. If the effective address of a branch instruction (however generated) is outside of the first 64K of real memory, then the higher order six bits of this full effective address will automatically be loaded into the Extension Address field of the PSD if the branch is taken. The remaining part of the effective branch address will be loaded into position 16-31 of the PSD. In addition, bit 15 of the PSD (the Extension Selector) will be set to 1. If the effective address of the branch is within the first 64K, then the Extension Address is not modified, the Extension Selector is set to zero, and the effective address is placed in bits 16-31 of the PSD. This means that once the Extension Address has been set it will remain set until it is either changed

MODE 1

0 1	7 8	11 12	14 15	31
1	O	R	X	REFERENCE ADDRESS (INDIRECT)

	0	10	31
	0		DIRECT ADDRESS

22-BIT
REAL EXTENDED ADDRESS

MODE 2

0 1	7 8	11 12	14 15	31
1	O	R	X	REFERENCE ADDRESS (INDIRECT)

PSD

42	47
EXT. ADDRESS	

0	15 16	31
1	E S	DIRECT ADDRESS

BIT 15 = 1

BIT 15 = 0

10	31

22-BIT
REAL EXTENDED ADDRESS

16-BIT DIRECT
ADDRESS

2-7
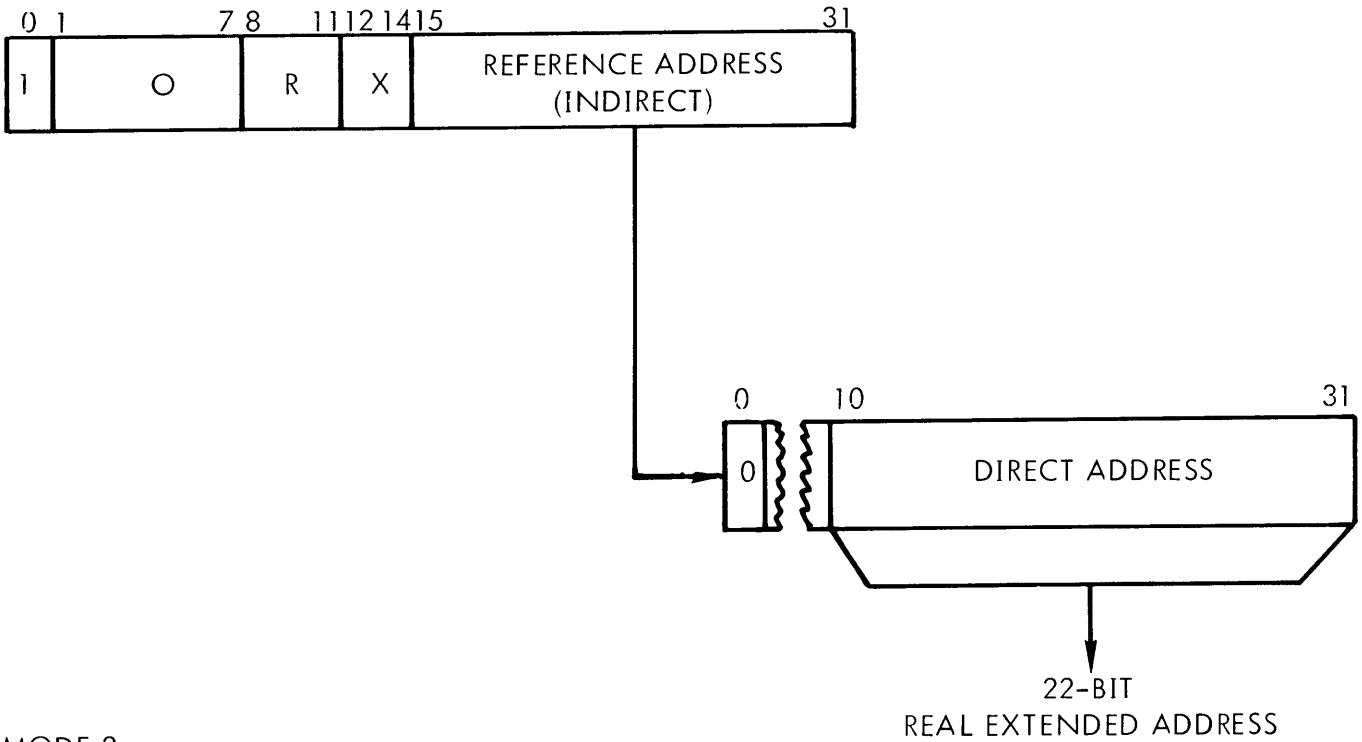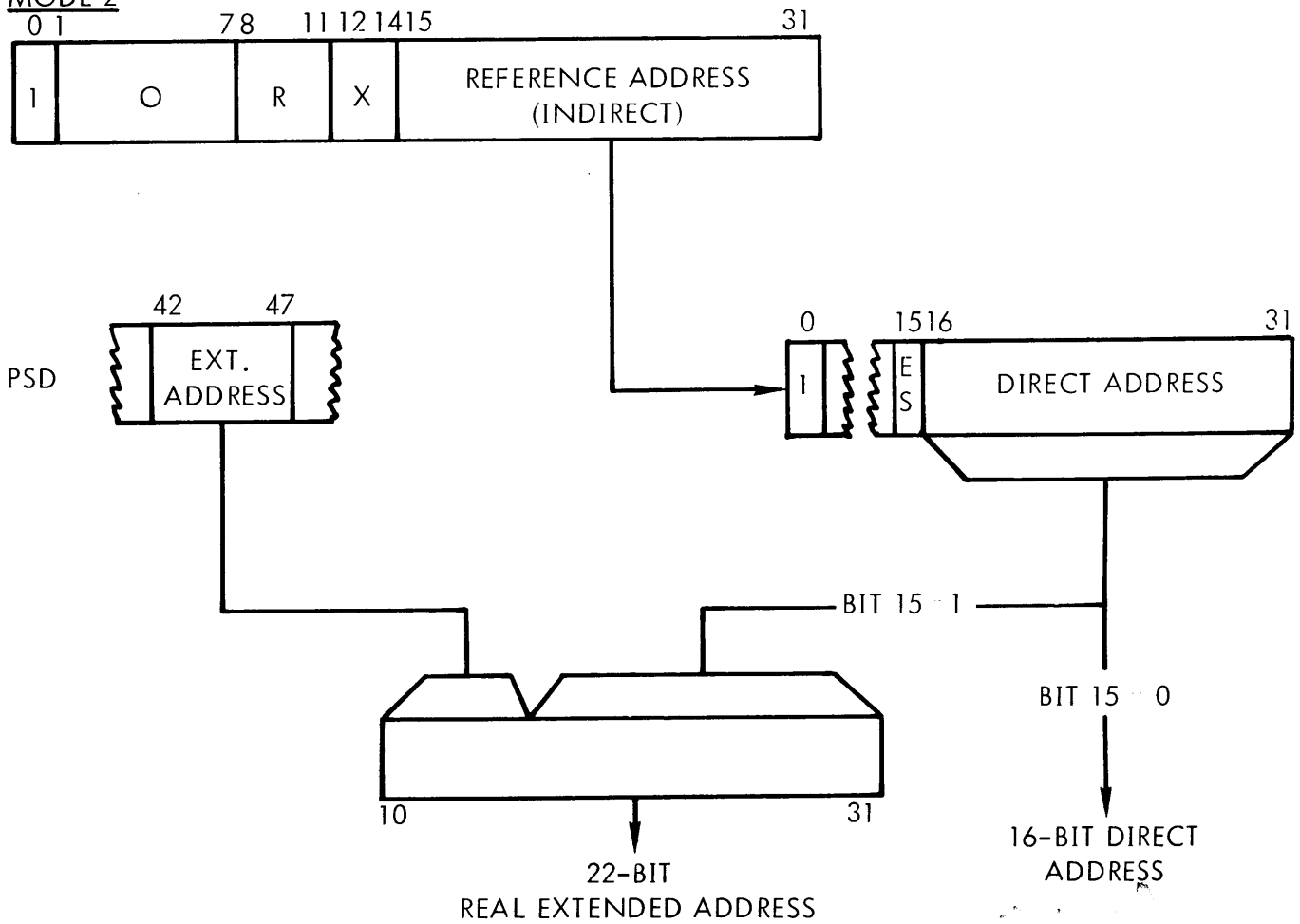
by loading a new PSD, or by the actual branching into another 64K region of memory.

A branch and link instruction in Real Extended Addressing will store the full address of the next instruction in the link register. If the Extension Selector in the PSD is zero at the time of the' branch and link, then the address stored in the link register will be the incremented 16-bit displacement from position 16-31 of the PSD. If the extension selector in the PSD is equal to one, then the address stored will be the incremented 16-bit displacement (PSD 16-31) plus the contents of the Extension Address (PSD 42-47) into bit positions 10-31 of the link register.

2.1.3 CPU FAST MEMORY

Sigma 9 contains a fast memory within the CPU. This section describes its general usage.

2.1.3.1 General Registers

The general register format for Sigma 9 is identical to Sigma 7. There are 16 registers in a block, seven of which may be used as index registers. The last four registers are the decimal accumulator used in decimal operations. The major difference between Sigma 7 and Sigma 9 registers is that Sigma 9 may have only four blocks of registers installed in the CPU. Two blocks are standard and two are optional.

2.1.3.2 Memory Map

The memory map on a Sigma 9 is held in fast memory within the CPU. The map storage is an array of 256 elements where each element is 13 bits in length. Only one map may be installed in a CPU.

2.1.3.3 Access Protection Storage

The access protection is a feature of the memory map. On Sigma 9, it is identical to the access protection of Sigma 7. The access protection requires 256 two-bit elements of fast memory. The access protection is loaded independently of the map storage in the CPU fast memory.

2.1.3.4 Write Lock Storage

The write lock storage consists of 256 two-bit elements in the CPU's fast memory. This is identical to the write lock storage on Sigma 7. It provides write protection for the first 256 pages or 128K words of real memory. There is no plan to provide write lock protection for real memory beyond 128K words.

2.1.4 INTERRUPT SYSTEM

This section describes the interrupt system of Sigma 9. The interrupt system is similar, but not identical, to that of Sigma 7. This section concentrates primarily on describing the difference from Sigma 7.

2.1.4.1 Organization of the Interrupt System

There are two changes in the interrupt organization of Sigma 9 which cause it to be different from Sigma 7. First, the memory parity interrupt of Sigma 7 (location x '56') has been replaced by a system fault interrupt. The system fault interrupt is used to notify the CPU in the system that some system element has detected an error in itself. The second change is the addition of a memory fault interrupt to location x '57'. The memory fault interrupt reports memory faults to the CPU. Uncorrectable read and write errors, data for checks, address bus checks, and port selection errors are the types of error conditions which cause the memory fault interrupt. At the time of interrupt, the snapshot feature in memory records the conditions existing in the failed memory for later examination by diagnostic programs.

2.1.4.2 Control of the Interrupt System

All control functions of Sigma 7 are retained for Sigma 9. In addition, one new Write Direct and three new Read Direct functions are added to Sigma 9. These new functions are designed to allow the programmer to interrogate the condition of the interrupt system at any time and to restore that system at a later time.

The new Write Direct function allows the programmer to set active all selected levels currently in the armed or waiting state. This new function allows interrupts to be returned to the active state. This was the only state to which direct control was not possible on Sigma 7.

Sigma 7 did not have any capability for reading the condition of an interrupt level. All the Read Direct functions for interrupt control are therefore new on Sigma 9. The three new functions allow the programmer to determine which interrupts are in the armed or waiting states, which are in the waiting or active states, and which interrupts are enabled.

2.1.4.3   Interrupt Entry Addressing

Because Sigma 9 can have a memory much larger than Sigma 7, the addresses of instructions used in interrupt locations have been extended from 17 to 20 bits. This has been done to ease the problem of where to place exchange tables and clock counters. For the XPSD instruction, this poses no problem since the high order three bits that make up the 20 bits were reserved on Sigma 7. This extension will, however, make indexing on modify and test instructions no longer possible, but it will allow any interrupt instruction to address one million words directly.

If the interrupt instruction makes an indirect reference, the indirect word may contain a 22-bit, four million word main memory address. This type of addressing is called objective addressing because it is not subject to any volatile state of the CPU.

2.1.5   TRAP SYSTEM

The trap system of Sigma 9 is similar to that on Sigma 7. There is one interrupt on Sigma 7 which is now a trap on Sigma 9. There are also some modifications of the Sigma 7 traps as they are used on Sigma 9. These modifications are described in the following subsections.

2.1.5.1   Trap Entry Procedure

Trap routines are entered by the execution of a Trap Instruction. An XPSD is the only Sigma 9 instruction intended for use as a trap instruction and has two options which are not operative at any other time.

One of these options involves the manner in which the exchange table of the XPSD is addressed. The table may be addressed either objectively or subjectively. If the addressing is subjective, the addressing is the same as the current type of addressing used by the program that was trapped. If objective addressing is used, the XPSD may address memory independently of the current type of addressing being used. The addressing is independent of any aids outside of the instruction itself or the memory. Bit 10 of the XPSD determines what type of addressing will be used. The state of bit 10 has no affect on the operation of an XPSD except when that XPSD is used in a trap location.

In the event of a trap entry, bit 61 of the PSD will be set if any general register or core memory location has been altered as a result of the execution or partial execution of the instruction which caused the trap.

2.1.5.2   Non-Allowed Operation Trap

The non-allowed operation trap has been changed from the manner in which it operated on Sigma 7. When operating in the Master Protected mode, a memory protection violation may occur if the access protection for a given page is violated.

Because the CPU is attempting to work on instructions and operands ahead of current execution, certain anomalies may occur. When most branch instructions branch, therefore, the CPU will (in many cases) try to access the instruction addressed by the branch as soon as possible. If the address generated results in a violation of protection or reference to nonexistent memory, the trap will not occur unless the branch conditions are actually met. In other words, if the instruction that follows a branch is a valid instruction generating no violation, then no trap occurs because the hardware attempted to anticipate the program's needs.

All instructions which reference a series of locations in the course of their execution that might cross a page boundary will test for trap conditions before commencing execution. If an access protection, memory protection, or nonexistent memory reference should occur during the course of the execution, the trap will now occur before execution begins.

When a non-allowed operation trap occurs as the result of a mapped operation that violates the memory protection feature, the virtual page address that caused the trap to occur will be stored in bits 48-55 of the doubleword pointed to by the effective address of the XPSD instruction. This solves the problem of determining where the violation occurred.

2.1.5.3 Memory Parity Trap

This is a new trap for Sigma 9. It represents a change from the memory parity interrupt on Sigma 7 to a trap on Sigma 9. The memory parity trap to location X'4C' will occur whenever a memory unit detects an error it cannot correct, as a result of a request from the CPU. Errors caused by requests from other processors will not cause this trap.

2.1.5.4 Instruction Exception Trap

The instruction exception trap is also a new feature on Sigma 9. Any new trap condition detected and allowed to cause a trap during the trap sequence (including execution of the XPSD instruction, or during the interrupt sequence, including execution of the XPSD, MTW, MTH, or MTB instruction), is defined as an instruction exception trap and causes a trap to location X'4D'. Many other errors will also cause the instruction exception trap; some of these are:

- An illegal instruction found in a trap or interrupt location

- Using an odd-numbered register with some of the doubleword and byte addressing instruction

- The register block pointer set to a non-existent register block

This trap clearly provides a solution to the problem arising from misuse of instructions.

2.1.6 SIGMA 9 INSTRUCTIONS

The Sigma 9 instructions are an extension of the Sigma 7 instruction set. This section describes the new instructions for Sigma 9 and also changes to Sigma 7 instructions.

2.1.6.1   New Instructions for Sigma 9

There are three new instructions for the Sigma 9; they are: Load Real Address (LRA); Load Memory Status (LMS); and Load and Set (LAS).

Load Real Address loads into a general register the real effective address of the operand pointed to by the instruction. The address loaded can be of a byte, halfword, word, or doubleword as determined by the setting of the condition codes at the beginning of execution. This means that the programmer can use LRA to obtain the real memory address of an operand under real, virtual, or real extended addressing.

The Load Memory Status instruction is used to interrogate the status of a memory bank. It can also be used to perform a diagnostic action on a memory bank. The effective address of the instruction is used to determine whic h memory bank is to be referenced. The setting of the condition codes just prior to execution determines what diagnostic action is to be performed. The LMS instruction could be used, for example, to retrive under program control the contents of the memory snapshot registers. This feature may be used by the operating system for error logging or by a diagnostic program to assist in fault locating.

The Load and Set instruction will not have an immediate use on the Sigma 9. Load and Set is used in a multiprocessor system to prevent a given CPU from accessing a memory location already being used by another CPU. Load and Set will, however, have a definite use for Sigma 9 enhancements.

2.1.6.2   Modified Instructions

The extended capabilities of Sigma 9 have required the modification of some of the Sigma 7 instructions. Most of the changes are concerned with the extended addressing capability of the Sigma 9. There are, however, totally new options available on some instructions.

Byte string instructions run five to six times faster on Sigma 9 when bytes are transferred on byte boundaries. Sigma 9 also has a speed advantage when bytes are transferred on word boundaries.

2-13

A new shift type instruction has been added. A number of applications require that a single bit marker be located within a word or doubleword filled with these markers. The new shift instruction allows the programmer to perform a searching shift on either a word or doubleword to determine the presence or absence of a particular bit or flag. The shift may be either left or right and will provide a count of the bits shifted while searching.

The floating point arithmetic on Sigma 9 is the same for Sigma 7 except for the following difference — the floating point is standard on Sigma 9 and is not an option. All floating point instructions will yield identical results with Sigma 7.

Sigma 9 decimal operation differs from Sigma 7 in that Sigma 9 decimal instructions are much faster than Sigma 7 (2 to 3 times faster). Also, Sigma 9 decimal instructions are capable of generating either EBCDIC or ASCII preferred sign and zone codes. This option allows the programmer to choose the mode of operation most convenient for him.

The Read Direct and Write Direct instructions have had several important optional forms added to their formats. The Read Direct instruction is now capable of reading the snapshot register within the CPU. The information held in the snapshot register can be used for diagnostic purposes to discover the nature of a fault in the CPU. The Read Direct instruction is now also capable of reading the state of the interrupt inhibits and sensing the various status of the individual interrupt levels within the CPU interrupt system. The ability to sense and record the status of individual interrupts becomes extremely useful when combined with a new feature of the Write Direct instruction. This new feature of the Write Direct gives the programmer the ability to set the state of the interrupts (all states including active). As an example, the programmer (using the Read Direct) may now sense and record the status of the interrupts and, at a later time (using the Write Direct) re-establish that same status in the interrupt system. The Write Direct instruction has also been modified to allow it to set the interrupt inhibits and to arm the snapshot feature. In arming the snapshot feature, the Write Direct supplies a clock counter and a condition select. The condition select describes what information is to be taken in the snapshot and the clock counter tells when it should be taken.

With one exception, the functions of the I/O instruction on Sigma 9 are identical with those on Sigma 7. The exception is a modification of the Halt I/O instruction that serves to generate a reset function for an entire IOP. This modification is called Reset I/O (RIO). RIO causes the selected IOP to generate an I/O reset signal to all devices attached to it. The use of several RIO instructions in sequence will cause the entire I/O system to be reset one IOP at a time. (Refer to Paragraph 2.3.1.4 for details of I/O instructions.)

2.2 CORE MEMORY

The core memory developed for Sigma 9 combines and extends the successful designs of the Sigma line memories. It offers significant improvements in the port structure to accommodate up to 12 processor interfaces for each concurrently operating memory unit, and with additional error detection hardware improves system viability. Sigma 9 core memory is large, flexible, fast, and easy to maintain.

2.2.1 BASIC MEMORY

A Sigma 9 basic memory (minimum allowable memory in a Sigma 9 system) is 128K (K = 1024 words) with two ports. This configuration is physically contained in four memory cabinets with 32K words in each cabinet. This minimum configuration allows two processors (CPU and IOP) to interleave between two 16K banks in modulo 2 fashion, or between four 16K banks in modulo 4 fashion. Both processors can access separate 16K banks simultaneously.

2.2.2 MEMORY SIZE OPTIONS

Core memory is expandable from 128K to 256K in 32K increments, and from 256K to 512K in 64K increments.

The maximum allowable core memory for Sigma 9 is 512K words, which physically occupy 16 memory cabinets with 32K in each cabinet. CPU's and IOP's in the Sigma 9 system have memory addressing capability of 4096K words; however, any requirements for memory sizes greater than 512K words must be handled on an RPQ basis.

2.2.3   PORT OPTIONS

Additional ports may be added to each memory unit as more processors are added to the Sigma 9 system. The two basic memory ports are numbered 1 and 2 with port 1 high and port 2 low in priority. The first port expansion to the basic memory configuration would be port 3, which would have a lower priority than ports 1 and 2. Subsequent port expansion may be made one at a time, up to a maximum of 12 ports. The lowest numbered port has the highest priority; the highest numbered port has the lowest priority. This hard-wired priority scheme may be selectively overridden by any processor connected to a port. If, for example, a processor on port 4 has a critical requirement to access a memory unit at the same time that the three other higher priority ports are trying to access the same memory unit, the processor on port 4 may send a "high priority request" signal to the memory unit that will allow him to override the hard-wired priority scheme and gain access before the other three ports.

Any requirement for port expansion beyond 9 ports must be handled on an RPQ basis.

2.2.4   MEMORY ORGANIZATION

A Sigma 9 memory system consists of a number (4 - 16) of independent memory units with each memory unit having two 16K banks. Each bank may be accessed independently of, and concurrently with any other bank in the memory system.

Each 32K memory unit may have from 2 to 12 ports connected to it and either bank within a memory unit may be accessed by any port (it is possible for one or two ports to be accessing both banks simultaneously).

2.2.5   INTERLEAVING

Interleaving is permissible between two 16K banks within a 32K memory unit (modulo 2) and between four 16K banks in two separate 32K memory units (modulo 4). Thus, interleave group sizes may be 32K or 64K. Each bank in an interleave group must be the same size and each

interleave group must begin on an address boundary equal to a multiple of the interleave group size.

2.2.6   MEMORY OPERATIONS

Sigma 9 core memory is capable of storing and transferring bytes, halfwords, and words in either an operational or diagnostic mode.

2.2.6.1   Operational Mode

The operational mode performs the normal read, full write, and partial write operations associated with storing and fetching data and instructions. Nominal access time for read and full write modes is 900 nanoseconds, and nominal access time for partial write mode is 1060 nanoseconds.

2.2.6.2   Diagnostic Mode

The diagnostic mode performs a variety of operations associated with diagnostic and multi-processor (CPU) configurations. The diagnostic mode is set by a CPU. Generally, the diagnostic mode allows a CPU to interrogate a memory bank to determine its environmental status. In this mode, the memory can transmit back to the CPU either normal data from core or status information from a 3-word snapshot register. The snapshot register contains status information such as:

- Memory fault type
- Port number
- Interleave mode
- Bank size
- Memory unit number
- Memory unit size

2.3   INPUT/OUTPUT SYSTEM

The input/output system for Sigma 9 represents major improvements over earlier Sigma I/O systems in throughput and flexibility. Two different types of input/output processors (IOP's) are available for use in a Sigma 9 system:

- Multiplexor IOP (MIOP) which controls up to 32 device controllers
- HIGH SPEED RAD IOP (HSRIOP) which controls up to four Model 7212 high speed RAD storage units

The minimum Sigma 9 system configuration includes one MIOP. As many as 10 additional MIOP's and HSRIOP's may be added to a monoprocessing system in any mix. In multiprocessing systems, the number of allowable IOP's in the system is reduced by the number of CPU's (maximum of four) added to the system. Since each processor (CPU or IOP) must have its own port to memory, a maximum of 12 processors can be configured in a Sigma 9 system.

In all systems, Homespace Bias is transmitted to the IOP from the controlling CPU during I/O instruction execution. The Homespace Bias is added to the address values of X'20' and X'21' for CPU/IOP communication via core memory during the execution of I/O instructions.

2.3.1 MULTIPLEXOR IOP (MIOP)

The MIOP controls up to 32 device controllers in a dual-channel mode of operation similar to the earlier configuration of an MIOP with a bus-sharing IOP (BSMIOP). The MIOP has both 1-byte (standard) and 4-byte (optional) interfaces. With the 4-byte option, the MIOP can operate in a "burst" mode which causes the MIOP and a device controller to exchange complete records of data during a single service connection. This is similar to the operation of the earlier Selector IOP (SIOP), thus maximizing the throughput of a single device during times when data overruns on other devices are not probable. Transfer rates of an MIOP are typically 470 kb/sec on a 1-byte interface and 940 kb/sec on the optional 4-byte interface.

A unique memory-to-memory move option (MMM) allows the MIOP to move a maximum of 16K words from one area in core memory to another. Unlike similar operations using CPU instructions such as move byte string (MBS), MMM is performed as an I/O operation which means the CPU does not have to be tied up performing block movements of core memory data.

2.3.1.1 Channels

The MIOP is functionally divided into two separate operating channels, Channel A and B (reference the following figure). Both channels can operate concurrently except for sharing the core memory bus (interface). I/O instructions provide for channel discrimination in their addressing scheme. Both channels are contained in a functional assembly called a Channel Control Unit (CCU), which provides the interface logic to core memory and the CPU's in the system. A third interface is provided for diagnostic uses and allows the IOP to be interrogated and controlled during preventive and corrective maintenance routines.
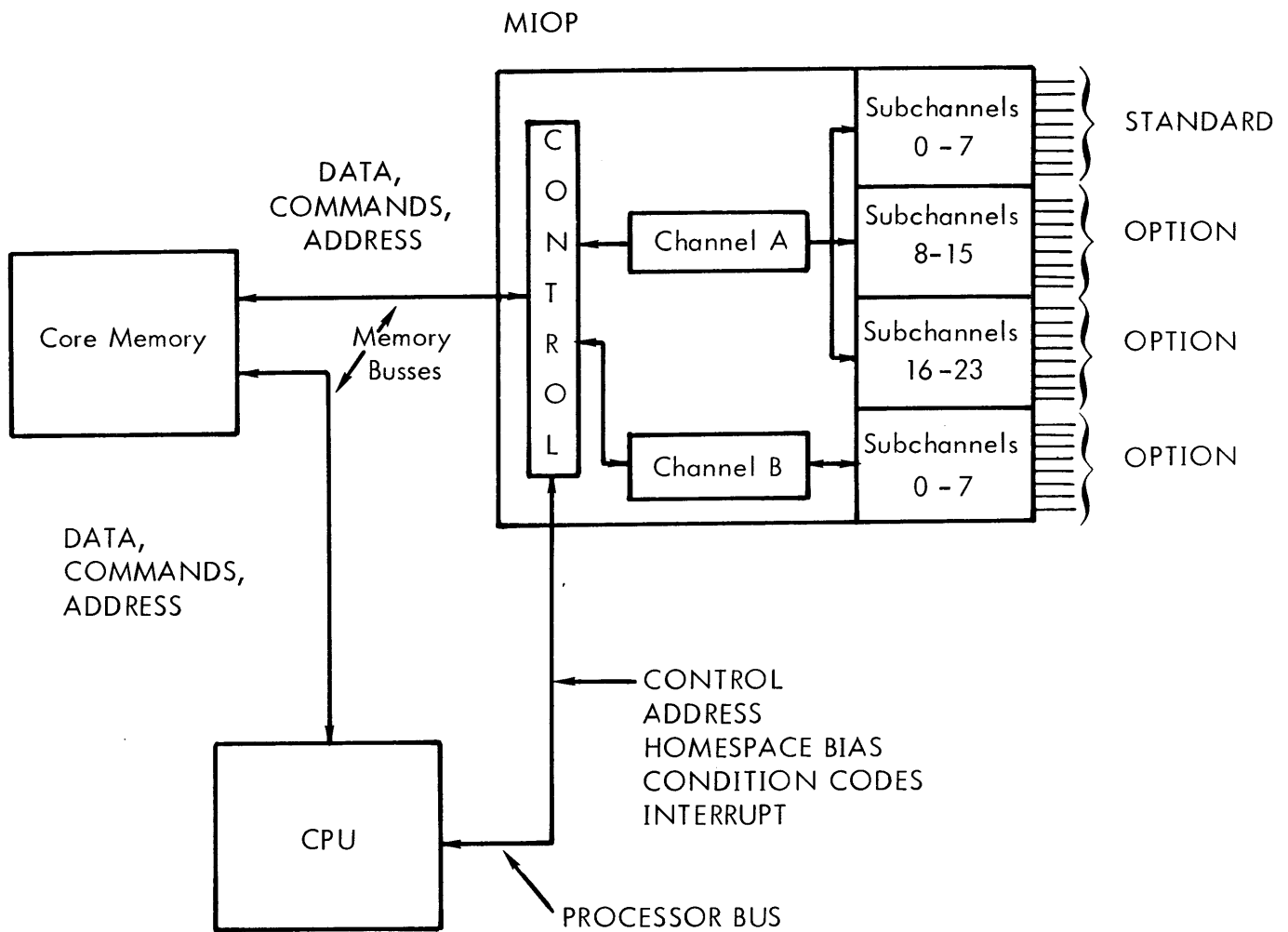
Multiplexor IOP — Functional Diagram

2-19

Channel A — Channel A is standard in the MIOP and comes with eight subchannels (numbered 0 – 7). Each subchannel controls one device controller. The basic set of eight subchannels can control both multi-unit device controllers (with up to 16 devices) and single unit device controllers. Subchannels can be added in groups of eight to increase the number of channel A subchannels to 8 – 15. The second increment adds subchannels 16 – 23. Subchannels numbered 8 – 23 can control only single unit device controllers. Subchannels 8 and 9 have special significance when the memory-to-memory move (MMM) option is installed.

Channel B — Channel B is an option which comes with eight subchannels. Channel B subchannels (numbered 0 – 7) can control both multi-unit device controllers (with up to 16 devices) and single unit device controllers. Thus, with both Channel A and Channel B installed, the MIOP can accommodate up to 32 device controllers. The ability of the Sigma 9 MIOP to control up to 16 multi-unit device controllers is an increase in flexibility over earlier IOP's where only eight multi-unit device controllers can be installed. Each subchannel has storage facilities (fast-access memory registers) in the MIOP to contain all the control parameters needed to operate a device controller. To improve accuracy and MIOP viability, parity generation/checking is performed on all subchannel storage registers.

2.3.1.2 I/O Interface

Both channels come standard with a 1-byte interface which allows each device controller to exchange a single byte of data every time it connects to the MIOP for service.

An optional 4-byte interface may be installed to each channel which then allows those device controllers that have similar interfaces to exchange up to four bytes of data every time it connects to the MIOP for service. Further, if the device controller is capable of exchanging more than four bytes during a service connection, it may initiate a "burst mode" operation and remain connected to the MIOP, exchanging data until zero byte count or channel end is encountered. The burst mode capability is available as part of the 4-byte option.

2.3.1.3 Memory-to-Memory Move (MMM)

A common requirement in efficient core memory management is the dynamic relocation of pro-
grams and data within core memory. In the past it has been incumbent upon the CPU to perform
this task of relocating blocks of data within core memory at the expense of increasing CPU over-
head time and sacrificing valuable program execution time. With the MMM option, this task is
transferred to the MIOP and is performed as a simulated I/O operation.

To accommodate this option, the MIOP must have the Channel A expansion option which provides
subchannels 8 – 15. Subchannels 8 and 9 are dedicated to MMM when it is installed. The MMM
option interfaces to subchannels 8 and 9 as I/O device controller simulators. Subchannel (device
controller) 8 is used to specify the area in memory to which the data is to be moved (destination)
and acts as an input device. Subchannel (device controller) 9 is used to specify the area in mem-
ory from which the data is to be moved (source), and acts as an output device. The MMM is mech-
anized within the MIOP and only transfers data on a word basis. Because all data transfers are ac-
complished internal to the MIOP, the 4-byte interface option is not required. The MMM is con-
trolled by I/O instructions addressed to Channel A, device controllers 8 and 9. The MMM trans-
fers data at approximately 143K words per second (572 kb/sec).

2.3.1.4 I/O Instructions

The I/O instructions available for the Sigma 9 I/O system are as follows:

- SIO – Start I/O
- TIO – Test I/O
- TDV – Test Device
- AIO – Acknowledge I/O
- HIO – Halt I/O
- RIO – Reset I/O
- POLP – Poll Processor
- POLR – Poll Processor and Reset

The instructions RIO, POLP, and POLR are new and consist of variations of the HIO instruction. The SIO, TIO, TDV, AIO, and HIO instructions operate as they did in Sigma 5/6/7 systems. To accommodate additional IOP's and to take advantage of the bus sharing feature of Channels A and B, the format of I/O instructions has been modified as shown in the following figure. The AIO instruction format is essentially unchanged. The POLP and POLR instructions cause the addressed IOP to return fault status to a CPU general register and are usually executed as a result of a Processor Fault Interrupt (PFI).

	15	16	17
HIO	0	0	0
RIO	0	0	1
POLP	0	1	0
POLR	0	1	1

Operation Code (O) = X'4F'

	19	20	21	22	23	
IOP 0, Channel 0	0	0	0	0	0	(X'00')
IOP 0, Channel 1	0	0	0	0	1	(X'01')
IOP 1, Channel 0	0	0	0	1	0	(X'02')
IOP 13, Channel 1	1	1	0	1	1	(X'1B')
Reserved for future use	1	1	1	0	0	(X'1C')
	1	1	1	1	1	(X'1F')

Multi-unit Device Controller Addressing (Except for POLP and POLR)

Single-unit Device Controller Addressing (Except for POLP and POLR)
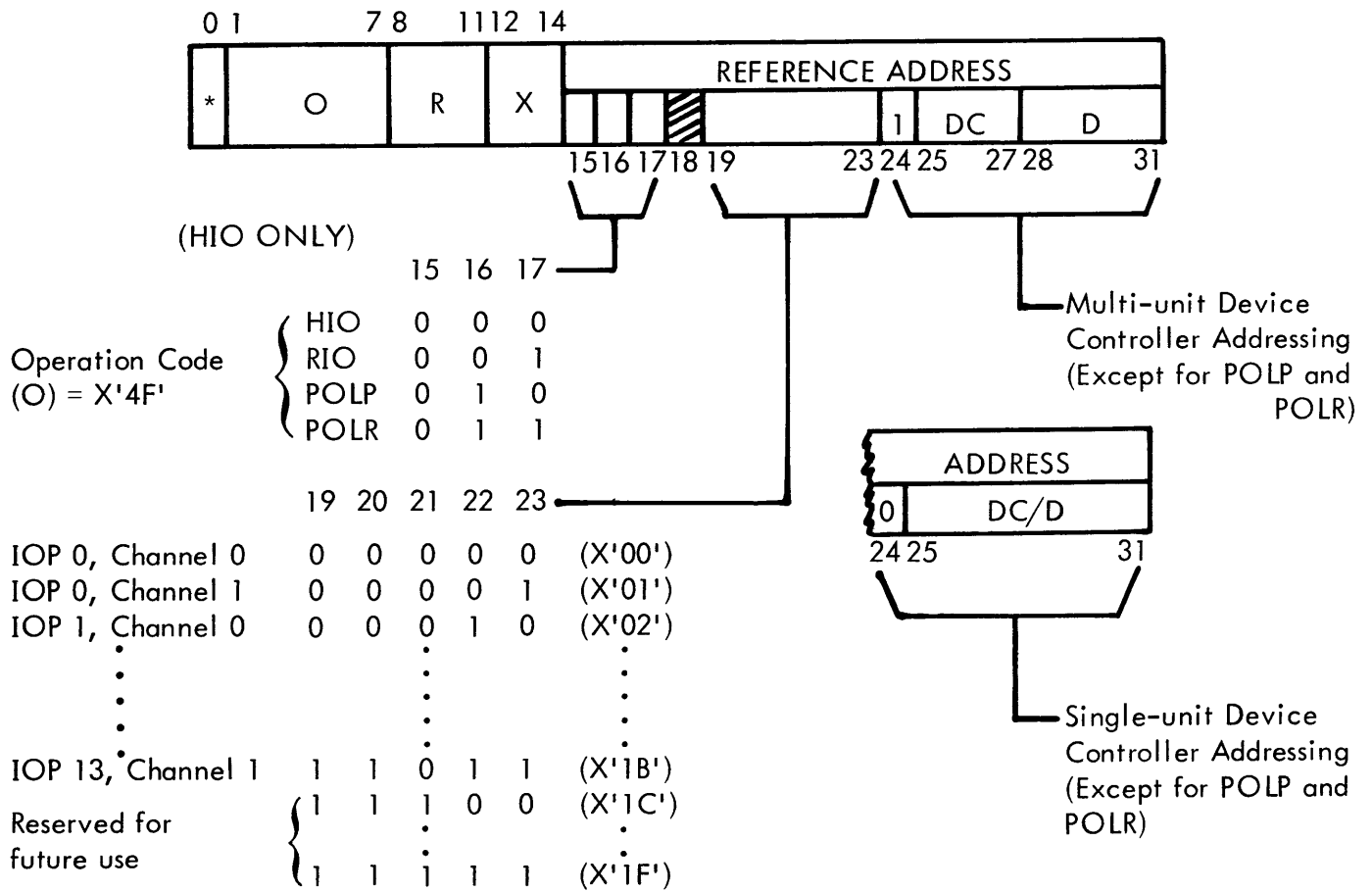
SIO, TIO, HIO, TDV Instruction Format

2.3.1.5   Maintenance Interface

The Sigma 9 MIOP contains a maintenance interface which allows the MIOP to respond to Read Direct (RD) and Write Direct (WD) instructions executed by the CPU. The primary function of the maintenance interface is to enhance the maintainability of the MIOP by allowing the CPU to interrogate IOP internal control elements (via a snapshot register) and to single phase an IOP to analyze details of an I/O operation for diagnostic purposes.

2.3.1.6   Alternate Processor Bus

The Alternate Processor Bus option is a group of modules which provide an additional bus to connect IOP's and CPU's in the Sigma 9 system and enables IOP's to be partitioned in a multiprocessor system for diagnostic purposes.

2.3.2   HIGH SPEED RAD IOP (HSRIOP)

The HSRIOP is a combination of an IOP and a device controller. Specifically, the field-proven designs of the Sigma 5/6/7 Selector IOP (SIOP) and the Model 7211 High Speed RAD controller have been integrated into a single unit now called the HSRIOP. The HSRIOP is solely dedicated for use with the existing Model 7212 High Speed RAD Storage Units. The HSRIOP may have as many as four Model 7212's connected to it with transfer rates up to 3MB per second. Each storage unit has a capacity of approximately 5.4MB.

2.3.2.1   Options

The only HSRIOP option is the Alternate Processor Bus described in Paragraph 2.3.1.6.

2.3.2.2   Maintenance Interface

The HSRIOP contains the same Maintenance Interface described in Paragraph 2.3.1.5.

2.3.2.3   I/O Instructions

The HSRIOP responds to the same I/O instructions described in Paragraph 2.3.1.4 except that it is not sensitive to channel addresses (bit 23 in the I/O instruction format).

2.3.2.4   I/O Operation

The HSRIOP and Model 7212 High Speed RAD Storage Unit always exchange data in the burst mode, and only one storage unit can be exchanging data with the HSRIOP at any given time.

2.4   CONFIGURATIONS

As mentioned earlier, user programs currently running on a Sigma 5, 6, or 7 are fully compatible with a Sigma 9 of equivalent or larger configuration. The RBM, BPM, and BTM operating systems will be modified so that they will also be compatible with Sigma 9. There will be, however, certain minor differences in operating systems. These differences will be totally transparent to the users and all discrepancies between hardware systems will be resolved by XDS's programming department.

2.5   UTS CONFIGURATION

The recommended minimum UTS configuration for a Sigma 9 system (shown on the following page) is conceptually identical to the recommended minimum UTS configuration for the Sigma 7. Since the equipment list for Sigma 9 is different, there are some hardware differences. Most importantly, only one port to memory is required to interface a memory bus to a 32K block of memory, while each separate 16K bank can be simultaneously active. The effect of this is to reduce the number of ports required. Actual operation is as in the past.

2.6   TRANSITION AND UPGRADE CONFIGURATIONS

Existing installations present the possibility of upward expansion to a Sigma 9 system. These installations pose no difficulty if they are currently operating under UTS. If another operating system is involved, the transition to Sigma 9 UTS will require examination, but will be smooth due to the compatibilities designed into UTS.

SIGMA 9 UNIVERSAL TIME SHARING SYSTEM

RECOMMENDED MINIMUM HARDWARE REQUIREMENTS
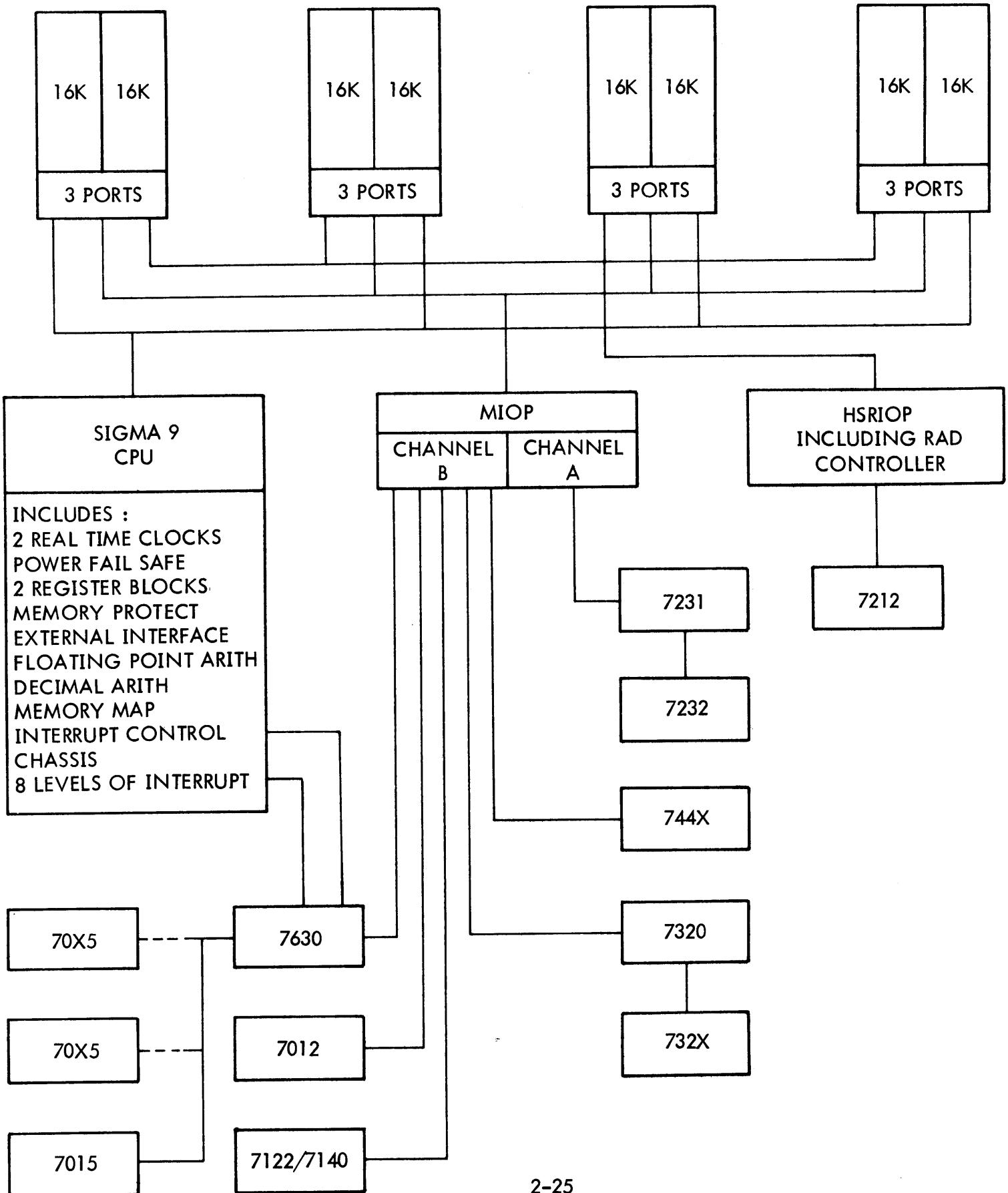
2.6.1 RBM USERS

RBM will be modified to run on the Sigma 9. Sigma 9, however, is not designed for a user whose primary applications are Real Time and, since UTS provides comprehensive real time services along with time-sharing and batch processing, an upgrade to Sigma 9 should only be considered if the customer has an application which can use all of the power available in a Sigma 9 UTS system.

2.6.2 BPM USERS

BPM will also be modified to run on Sigma 9. However, in a batch only environment or in batch and a small real time environment, Sigma 9 and XOS is the best system. If the customer's needs are growing, he should be considering the flexibility and adaptability of a universal system — Sigma 9 with XOS.

2.6.3 BTM USERS

While BTM will be modified to run on Sigma 9, it is a considerably less powerful operating system, than UTS. UTS is more efficient in swapping, resource use, and throughput. BTM users should be most anxious to upgrade to UTS, especially if they have a large number of time-sharing users or any real time applications.

Although RBM, BPM, and BTM will be modified to run on the Sigma 9, they will not support any hardware that is unique to Sigma 9 (i.e., they will support a maximum of 128K words, etc.). Using Sigma 9 with any of the systems probably would not take advantage of the features that Sigma 9 offers and a better solution would be UTS or XOS. UTS or XOS are designed to take maximum advantage of Sigma 9 and will give the customer the maximum performance for his investment.

8610E

Σ 9

MIOP Channel A

8671 Ext. Width Interface

8675 Channel B

7240 C. U.

7241

7246

7231 RAD C. U.

7235

7232

7012

7140

7160

7446

7322

7322

7320 C. U.
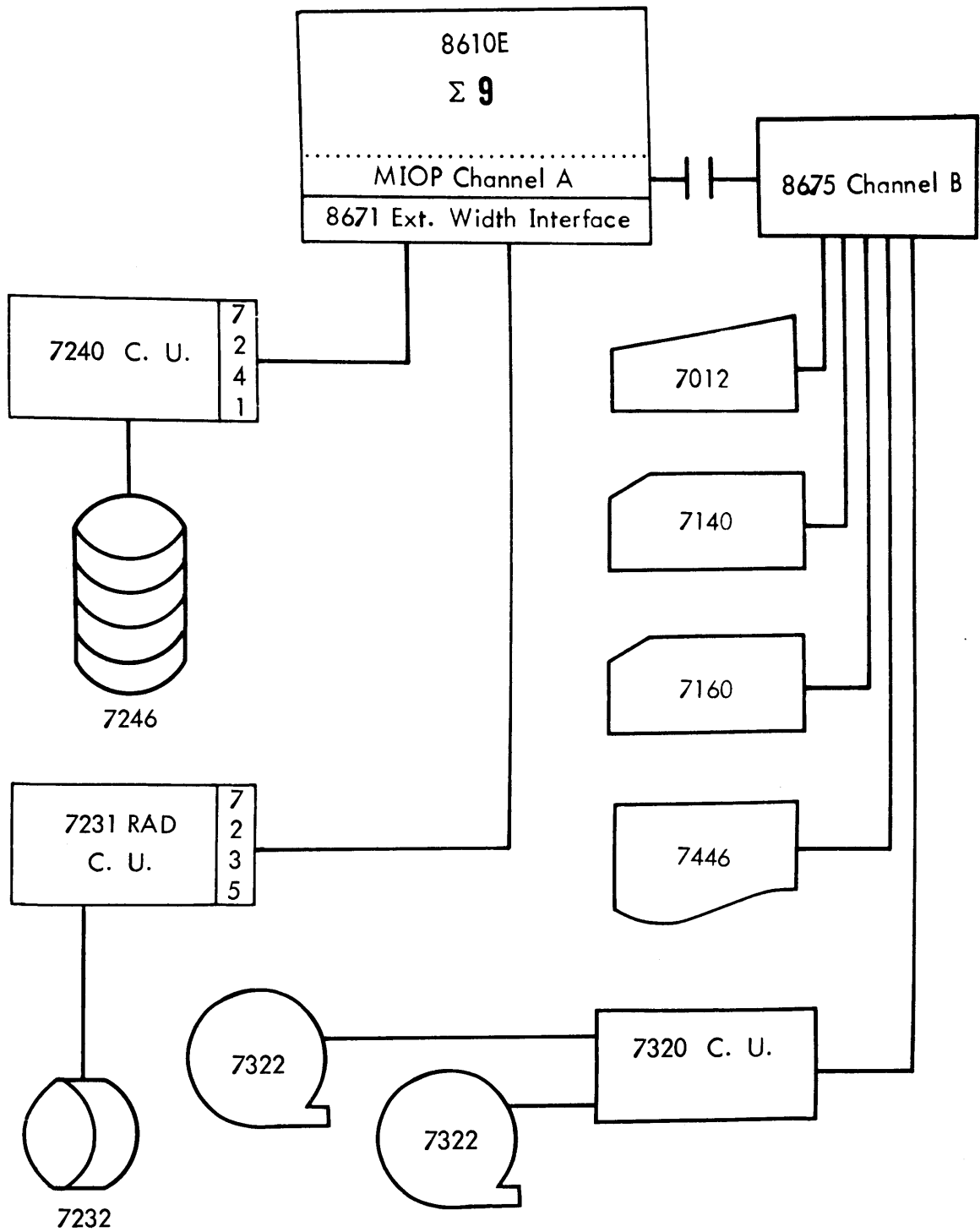
XOS MINIMUM CONFIGURATION

EQUIPMENT LIST

Quantity	Model	Description
1	8610E	Sigma 9:
		512K bytes Core Storage
		Decimal Arithmetic Unit
		Floating Point Arithmetic Unit
		Memory Map
		Write Protection
		Two Register Blocks
		Two Real Time Clocks
		Power Fail-Safe
		External Interface
		Interrupt Control Chassis
		Eight Interrupt Levels
		Multiplexor I/O Processor
		Channel A
		Eight Subchannels
		Motor Generator Set
1	8671	Extended Width Interface
1	8675	MIOP Channel B w/8 subchannels
1	7012	Keyboard Printer
1	7140	Card Reader (1500 CPM)
1	7160	Card Punch (300 CPM)
1	7231	Extended Performance RAD Controller
1	7232	RAD Storage Unit
1	7235	Extended Width Interface/7231
1	7240	Removable Disk Controller
1	7241	Extended Width Interface/7240
1	7244	Disk Pack (11-High)
1	7246	Single Spindle Disk Storage Unit
1	7320	Tape Controller
2	7322	Tape Unit (9-Track)
1	7446	Line Printer (1500 LPM)

SECTION III
OPERATING SYSTEMS

3.0 SIGMA 9 OPERATING SYSTEMS

The two primary operating systems available for the Sigma 9 are UTS and XOS. This section details the operational characteristics of each.

3.1 UNIVERSAL TIME-SHARING SYSTEM (UTS)

3.1.1 INTRODUCTION

The Universal Time-Sharing System (UTS), culminating years of XDS experience in the time-sharing environment, is a field-proven operating system offering users the full spectrum of concurrent operating modes: interactive time-sharing, local and remote batch, terminal batch entry, and real time. A wide selection of compatible interactive and batch mode subsystems lets the user apply the system easily and selectively to a variety of computing tasks. Further, comprehensive monitoring and reporting facilities allow installation management personnel to "tune" the system's capabilities, through dynamic allocation of its resources, to more closely fit user requirements, and to gather extensive accounting information on system activities.

The UTS software system is designed to exploit the potential of the Sigma hardware. It is the most advanced operating system available today . . . and is easily field-expandable to keep pace with future growth.

3.1.2 OPERATION FEATURES

3.1.2.1 UTS System

UTS has the unique ability to handle on-line interactive time-sharing, batch, and real time operations concurrently . . . UTS incorporates advanced features that assure efficient batch processing

concurrently with on-line conversational time-sharing. Real time operations typically are performed along with other modes of operation. Compatibility of batch and on-line programs and files allow for flexible use of the desired mode. UTS can efficiently support various combinations of these operating modes by appropriate choice of hardware and suitable setting of resource-allocation parameters.

- **128 on-line time-sharing users may use the system concurrently.** The UTS system allows users to initiate and access the on-line services via a variety of character-oriented terminal devices (teletypes, keyboard display units, etc.), by means of the character-oriented communications (COC) system. With the rapid access afforded by UTS, each user appears to have a dedicated system at his disposal. Users are stored on the high speed Model 7212 RAD for rapid access to core memory during execution phases. Full and half duplex terminals can utilize switched or dedicated lines, and remote multiplexing of terminals is offered for economy of communications utilization.

- **Batch operations may be handled either locally or remotely (from remote batch stations or time-sharing terminals).** UTS permits users to initiate and access batch processing services via three alternate operating modes: (1) local batch, through the computer operator at the CPU site; (2) remote batch, using high speed Model 7670 Remote Batch Terminals at any remote site; and (3) terminal batch entry, utilizing a service subsystem that provides access to batch job stream from any on-line terminal.

- **Real time operations may be made core-resident if required.** UTS lets users initiate and access real time processing services via the high speed interrupt system. The system processes both resident and non-resident real time activities from a variety of external data sources. Core residency insures rapid response to the most crucial real time environments. Applications of this type are given the highest priority and all facilities of the system are made available for fast response to these stimuli.

- **All on-line processor subsystems are reentrant and are shared by all users, maximizing usable core space.** Even though multiple users are compiling under a given processor (e.g., FORTRAN), only a single copy of that processor is resident in core memory. Through the use of reentrant processors, each user appears to have this processor dedicated to himself. The

memory space which is freed by not having multiple copies of the same processor resident allows many more users to reside in core (with rapid execution) than would otherwise be possible. Installation management may add shared processors to the system at System Generation time.

- Installation management can dynamically "tune" the system during operation to adjust for varying conditions of usage. Installation management personnel can quickly and easily alternate operations between full batch usage and concurrent batch/on-line usage, through use of the UTS terminal startup/shutdown facility. Other parameters allow for adjusting the relative amounts of resources allocated to batch operations versus on-line operations, as well as for sharing resources among users. By analyzing system utilization, installation management can use these parameters to allocate resources to the areas where they are most needed in any desired proportions.

- The accounting system automatically keeps track of each user's computer time. Accounting and billing are automatically provided by the UTS system which keeps track of all system resources utilized by each user. Service bureau management simply enter the appropriate charge rates and prepare billings directly from system reports. Management of an in-house installation can accurately depict utilization by department and/or user. The level of detail provided by UTS integrated system monitoring and reporting are virtually unavailable elsewhere. Further features include:

  - Complete diagnostic and recoverability capabilities to cover a wide variety of hardware and software contingencies
  - Hardware exercised routines to predict potential failures
  - New copies of software brought into the system if existing software malfunctions
  - Partitioning, where various hardware components are automatically removed from the system if a failure is detected, and alternate devices or paths are substituted
  - Detailed reports on all malfunctions for analysis, to determine suitable installation action

3.1.2.2 The Sigma Hardware

- **The memory map makes effective use of the core memory.** UTS memory mapping allocates core memory on a non-contiguous basis. Thus, users are fragmented in core memory by page, and are never denied access to core if 512 contiguous words out of 512,000 are available. This highly efficient allocation of memory increases the probability of usable core space many-fold over techniques used in competitive systems. Lengthy procedures associated with core shuffling, compacting, and relocation are totally avoided.

- **Rapid Access Data (RAD) storage systems provide high speed swapping of programs and data as well as storage for the operating system.** The Model 7212 High Speed RAD provides swapping users with rapid access to core memory for imminent execution. Slower swapping devices typically cannot keep pace with high speed central processing units and, thus, cause higher percentages of CPU idle time.

A multiple memory port structure allows CPU's and input/output devices to access the memory simultaneously. Core memory is expandable in 32K word increments from 128K to 256K words and in 64K word increments from 256K words to 512K words (2MB). Different banks can be simultaneously accessed by multiple system components; i.e., CPU, I/O processor, thus allowing many operations to occur simultaneously.

3.1.3 SYSTEM DESCRIPTION

The Universal Time-Sharing System is a compatible superset of the Batch Processing Monitor (BPM). As such, it includes all of the capabilities of BPM, providing complete compatibility with batch processing operations and services. The job-stack scheduling of BPM is retained, but is made a part of an overall scheduling logic that integrates batch operations with efficient scheduling of on-line time-sharing operations. Control parameters permit installation management to allocate system resources to any combination of on-line, batch, or critical real time usage.

All services and processors available under BPM are retained in UTS; they include:

- Comprehensive file management service
- Symbiont processing

- Input/output handling and queuing
- Checkpoint restart
- Overlay Loader
- Debugging, snapshot, and dump service
- Job stack priority scheduling
- Job accounting
- Clock service
- Error control
- Automatic volume recognition
- Remote batch processing
- Real time foreground privileged service

Additional services and functions provided by UTS to accommodate the demanding requirements of on-line time-sharing include:

- Communications Processing for conversational terminals
- On-line user scheduling
- Batch scheduling control
- User swapping algorithm
- Dynamic core memory allocation and map handling
- Usage accounting
- Save and continue on-line user execution
- System error detection and automatic restart
- Operational activity monitoring
- System management control

The configuration diagram on the following page shows a UTS system of typical size. It may be equipped with a maximum of 512 words of core memory and fifteen or more remote batch stations. RAD storage is required for symbiont operations, system utilization, and user file storage. Additional RAD units may be needed for particularly heavy file storage requirements; or, as a lower cost alternative for large-volume storage, removable disk storage units or magnetic card memory may be used.

LOCAL BATCH I/O OPERATIONS

- Business/Administrative

- Data Processing

- Scientific Data Reduction

ON-LINE TERMINAL USERS

1

Engineering development (on-line problem solving)

5

Immediate access to stored information base

12

Program development and debugging

57

Administrative work on-line, followed by terminal batch entry

128

Design engineering (user has just entered a terminal batch job, and has switched to a circuit design problem in the on-line mode)

BATCH JOB STACK

- Central Site Job
- Terminal 125
- Remote Station 1
- Central Site Job
- Central Site Job
- Central Site Job
- Terminal 57
- Central Site Job
- Terminal 13
- Central Site Job
- Remote Station 2
- Central Site Job
- Central Site Job

SIGMA 9 COMPUTER
WITH
128K CORE MEMORY

MODEL 7232 RAD

Symbionts,
Permanent programs,
and Files

MODEL 7212 RAD

Monitor,
Shared Processors, and
Current users
(swapping)

MODEL 7670
REMOTE BATCH STA = 1
BATCH JOB STACK

Scientific data reduction from a remote research laboratory

MODEL 7670
REMOTE BATCH STA = 2
BATCH JOB STACK

Business data processing from a remote manufacturing facility

REAL TIME USER

Circuit design from a Model 7580 Graphic Display terminal

REAL TIME USER

Monitoring laboratory experiments

The number of swapping RAD units required is a function of the number of on-line users as well as of the average user size (i.e., amount of core memory required per user).

3.1.3.1   Local Batch Processing

For those tasks entered in the local batch processing mode (local card reader), all of the features and functions of the batch processing monitor are available. These services include full use of symbionts, file management, batch processors (language and service), and I/O handling routines.

All tasks entered in the batch processing mode form a single batch job queue within a priority ordering. Tasks are then processed one at a time in priority order.

3.1.3.2   Remote Batch Processing

Remote batch processing is provided via the Model 7670 Remote Batch Terminal, which consists of a card reader/punch and a line printer. Jobs submitted at these terminals may use all of the features of BPM, except that binary coded input/output cannot be sent from/to the Model 7670 terminal. Jobs entered at a remote batch terminal are brought into the same batch job queue as local batch jobs.

Various commands are available for controlling and displaying the status of remotely entered jobs. These jobs may be controlled either from the central site or the remote site. Output need not be directed to the same remote batch terminal that submitted the job. If the terminal is functioning in the unattended mode, output will be sent when ready; otherwise, output will be saved until requested.

Memory requirements vary with the number of terminals used in the system. Assuming that the system is symbiont, then the additional memory requirements are for the handler and symbiont space for each terminal. The figure on the following page, Number of Remote Batch Terminals, shows this relationship.

WORDS OF CORE MEMORY VERSUS NUMBER OF REMOTE BATCH TERMINALS
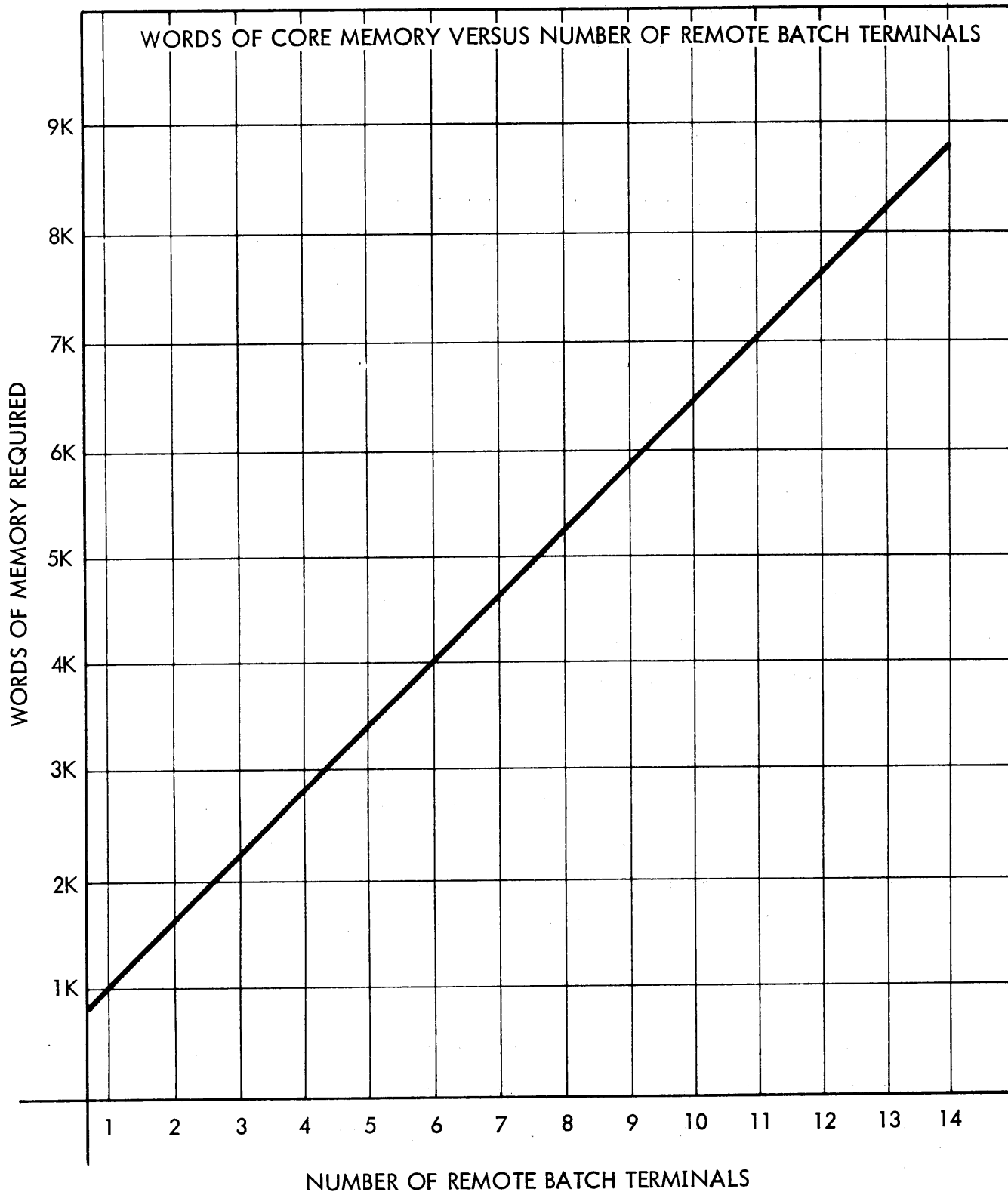
3.1.3.3 Terminal Batch Entry

The on-line terminal user has a choice of two processing modes — conversational or deferred batch. In those cases where the user does not wish to sit at the terminal attending the execution of a long process, he may conveniently employ the terminal batch entry facility of UTS to enter the job into the batch job stack and then disconnect from the system (or start an on-line task). This service allows the user to create and edit control command files to direct the execution of his job. At any time he may use the service to determine the status of his job. If it is still in the job stack, its current position, relative to other jobs, will be given. The job may be given a standard batch priority number, but only within the legal range assigned to that user by the system manager. If the on-line user decides his unexecuted job should not be run, it may be cancelled while waiting in the job stack. Even if batch processing is not being performed concurrently with on-line time-sharing, jobs may be entered into the job stack via the Terminal Batch Entry feature for subsequent execution as soon as batch processing service is activated.

A typical sequence of commands and responses for performing terminal batch entry appear as follows:

						COMMENTARY
!	Batch	Fid				enter job
!	Job	Jid	Submitted	2-18-69	2:30 p.m.	job accepted
!	Job	Jid				status requested
!	Status	Jid	Complete			status given

3.1.3.4 Critical Real Time Tasks

UTS will accommodate real time tasks that require privileged services and recognition of hardware priority. The services provided for such tasks are the same as those available under the batch processing monitor. Both core-resident and non-resident foreground tasks are established at System Generation time, with certain system resources (e.g., peripheral devices, external hardware interrupts, disk storage, and core memory) reserved for real time use as needed. Once established in the system, real time tasks may then be initiated by operator control commands and

activated by hardware interrupts. If the real time task is not resident, the monitor will transfer it into core memory on a high priority basis for immediate execution. A number of trap locations are reserved for installing master mode functions or I/O handlers required for special real time needs.

The following privileged services are available for real time operations:

- Trigger interrupt
- Connect trap location to specified address
- Disable/enable
- Arm/Disarm interrupt
- Set program status to Master/Slave mode
- Terminate real time task and disarm associated interrupts
- Control types of exit conditions
- Connect trap location to monitor abort routine

Real time operations may be carried out concurrently with batch and/or on-line service. However, performance and system loading must be carefully planned for in all cases.

3.1.3.5   On-Line Terminal Processing

UTS will handle 128 or more remotely connected on-line terminal users. A terminal executive language and processor (TEL) is available for processing requests from users. Most commonplace activities associated with FORTRAN and assembly-language programming can be carried out directly in TEL, while more involved activities require the use of associated processors.

The on-line user has available all of the tools required for creation, modification, debugging, and use of programs, as well as for creation, modification, examination, and deletion of files. File security and program security are afforded by means of the password concept. System and individual account status may be queried through the SUPER processor. File management and I/O are transparent to the user and are initiated through a variety of monitor calls.

3.1.4   SYSTEM MANAGEMENT

UTS system management routines perform all of the functions associated with the running of programs and the providing of required system resources. The major functions of UTS system management are described below.

3.1.4.1   User State Queues

Each user in UTS is associated with a state describing his current activity. Examples of states include LOGGING ON, COMPUTING, and I/O IN PROGRESS. Each user has one (and only one) entry in a state queue which is a list of users in the same state. Twenty-eight such queues are maintained by the scheduler. Three subsets of these 28 queues are used to select users for: (1) swapping in, (2) execution, and (3) swapping out. Results of this structure include:

- A flexible manner of establishing user priorities
- Intelligent selection of users to swap in and out

The queues are also used for entry and use of processors, I/O activities, and waiting for "wake-up" at a preset time.

3.1.4.2   Schedule and Swapping Routines

The UTS scheduler routine performs two major system functions: (1) selection and organization for users to be swapped in and out, and (2) selection of users for execution. The scheduler keeps track of the state of each user as a function of the events reported to it and, based upon this, makes decisions concerning the initiation of swapping and execution at these event times.

Typically, highest priority users residing in core memory are executed first; then, highest priority users residing on the swapping RAD are swapped in, with lowest priority users residing in core memory being swapped out to make room for them.

The swapping RAD acts as a high capacity, low cost extension to core memory and is used to store the overflow of users that cannot fit into core memory. Swapping out is possible for those users

residing in core memory for whom execution is not imminent. Swapping in is required for those users residing on the RAD for whom execution is imminent. Execution can take place only if the user currently resides in core memory. Swapping in can occur only if there is enough unused core space available to fit the user, or if lower priority users can be swapped out to make room for him. Only one user is swapped in at a time, although several users may be simultaneously swapped out.

All programs are organized into two areas: The procedure (program instructions) area, and the data area. Typically, the procedure area is not modified in the use of a program. During a swap in, the procedure and data are read into core memory. The associated processor is also read in if it does not already reside in core memory. A copy of the procedure and data remain on the swapping RAD after the swap in is completed. The procedure is swapped out only if it has been modified (e.g., by debugging). The associated processor is never swapped out. Thus, processors frequently remain in core memory even though they are not in use. In the event that a processor is not in use and more room in core memory is required, the processor is written over.

3.1.4.3 Memory Management

Virtual memory will consist of 128K words (512K bytes) although physical memory may be larger. The virtual memory layout within which programmers will structure their programs is shown below in the chart.

Monitor Area		User Area				Special Area
Data	Program	Context	Data	Program	Symbol Table	

- Monitor Area contains the monitor and its associated tables. The user is denied all access to this area

- User Area contains: (1) context, consisting of a job information table (JIT), data control blocks (DCB's), and buffers (file buffers and coop buffers); (2) Data (some pages are assigned directly to the program while others are assigned dynamically as needed); (3) program (procedure); and (4) Symbol table

- Special Area contains special shared processors (TEL, LOG-ON, DELTA, FDP, and public libraries)

Various buffers are required for the movement of information to and from core memory. These include:

- Terminal I/O buffers which reside in the monitor area
- File I/O buffers which reside in the user's context area
- Symbiont (input) buffers which reside in the monitor area
- Coop (symbiont output) buffers, which reside in the user's context area

The recommended core configuration will typically accommodate one batch user and three or four on-line users in core memory simultaneously. In the initial version of UTS, only one batch job will be processed at a time. Once a batch job is brought into core memory for execution, there is virtually nothing to distinguish its treatment from that of an on-line job. In particular, the batch job is eligible for swapping along with the on-line jobs. It is possible to modify this treatment of batch jobs by setting the appropriate system parameters (explained in a later section).

The memory map is used to dynamically fragment and relocate programs, data, and processors on a page basis. A given program will be broken up into its constituent pages which will be non-contiguously located in memory (physical memory). However, during the running of the program, the memory map translates all address references so that the program runs as if the pages were contiguous (virtual memory).

This translation process is accomplished by a series of map translation registers — one for each page of memory. Each register has an access protection code which defines whether a given page can be written into, read from, or executed by a given user. The map translation registers and access protection code registers are changed whenever a new program is allocated to CPU.

The access codes are:

- None – no access of any kind permitted
- Read – read access only
- Execute – execute or read access
- Write – write, execute, and read permitted

During user execution, the following access protection is in effect:

AREA			ACCESS CODE
Monitor Area			– None
User Area			
Context	–	JIT	– Read
		DCB	– Read
		Buffer	– None
Data			– Write
Program			– Execute
Symbol Table			– Read
Special Area			– None or Execute

The basic purpose of memory protection is to protect the monitor and resident real time programs. The memory protection locks and keys are not changed during the running of different programs. This is in contrast to access protection which is changed every time a new program is run. Access protection is used to protect one user against another user and to protect a program from storing in its procedure area. Through a combination of memory protection and access protection, the UTS system is able to provide all the desired forms of protection.

3.1.4.4 RAD Management

A UTS configuration will contain one or more Model 7212 High Speed RAD's for swapping storage and storage of the operating system.

Contents
. Monitor
. Shared processors
. Users being swapped
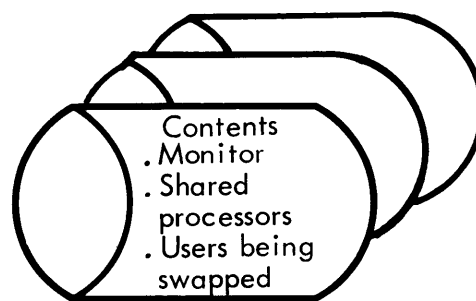
When a user logs on to the system, space is allocated to him on the swapping RAD. As his page requirements grow during his session, the RAD area necessary to contain him grows into consecutive sectors following his initial page, so that the individual user on the RAD is organized for swapping as efficiently as possible.

LAST PAGE ————————→   ←———— FIRST PAGE

ROTATIONAL
DIRECTION

RAD
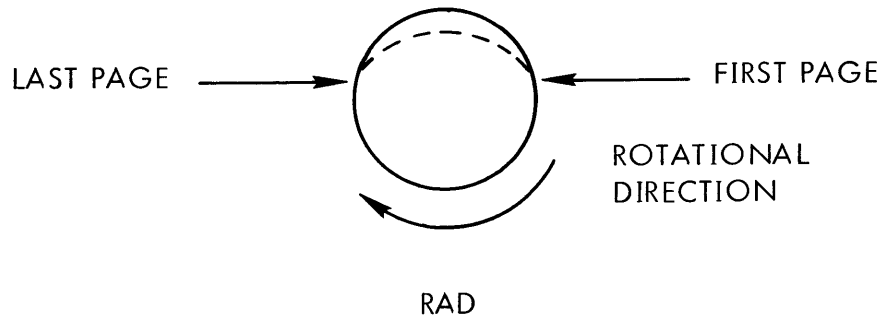
When multiple users are being swapped out, the I/O commands for each user are sorted and chained together so that the end of one user's area is the shortest possible distance from the beginning of the next:

I/O COMMAND CHAIN

USER NO. 3   USER NO. 4

————— SENSORS

RAD

USER NO. 2   USER NO. 1

USER NO. 2 ————→ USER NO. 1 ————→ HALT

USER NO. 3   START

————— USER NO. 4 ←———
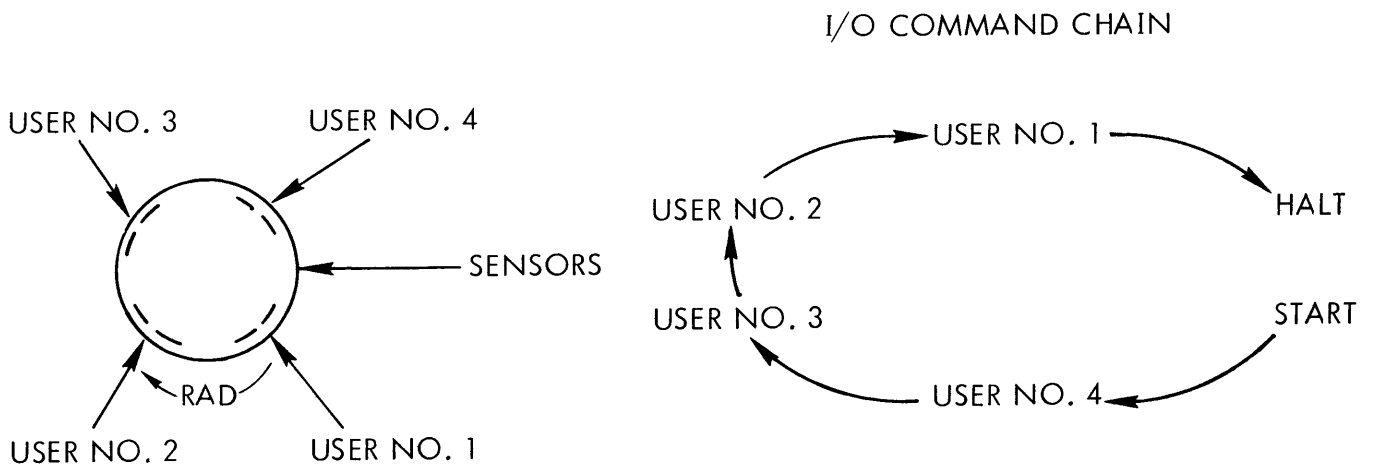
After this chaining is accomplished, a command is issued to the RAD to find where its sensors are located at that time. With this information, the chain of commands is broken and I/O is initiated to the RAD. This procedure reduces latency well below the RAD's 17 millisecond average access time. Similar logic is applied to the swapping in of multiple processors along with a user.

Additional storage is required for symbionts and permanent user programs and files. Typically, a combination of Model 7232 Extended Performance RAD's and Model 7242/6 Removable Disk Storage systems is used for this purpose. This combination provides the large capacity required for permanent storage.

3.1.4.5   Character Oriented Communications (COC) Handling

On-line terminal communications in UTS are generally handled by the COC handler. Full support for both full and half duplex terminals is included. Individual terminals are accessed by means of a table translation technique which defines the appropriate character code for that terminal type. All requests by a user for I/O to or from a terminal are handled through CAL instructions which consist of requests for read, write, and format control.

3.1.5   SYSTEM SERVICES

UTS provides a wide range of system services and facilities to aid both users and installation management in utilizing the system most effectively. These include facilities for logging on and off the system and for summarizing system utilization at log off time; flexible and convenient file management facilities; provision for extensive reporting and accounting of system use for installation management, including complete accounting of system use by separate account; and provisions for system integrity via diagnostic testing and recoverability procedures.

3.1.5.1   Log On

The log on procedure from an on-line terminal is used to initiate a user session. Log on may be done automatically or manually by connecting to the system and typing one's name, account number, and password —

```
UTS  AT  YOUR  SERVICE
ON  AT  12:01   MARCH  1, 1970
LOG  ON  PLEASE   NAME, ACCT, PASWRD
! . . . . . . . . . . . . . . . .
```

User sessions are ended by simply typing the command "OFF".

```
! ...
! OFF
```

3.1.5.2   File Management

The file structures described here are designed for utilization of random access storage media.
Three basic file organizations are supported: CONSECUTIVE, KEYED, and RANDOM. CONSE-
CUTIVE files are those which are created and accessed in order. KEYED files consist of a collec-
tion of records, with each record identified by a key. RANDOM files provide the user with a
contiguous random access storage space to be organized at his own discretion.

Data is read and written into random access modules in a block size defined by the granule size
of the particular device. When non-contiguous allocation is used, individual granules are
chained together to provide sequential linkage. RANDOM files utilize contiguous space and, thus,
do not require the chaining overhead. Records within a file may be fixed or variable in length.
KEYED and CONSECUTIVE files can have blocked or unblocked I/O. With blocked I/O, the
records to be inserted in the files are sent to a blocking buffer which is filled before any transfer
to the file takes place. Files may be read by more than one user.

UTS file management supports the use of magnetic tape, RAD, removable disk storage units, and
magnetic card memory.

3.1.5.3   Reporting

In a sophisticated time-sharing system such as UTS, it is particularly important to be able to mea-
sure the operation and performance of the system. UTS continuously gathers information of this
type and displays the results via a dedicated terminal. The items are displayed at regular inter-
vals.

Installation management may choose which items of information are to be displayed at various intervals — some every minute, some every hour. The items which can be displayed are grouped as follows:

- Summary
- CPU use
- Processors (CPU time)
- Processors (numbers of users)
- I/O rates
- Console time
- Users (number of)
- Interactions (number of)

The following is an example of a summary report:

Number of users	57.0
Tasks per minute per user	3.7
% of tasks which are interactive	95.8
CPU msecs per interactive task	5.0
90% point for response time (sec)	0.1
Execution multiplication factor	5.0
Number of users in core	10.0
RAD & tape read and write per second	25.0

3.1.5.4 Accounting

UTS satisfies installation requirements to properly account for resource usage in the dynamic time sharing environment. Cumulative records are maintained by user account number and name. UTS resolves the problem of variable charge rates dependent upon the time of day and/or type of user. All accounting data uses resident rate tables to generate actual accounting values. When the rate changes because of the time of day, the rate tables may be switched and the accounting is thereby modified to reflect the new rate.

Accounting controls for effective system management are provided by a special supervisor service routine available only to authorized individuals. Although supervisor-authorized account numbers may be entered or deleted from the system, cumulative resource changes for selected accounts may be inspected and limits set for individual accounts.

Items included in the accounting process are:

- CPU time
- Terminal connect time
- Core usage
- Terminal interactions (number of)
- I/O CAL's (number of)
- Tapes used (number of)
- Disk Packs used (number of)
- Pages of RAD used (number of)

3.1.5.5   System Integrity

One of the major requirements of an on-line time-sharing system is that it be highly reliable and available to its users at all times. To meet this critical need, a number of system integrity facilities have been designed into UTS. Three major objectives are sought: (1) to provide the highest possible security for user files, even in the event of total system failure; (2) to provide automatic high speed restart in the event of a system failure; and (3) to record sufficient information to isolate errors and failures caused by either hardware or software.

Following are some of the facilities included in UTS to help meet these objectives:

- Partitioning non-functioning hardware out of the system
- Watchdog timer for detection of software or hardware errors causing system hang-up
- Power fail-safe interrupt to save and restart system when power failure occurs
- Memory parity error detection and recovery
- Error-reporting log for maintenance purposes

- Automatic system recovery or restart in the event of system failure
- Snapshot of failed system (monitor) for subsequent diagnosis
- Diagnostic programs and exercisers

Particular attention has been given to the integrity of files by providing for the following:

- RAD file backup dumps
- Rollback of files from backup storage
- File survey and restructuring routine to detect errors, clean up storage, and report and restore bad files from backup storage

To handle the problem of an accidental user-terminal disconnect, UTS provides an automatic save of the user's current environment in a special file whenever a disconnect without a log out procedure is detected. When the user dials back into the system, he merely calls for this "save" file and continues his operation.

The system recovery function is provided to restore UTS to operational status very quickly after an un-recoverable failure. This is done by cleaning up all open-ended information (both user- and system-oriented), and restarting the system at its initialization point.

Entry will be made automatically to the recovery routine by various hardware and software detected errors. A manual entry point is also provided for use when the system cannot automatically recover. The recovery procedure includes the following functions:

- Display cause of failure
- Save monitor data and current user information for later analysis
- Close all open files with default options
- Package or release all partial symbiont files
- Package error log
- Inform users of interruption
- Save time, date, current system ID, error log pointers, accounting information, account numbers of users, symbiont file directories, and RAD granule usage map

3-20

- Package current backup tape

- Restart system and add items saved above

3.1.6   ON-LINE PROCESSORS

The performance and facilities of the Universal Time-sharing System monitor services are equally matched by the quality of the UTS processor subsystem. Each of the processors reflects the most up-to-date and most powerful version available in the industry.

All on-line processor subsystems are reentrant, guaranteeing maximum system performance by using only one copy of the program for all users. This means less swapping and less storage requirements for core memory and the swapping RAD. More users can be supported with good response time on an economical basis.

All on-line language processors operate in the batch mode as well, thus complementing the user's ability to conveniently enter jobs from an on-line terminal into the batch job queue.

The 12 language and utility processors available as UTS on-line subsystems are discussed below.

3.1.6.1   TEL

TEL is the executive service subsystem which conveniently interfaces the user to a variety of UTS monitor facilities as well as to other UTS subsystems. In particular, TEL permits direct access to most of the functions associated with program development in FORTRAN or assembly language. These include:

- Create or modify source program text files (EDIT)
- Compile or assemble source programs into relocatable object modules (FORTRAN IV, META-SYMBOL)
- Link relocatable program object modules into load modules (LINK)
- Modify object module symbols (SYMCON)
- Load and execute object programs, including searching for library subroutines

- Execute object programs under control of one of the debugging systems (DELTA, FDP)
- Resume execution of programs that have been interrupted or stopped by the user for debugging purposes
- Save core image of program being executed and retrieve it subsequently for continued execution
- Copy and delete permanent files

TEL also provides information service to the terminal user, such as his current session accounting charges and the status of available system resources.

3.1.6.2   UTS BASIC

The BASIC language, which was specifically developed for simple and convenient on-line use, has been further refined by XDS to provide more power and efficiency as well as more interactive facilities.

UTS BASIC includes the following features:

- UTS BASIC is truly conversational. On-line syntax checking is provided as each statement is entered from the terminal, and corrections may be made at any time to any portion of the program
- A powerful debugging tool is provided in the form of a direct execution capability. This facility permits statements, including file and input/output commands, to be immediately executed. Additional debugging aids include both compilation and run-time diagnostics, as well as a "snapshot" command
- Instantaneous compilation is performed in core at a rate of 500-600 statements per second to insure efficient object code generation and rapid program execution. A run/fast compile option allows production status to execute faster than when in the checkout stage
- UTS BASIC includes extensive intrinsic functions and matrix operations, including some which are new to the BASIC language
- Complete alphanumeric character string manipulation is provided to enable programs to "converse" with the user in English

- Simplified print editing and formatting facilities are provided, including tab, precision, and print, using image statements (providing maximum printing flexibility)
- UTS BASIC provides a chaining facility to allow programs to be written which would otherwise exceed memory capacity
- Programs created on-line may also be executed in a batch mode. This flexibility is unique to UTS BASIC

3.1.6.3   Extended XDS FORTRAN IV

For quality and performance, the Extended XDS FORTRAN IV compiler has long been noted as outstanding in the industry. Now it is available under UTS for use on-line as well as in the batch mode. To complement this powerful compiler capability, a FORTRAN debugging package, FDP, is also provided.

Extended XDS FORTRAN IV is a superset of most FORTRAN's available in the industry. It provides a large number of extended features, including:

- Mixed expressions
- Extended DATA statement capabilities including automatic data type conversion
- Unlimited dimensioning
- Simplified input/output for on-line use
- Extended relational operators and expressions
- Formatless input/output and automatic conversion during input/output
- On-line machine language coding
- Generalized DO loop parameters
- Use of expressions for DO and GO TO parameters, as well as for output lists and subscripts
- Multiple subroutine entries and optional returns

Diagnostics generated during compilation and execution are exceptionally comprehensive, leaving no ambiguities in language usage and specifying the exact nature of programming errors.

A second highly conversational type of FORTRAN, called UTS Conversational FORTRAN, will be available for on-line use only.

3.1.6.4 FDP

The FORTRAN Debug Package (FDP) provides the conversational user with many powerful features to reduce checkout time. These features, which are dynamically controllable from the terminal at execution time, include the following:

- Program execution control
- Statement stepping
- Conditional breakpoints
- Data breakpoints
- Flow tracing and history recording
- Examination and correction of scalars and array elements
- Branching
- Program restart
- Program examination
- Statement cancellation
- Argument display and automatic checking

Users may reference variables by name and statements by source line number or symbolic label. Such references may be further qualified by subprogram name.

3.1.6.5 META-SYMBOL

META-SYMBOL is one of the most powerful and flexible macro assemblers available. It enables a user to design and implement specialized higher order programming or application languages with high efficiency. Selective generation of code, controlled at the time of assembly, permits programs to be highly parameterized for different system configurations and operating environments.

META-SYMBOL is a two-pass assembler which provides the following significant features:

- Full use of lists and subscripted elements in procedure references
- Argument fields, containing both arithmetic and Boolean (logical) expressions, using constant or variable quantities
- Command procedures, permitting generation of many units of code for a given procedure call line
- DO and WHILE directives, allowing selective code generation with parametric constants or expressions determined at assembly time
- Function procedures, returning values to the reference line
- Call line and its individual parameters, to be tested both arithmatically and logically
- Procedures which can be nested and which can call each other

META-SYMBOL provides complete assembly documentation of a user's program, including symbol tables, concordance listings, and comprehensive error messages. Selective control may be exercised over documentation where only new portions of code need be fully documented. For conversational debugging under DELTA, META-SYMBOL optionally generates symbol tables within the object program output. Source programs may be input in various forms, including EDIT compatible, compressed, or compressed with updates. Output programs may be either compressed or not compressed.

3.1.6.6 EDIT

EDIT is a conversational line-at-a-time context editor designed for creating, modifying, and searching data text files. EDIT takes advantage of keyed file, random access storage for efficient and responsive file manipulation. EDIT functions include:

- Creation, insertion, deletion, reordering, and replacement of text lines or groups of lines
- Selective printing and renumbering of text lines
- Searching text files by context for matching, deleting, moving, and substituting line by line
- New file creation, file copying, and file deletion
- Intra-line editing for convenient modification of portions of a line, including shifting the text left or right

- Software tab controls setting for text formatting

3.1.6.7   PCL

PCL is a media conversion service for moving files of data in various forms from one type of peripheral device to another. It operates in the conversational on-line mode as well as in the batch mode. PCL provides comprehensive facilities that allow the user to:

- Transfer single or multiple files
- Select specific records within a file for sequencing, formatting, and conversion
- Delete files
- List or dump files
- Call for magnetic tape handling functions
- Copy files to devices (e.g., line printer)

3.1.6.8   DELTA

DELTA is a powerful conversational debugging service for checking and modifying assembly language programs. Patterned after DDT debugging systems, it permits the use of full symbolic references in an object program to perform the following functions:

- Examine, insert, or modify various elements of a program (instructions, numeric values, encoded information, etc.). Data may be referenced in all types and formats
- Control program execution with insertion of conditional breakpoints into a program, and breakpoint based on changes in elements of data
- Tracing execution by displaying information at designated points in a program
- Searching programs and data for specific elements and values

All UTS assemblers and compilers generate symbol tables describing data representation and program locations, which are used by DELTA to provide full symbolic referencing.

3.1.6.9 SYMCON

SYMCON is a convenient tool for program development tasks in which smaller subprograms are put together to form a larger process. SYMCON enables the user to examine internal symbolic references to determine if there are any conflicts in naming between the various subprograms. The user can eliminate those internal names which are no longer necessary, and he can change symbol names which may be in conflict.

3.1.6.10 LINK

The LINK subsystem forms executable program load modules from relocatable object modules. It is a one-pass linking loader used to combine a number of program elements into an executable entity. LINK merges internal symbol tables of the object program elements and searches subroutine libraries for external references.

3.1.6.11 SUPER

SUPER may be accessed only by installation management for the purpose of displaying user accounting statistics, defining legal users for the system, and deleting users from the system. It may be used on-line or in the batch mode.

3.1.6.12 CONTROL

UTS provides extensive resource-control facilities so that installation management may smoothly and effectively control system operation and optimize utilization of the hardware and software. At the time the system is generated, a number of parameters are defined to allow the system to be tailored to the specific requirements of the installation. These parameters may be dynamically modified at any time by using the CONTROL processor.

These parameters are:

- Maximum core allowed for all on-line users
- Maximum file space allowed for all on-line users

- Maximum number of on-line users to be conveniently served

- Maximum number of tapes allowed for all on-line users

- Batch bias - the percentage of execution time which batch receives

- Batch priority - less than or equal to compute bound on-line users

- Batch lock - locks a batch program into core memory

- Normal quantum by which users are time-sliced

- Minimum quantum guaranteed each user

Using the information provided by system monitoring, UTS will allow installation management to adjust the above parameters using the CONTROL processor for optimum resource utilization. This process is called "tuning" the system.

3.1.7   SYSTEM FLEXIBILITY

UTS features that contribute to system flexibility include:

- Terminal startup/shutdown facility

- Wide variety of parameter adjustments for system tuning

- Broad range of hardware configurations

- Easy upgrading from existing Sigma Batch Time-sharing Monitor (BTM) systems

3.1.7.1   Terminal Startup/Shutdown

Through the use of systems commands, on-line use of the system can be halted and available core memory can be redistributed for batch use. On-line operations can be resumed when desired.

3.1.7.2   System Tuning

UTS installation management personnel can dynamically adjust a number of system parameters (i.e., tune the system) by using the CONTROL subsystem.

The minimum and normal time quanta can be dynamically adjusted. The minimum percentage of CPU time to be allocated to batch processing can also be dynamically adjusted to insure that a

desired level of batch throughput is maintained. The number of active time-sharing users which the system will service concurrently is another dynamically adjustable parameter. Reducing this number will improve the response time for those on-line users being serviced. The adjustment of these and other parameters will insure efficient allocation of the system's resources as its environment changes.

3.1.7.3   Configuration Flexibility

The UTS operating system can accommodate a variety of Sigma hardware configurations, depending upon installation needs. The following are key elements that are used to create the optimum configuration for any specific operational environment:

- Separate storage systems with independent controllers for maximized file I/O and swapping. Separate controllers are recommended to avoid congestion between swapping and extensive file I/O activity

- High Speed RAD (Model 7212) for fast swapping to maximize the number of concurrent on-line users

- Additional core memory for accommodating larger user jobs or more simultaneous users in core memory. UTS was designed to operate efficiently with multiple users in core and this is best accomplished with a minimum of 128K of core memory if a variety of processors are in use

- Additional RAD storage for providing more permanent file space for users and temporary file storage for executing user programs. The Extended Performance RAD (Model 7232) is recommended for this purpose. Additional storage capacity at greater economy is provided by the Removable Disk Storage System (Model 7240/7241/7242).

3.2   XEROX OPERATING SYSTEM (XOS)

3.2.1   INTRODUCTION

The Xerox Operating System (XOS) brings to the Sigma user a true multi-use, multiprogramming operating system. XOS provides extremely high batch throughput plus concurrent time-sharing and real time operation. Whereas the emphasis in UTS is on its time-sharing powers, the emphasis of XOS is placed on its local and remote batch processing capability. Even though the entire

stream of jobs through XOS benefits from its multiprogramming capabilities, it is in the process-

ing of the batch job stream that XOS displays its real power. A high rate of multiprogramming

results from the working together of the major elements in the system. These elements include:

- Efficient job, memory, and resource management
- High CPU utilization through task management employing the proven Sigma priority inter-
  rupt system
- A comprehensive file management system supporting removable volumes and easy file crea-
  tion, maintenance, and user access

These monitor features, combined with a complete set of system utilities and language processors,

provide the user with a very capable and efficient operating system.

3.2.2   THE JOB STREAM

The XOS user has at his disposal several methods of introducing his jobs into the system. He may

submit them through local card readers or card readers associated with remote terminals. He may

submit them via a time-sharing terminal or he may have the jobs resident in the monitor, available

for immediate use as might be required by real time applications. Thus, the job stream present in

the system can have many sources. These sources are described in the following sections.

3.2.2.1   Local Batch Processing

The batch processing mode shows XOS's true computing power. Jobs are read into the system on

one or more card readers and are stored on disk units until they are processed by the CPU. Sym-

biont programs move the job stacks from the card readers to the disk units with minimal disruption

of the on-going processing in the CPU. Several queues of waiting jobs can be maintained by the

system. A job is moved from a waiting queue to an active state when it is the highest priority job

in its class (see below for descriptions of the different classes), and the system resources that it

requires are available for use. Once a batch job begins execution, it may be temporarily sus-

pended. The suspension may be caused by its asking for I/O, by an event occurring in a higher

priority program also present in memory, or by several other factors all assumed to have a higher

priority than the currently executing job. This assures that the highest priority jobs will be processed first and that CPU utilization will be as high as possible.

3.2.2.2   Remote Batch Processing

The computing power available to the batch user may be extended to the remote batch user through XOS's remote batch processing features. Once a remote batch terminal is activated, telesymbionts read jobs into the system and return results to the terminal. Telesymbionts perform the services for remote terminals that local symbionts provide for local peripherals. A remote batch job, once it is in the system, is processed as any other local batch job. This includes the ability to make use of the central site peripherals.

3.2.2.3   Real Time Processing

The real time capabilities of XOS, plus the extended interrupt features of the Sigma 9, enhance XDS's prominence in the realm of real time computing. XOS allows the user to include real time programs as extensions of the monitor. XOS also provides monitor facilities which enable immediate response to critical events external to the computing system.

3.2.2.4   Time-Sharing

(At the time of this writing, it is not yet known exactly what form time-sharing under XOS will take. There is a very good chance that a system equivalent to BTM will become the time-sharing portion of XOS.)

3.2.2.5   Data Base Sharing

Data base sharing under XOS enables the exchange of data between files centralized in a mass storage unit and a number of remote users. From keyboard/printer or display terminals, the user may update or retrieve data from shared files. XOS includes the facility of second level multi-programming (multi-tasking) which provides efficient management of all the secondary systems of data base sharing.

3.2.3   MULTI-PROGRAMMING FEATURES

As do most large operating systems, XOS had a high rate of throughput as one of its goals. However, unlike other large operating systems, XOS has the distinct advantage of being coupled with Sigma hardware and its unique capacity to process and overlay I/O. This parallel operation is not the only feature of Sigma architecture exploited by XOS. XOS makes extensive use of the Sigma interrupt system in managing the different tasks that may be simultaneously present in memory. Sigma architecture provides a hardware system capable of high throughput and XOS provides monitor routines and services to use that capability to its fullest. XOS includes routines to efficiently manage both the system's resources and the many users simultaneously present in memory.

Each user in memory belongs to one of several classes of jobs. The three following sections describe these classes and the types of jobs they normally contain.

3.2.3.1   Parallel Job Class

The parallel job class consists of catalogued items containing only one job step. These jobs are initiated by the operator from the operator's console. Jobs found in the parallel job class include media conversion jobs and jobs that initialize real time tasks. The cataloguing of jobs is performed by the command card interpreter (discussed later). Cataloging the command cards of a job not only reduces the duplicated effort involved with a frequently run job, but also assures that the job will be run exactly the same way each time. There is no fixed upper limit on the number of parallel jobs that may be present in memory. The only restraint is the physical size of the memory in the system and the availability of other system resources.

3.2.3.2   Production Job Class

Jobs belonging to the production class normally enter the system through a card reader, have several job steps, and require relatively large amounts of I/O during their execution. The production job class is divided into five subclasses. Each subclass may have one active job in memory and a file of waiting jobs in secondary storage. It is therefore possible to have a maximum of five production jobs simultaneously present in memory and five open files of waiting jobs. At any given instant, there is only one active job; i.e., one job in memory, per file. The waiting job queues

enable the submission of concurrent job streams and obliges the job scheduler to overlap them step by step. This provides a means of control of the multiprogramming and a tool for planning the activity of an installation.

3.2.3.3 Serial Job Class

Jobs belonging to the serial job class normally are long, heavy computing jobs requiring relatively small amounts of I/O. They normally enter the system through a card reader and contain several job steps. XOS maintains a file of waiting jobs for the serial job class. Only one serial type job is in memory at a time.

3.2.3.4 Multiprogramming Rates

XOS affords a high multiprogramming rate. In fact, it is possible to have present in memory at one time the following user jobs:

- Any number of parallel class jobs. The number is limited only by the available system resources
- Up to five jobs from the production job class; one from each of the five subclasses
- One job from the serial job class

In addition to the above jobs, also active in memory may be the input, output, and telesymbiont service jobs.

3.2.4 BASIC FUNCTIONS

The basic functions constitute the nucleus of the monitor. They determine the system's relationship with the hardware.

3.2.4.1 Task Management

To the nucleus, every active job is a task. Tasks have different states: active, interrupted, idle, and waiting. Task management treats the state changes.

The tasks present in the system are distributed on different levels of external interrupt. The assignment of a task to an interrupt level depends upon the class to which the corresponding job (task) belongs. A task is initiated by triggering its corresponding level of external interrupt. If a higher priority level is active, the task cannot gain control and is in a wait state. If a lower priority level is active, the task interrupts the lower priority task and gains control of the CPU.

3.2.4.2   Memory Management

XOS manages memory through the Sigma memory map. A maximum of 128K words of virtual memory is available to each task. However, through control cards the user may define smaller amounts of usable virtual space available to a job or job step. It is possible for a task not to need all of the virtual memory assigned to it. XOS can detect this situation and will move non-resident portions of the monitor into the unused pages. Should the program wish to expand into the pages now containing portions of the non-resident monitor, a monitor routine will first release those pages occupied by the less frequently used segments of the non-resident monitor. With this method of memory management, some heavily used portions of the non-resident monitor may appear to be resident.

The pages freed at the completion of a task are returned to a free-page pool. There the monitor may assign them to a newly initiated task or to an existing task in response to a user's dynamic request for more memory.

3.2.4.3   Input/Output Supervisor

The input/output supervisor responds to all requests for I/O from the user and the rest of the monitor. In responding to these requests, it performs the following functions:

- Queues the I/O requests at the user task level
- Initiates all I/O
- Prepares recovery operations in case of an error
- Processes all I/O interrupts
- Communicates to task management the status of I/O requested by a task

The I/O supervisor optimizes the flow of data through the system whenever possible. In particular, it optimizes the movement of data to and from the RAD's and removable disk storage units. When communicating with the RAD, the I/O supervisor takes into consideration the position of the disk. RAD I/O requests are chained to one another to form a single request according to their initial sector address and transfer length. The resulting single request is forwarded to the controller. When communicating with the disk packs, the I/O supervisor attempts to optimize the disk's arm movement. Two files of waiting requests are maintained for each disk.

Requests are placed in either file according to the request's cylinder address and the address of the active file. When the active file is exhausted, the I/O supervisor switches to the other file and initiates those waiting requests.

3.2.5 SERVICE FUNCTIONS

The service functions provide the communication paths between the monitor and the user.

3.2.5.1 Operator Communications

This function controls the communication between the monitor and the operator. The exchange of information may be initiated by either the system or the operator. The system sends messages concerning the peripheral devices and changes of state of jobs submitted to the system. The operator may request the initiation of real time tasks, a display of the status of components in the system, or the logical removal of a failing peripheral from the system.

3.2.5.2 Symbionts

Symbionts are tasks belonging to the monitor. They are tasks of the parallel class, in that they are activated by the operator from the operator's console. XOS includes input/output and telesymbionts. Input symbionts see each incoming job as a set of command cards with associated data cards. It moves these cards onto the disk and calls for the command card interpreter. There may be several input symbionts active at the same time.

Job results are not transmitted directly to the slow output peripherals. Instead, they pass from memory to the relatively fast disks and then, under output symbiont control, from the disks to the slower peripherals. This avoids long waiting times for the CPU.

Telesymbionts enable the use of remote batch terminals. A telesymbiont simultaneously performs the input function and output function of the local symbionts.

3.2.5.3   Command Card Analysis

This function is performed by the control command interpreter. The interpreter has three modes of operation:

- Analysis of job command cards to check syntactical correctness without actually executing the job
- Analysis of job command cards followed by job execution (if cards are correct)
- Analysis and cataloging job command cards without job execution

The command card interpreter is also capable of chaining several job steps into one large job.

The cataloging of command cards simplifies the activation of jobs that are periodically run. Certain elements appearing in the option fields of a command card may remain undefined during the cataloging process. The control command interpreter has the ability to later provide definitions for these undefined elements. The definitions may come from the operator or other control cards. This feature of the monitor provides for command card parametering.

3.2.6   SUPERVISORY FUNCTIONS

The supervisory functions include the job scheduler and job management. The job scheduler selects the jobs to be activated. Job management initiates jobs activated by the scheduler.

3.2.6.1   Job Scheduler

The selection criteria used by the scheduler to activate a job are:

- The class (parallel,production,or serial) to which a job belongs
- The availability of the system resources needed by the job
- The relative priority of the job (for serial class jobs)

At system generation time, the user defines the different classes among which user jobs will later be distributed. The system implicity recognizes two classes, the parallel job class and the serial job class. Up to five subclasses of production class jobs may be explicitly defined. The user specifies the relative priorities of these subclasses and the serial class. The parallel job class automatically has the highest priority.

The job scheduler leaves the idle state and takes control of the system during the release of system resources,at the completion of a job or job step, and during the introduction of a new job into the system after it has been processed by the control command interpreter.

For each job class (and subclass) there may exist a file of jobs waiting to be activated. The scheduler examines the waiting job queues in the order of class priority. If any parallel class job are waiting, the scheduler will attempt to activate them if the required system resources are available. If resources are not available, the scheduler returns to the idle state. If the parallel job waiting queue is empty, the scheduler then examines the waiting queues for the production and serial classes.

If a production class job is already active, the scheduler examines the next file in order (which may belong to the production or serial class). If no production class job is active, the scheduler will follow one of two paths. If the resources required by a job are available, the job will be activated and the scheduler will proceed to examine the next file in order. If resources are not available, the scheduler returns to the idle state. Within the waiting queues for the production subclasses the scheduler examines jobs in chronological order.

The job scheduler activates serial class jobs similar to production class jobs except that the single waiting job queue for the serial class is sorted by job priority. If no serial class job is active, the scheduler looks for the first job which can be activated; i.e., one for which the necessary resources are available. If no job in the waiting queue fulfills this condition, the scheduler returns to the idle state.

The job scheduler recognizes three types of system resources. These are:

- The shared resources - CPU time, memory space, system disk space reserved for handling temporary files
- The common resources - peripherals that the user wishes to appropriate for the duration of the execution of his job without precisely naming them. The system chooses the specifier devices from the free devices
- The reserved resources - peripherals requested by the user supplying a specific device address

During system generation, the maximum values of the different types of resources are established by the user. When a job is brought into the system, its resource demands are compared with the maximum available resources for the class in which the job is to be executed. If there is an incompatibility, the job is rejected. The same resource may be assigned to several classes. For example, if there are three tape units in the system, all three may be assigned to several classes.

A job is effectively activated only when the totality of the resources it requires are available. However, the user may optimize his management of resources by furnishing the system supplementary information with the following command cards:

- The S LIMIT card enables an increase, during the job step to which it applies, of the implicit or LIMIT card fixed limits of the shared resources. The presence of this card causes, at step activation time, a call to the job scheduler for verification of the availability of the concerned resource. In case of unavailability, the job is put on wait. At the end of the job step, the limit values are brought back to the initial values

- The resource card allows the user to specify to the system, for the common resources it identifies, some minimal requirements for job initiation. The job will be able to make resource requests which exceed these values. The job scheduler will be activated and the job placed in wait if the request cannot be satisfied. The reservation of common resources which are not stated in the RESOURCE card take place globally at the time of activation

Restitution of the common resourcer is done as soon as possible. When a certain number of units are no longer necessary for the following job steps, they are returned to the system.

3.2.6.2   Job Management

Job management controls the initiation of jobs activated by the scheduler, program loading and the chaining of job steps. Before the initiation of a program, job management verifies the presence of the removable volumes which will be used during execution. It sends "mount" messages to the operator asking for the volumes not present and sends "dismount" messages for volumes to be saved at the end of a job.

Job management also keeps accounting data to allow the installation to evaluate the cost of each job.

3.2.7   MONITOR SERVICES

Monitor service routines provide the connection between the user and the nucleus of the monitor. The user invokes the monitor services through macro-instruction in META-SYMBOL.

3.2.7.1   Management of Job Storage Space

These services allow the user to dynamically check the number of free pages available to him. If any pages are available, he may then request a number of pages up to the limits established for him. The free pages may come from the user's local dynamic area or from the common dynamic area shared by the user and the monitor. When the user has finished with a block of space, he frees it to be used by others.

3.2.7.2 Program Management

Program management services enable the user to bring into memory an overlay (program segment) or a whole program. The user may choose to retain or destroy the calling program.

3.2.7.3 Checkpoint/Restart Services

The checkpoint-restart services give the user the capability to preserve the state of his job at a given moment and to return at a later time to the same state. The system preserves all of the information necessary to restart the user job and the corresponding states at the time of the call to checkpoint. The information is saved on a file established by the user. After the call to checkpoint is processed, the user's job continues execution in sequence.

When the call to restart is made, the job is re-initiated in the state in which it had been when the checkpoint was created. In particular, the removable volumes which were unloaded at the creation of the checkpoint are now remounted and repositioned. The Data Control Blocks are reset to their state at the time of the checkpoint.

3.2.7.4 Trap Management

Trap management allows the user to specify to the system the action to be taken when a trap occurs during the execution of a program. The user may request that his program be given control on the occurrence of a particular trap. He may ask the system to ignore such traps as those associated with fixed point and decimal arithmetic. Should the user decide that he no longer wanted to maintain control over traps, he can instruct the system to establish the form of trap management that was in effect prior to his asking for control.

3.2.7.5 Input/Output Services

XOS includes monitor services that assist the user in accessing his data files. The monitor services enable the user to:

- Access the next record or a specified record in a file

- Insert or modify a record in a file

- Skip to the first record of the next block from any position in the current block

- Delete records from files

- Move to the next volume of a multi-volume file

- Note his current position in a file

- Position a file at a specified place

- Send specific commands such as page skipping on the printer and forward and backward skipping on magnetic tape to certain types of devices

- Select a partition in accessing files with partitioned organization

- Insert a new principle partition identifier in the index of a partitioned file

- Read and write physical records

- Check the completion of I/O operations

3.2.7.6 Debug Control

XOS debug controls provide the user with a means of displaying selected parts of his program or data during the execution of his program. The display may be produced conditionally or unconditionally. Monitor services are available to set up the conditional tests.

3.2.7.7 Miscellaneous Services

Included in the miscellaneous procedures are those that: provide the time and date; set and check interval timers; provide user communication with the operator; set, reset, and check the 32 pseudo keys; put a program in the wait or idle state.

3.2.8 FILE MANAGEMENT SYSTEM (FMS)

The File Management System consists of a collection of monitor services which provide the connection between user programs and their associated data files and the easy reading and writing of these files by the programs.

Services available through FMS optimize the global data flow of the installation – inputs and outputs move through buffers and peripherals operate in asynchronous ways in connection with waiting queues managed by the monitor. A complete simultaneity is thus achieved between internal processing and an unlimited number of I/O operations. The system uses chains of I/O requests. One I/O operation may be started as soon as another is finished without disturbing the CPU. This eliminates all waiting, realizes a maximum simultaneity of the processing, and best uses the peripherals.

3.2.8.1   File Organizations

Through FMS, the user may organize his files in any of the following ways:

- Sequential organization: Records are written one after the other sequentially on the medium. Magnetic tapes may use only the sequential mode
- Indexed sequential: An indexed sequential file consists of a sequential file together with indexes. The indexer permits quick access to a record of data and the sequential format permits rapid sequential processing.
- Direct organization: In this organization, a file is considered as a group of blocks, each block being directly accessible if its position, relative to the beginning of the file, is known
- Partitioned organization: A partitioned file consists of several sequential groups of records. The file also contains a directory containing the name and beginning location of each group. The names in the directory are sorted in alphabetical order.

Disk files may use any of the above organizations. Tape files may use only the sequential organization.

3.2.8.2   File Access Methods

XOS provides both assisted and unassisted access to the files described in the preceding section. In assisted access, the monitor supplies automatic blocking, deblocking, and buffering. In the unassisted modes, none of these services are provided.

Assisted access methods:

- Sequential – this method enables the reading or writing of sequential records
- Indexed Sequential – this method enables the creation of an indexed file, access to records in that file according to their keys or sequentially, and updating of the file with modification, deletion, and insertion of records
- Partitioned – this method provides direct access to the beginning of a file partition. Once the beginning of the partition is found, the data records within the partition may be processed using the same macro instructions used in the assisted sequential access method

Unassisted Access Methods:

- Sequential – unassisted sequential access allows the reading and writing of physical records in any file
- Virtual Direct – this method, which can be applied only to direct access devices, allows reading or writing of physical records of any length, always beginning on a block boundary
- Real Direct – this method, which also can be used only with direct access devices, allows the reading or writing of data starting with a real disk address

3.2.8.3   Other FMS Features

FMS supports removable volume. These physically are removable disk packs and reels of magnetic tape which are identified by serial numbers. The serial numbers are written on visible labels on the volume as well as being recorded on the volume by a FMS service routine. The following types of volumes may exist in the system.

- Private volumes – this is a removable volume belonging entirely to one user
- Public or common volumes – a removable volume which can be used by any user for file creation. The creation of a permanent file on a public volume transforms the public volume into a private volume. The creation of a temporary file will not cause this transformation. At the end of their use in a job, public volumes carrying temporary files are automatically returned to the system and are available for use by others

- Accounting volumes — a removable volume used by the system for the retention of accounting data. The serial number of the accounting volume is permanently known by the system.

On both magnetic tape and disk, FMS supports the following organizational structures:

- Monofile, monovolume
- Monofile, multivolume
- Multifile, monovolume
- Multifile, multivolume

Magnetic tape labels associated with volumes and files contain specific information about the file and the volume. The labels and volume structures conform to USASI norms. Magnetic tapes may also have non-standard labels or no labels at all.

Labels on magnetic disk contain:

- A volume label with volume number and owner number if necessary
- An allocation table representing the disk space available on this volume
- A table containing file labels which describe the placement of files on this volume

Disk packs may also have no labels or non-standard tables.

3.2.9   SYSTEM PROCESSORS

Included in XOS are system processors to assist the user in forming load modules, generating the system, and performing utility type functions for the file management system.

3.2.9.1   Linkage Editor

The major functions of the linkage editor are as follows:

- Create the linkages between modules resulting from different assemblies or compilations and

library modules in order to build a single program

- Combine modules according to a user defined tree structure

The load modules created by the linkage editor may be either absolute core images or relocatable.

Options on the control cards that direct the operation of the linkage edits provide, among other things, the ability to correct an existing load module. The correction may be in the form of a change to existing instructions or the insertion of new instructions. The linkage editor also allows for the redefinition of an externally defined symbol within a load module.

3.2.9.2   System Generator

The monitor's system generator is a group of procedures which permit the generation of a system adapted to both the hardware configuration and the software demands of an installation. System generation can be executed as a serial class job under XOS.

The routines in the system generator are:

- SYSPROC - adapts system modules to the installation
- SYSEDIT - a link editor
- SYSREL - a relocating loader

The modularity of the system generator allows for great freedom of operation. For example, a system does not have to be generated all at once. It may be generated in discrete pieces and later joined into the resulting operating system.

3.2.9.3   FMS Utilities

The working system employs several service processors linked to the file management system. They provide such services as the pre-marking of volumes and the reorganization of indexed or partitioned files. It also includes the file management processor which allows the user to easily add, copy, or delete files during the execution of their jobs.

3.2.10   LANGUAGE PROCESSORS

XOS includes the META-SYMBOL, FORTRAN, and COBOL language processors. Descriptions of these processors may be found in Section VIII, Proposal Material.

SECTION IV

APPLICATIONS PROGRAMS

4.0   INTRODUCTION

This section contains listings of applications programs available for Sigma 9, a brief description of each, and references for additional data.

4.1   XDS APPLICATIONS PROGRAMS

In addition to its powerful operating system and utilities, the Sigma 9 is supported by a large and growing library of applications programs. The following eight program packages are examples.

- ## CIRC-DC

  Subsystem — Circuit Design Tool — Batch or Conversational

References:	64-78-17A	Data Brief
	90-16-97A	Reference Manual

- ## DMS

  XDS Generalized Data Management System

References:	64-70-13A	Brochure
	67-04-30	Sales Presentation (8-1/2 x 11 or 17 x 22 charts)

- ## GDL

  Graphic Display Library — Programming package of subroutines for creating and manipulating images on the Model 7580 Graphic Display device

References:	64-00-12A	Brochure
	Product Marketing Bulletins Nos. 8, 9, 10, and 12	

67-04-25   Sigma Graphics – 7580 Presentation – (8-1/2 x 11
                or 17 x 22 charts – also narrative)

● FMPS

Functional Mathematical Programming System — Linear programming package. Operates also in non-linear mode. FORTRAN oriented control language. Optional GAMMA III Matrix Generator and Report Writer provides convenient form for describing input and reports.

References: 64-79-09A   Brochure
               90-16-09A   Reference Manual
                            FMPS Sales Presentation – (8-1/2 x 11 chart)

● MANAGE

A generalized file update and retrieval system. Terminal Oriented MANAGE (TOM) capability allows rapid and easy entry of retrieval and report specification.

References: 90-16-10A   Reference Manual
               67-04-01   Sales Presentation – (8-1/2 x 11 or 17 x 22 chart
                            and narrative)
               64-79-10   Brochure

● SL-1

XDS Simulation Language — Problem oriented language, for digital or hybrid simulation. A superset of CSSL (Continuous System Simulation Language), the standard language specified by Simulation Councils, Inc., for simulation of continuous systems.

References: 66-07-12A   Brochure
               90-16-76A   Reference Manual
                            User News – June/July 1970
                            SL-1 Sales Presentation – (8-1/2 x 11 or 17 x 22
                            chart)

- GPDS

  General Purpose Discrete Simulator — A superset of the IBM GPSS/360 general purpose system simulator.

  References: Drawing 702580 Functional Spec

- CPM

  Sigma Project Management System — Critical Path Method

  References: 90-15-04A   Reference Manual (9-series, but serves as interim until new Sigma manual completed)

4.2 USERS' LIBRARY PROGRAMS

The Users' Library now contains a formidable inventory of useful programs (well over one thousand), and continues to grow at a good rate. It currently contains many useful contributions to the Sigma 9 applications programmer, of which the following are but a few.

- ALGOL

  ALGOL 60 — Vanderbilt University

  References: 890357-11   BPM - 117 pages

- LISP 1.5

  Vanderbilt University List Processing Programming Language — Has been used for Artificial Intelligence Research. Suitable for symbolic calculations in Calculus, Mathematical Logic, Game Playing, and many other fields. Batch and interactive.

  References: 890366-11

- SOL

  Vanderbilt University — General purpose language for simulating complex systems.

  References: 890363-11

- **GASP II**

  XDS — Event oriented generalized activity simulation program for discrete simulation.

References:	890559-11	Interactive
	890560-11	Batch

  "Simulation with GASP II", by Pritsker & Kiviat, Prentice Hall, 1969

- **SNOBOL4**

  XDS — Derived directly from the original Bell Telephone SNOBOL compiler by implementing the macros in which the compiler is written.

  References: 705848-11

- **Bucknell Data Processing Package**

  References: 890619-11   (Includes documents listed below)

  - Payroll Package — 16 programs (COBOL & FORTRAN) for producing checks, printing payroll distributions, check registers, time reports, social security reports, local tax reports, and all associated record keeping (890562-11)
  - General Ledger (890592-11)
  - Bookstore (890629-11)
  - Accounts Receivable (890626-11)
  - Accounts Payable (890620-11)

- **COBOL**

  Keyed-File Utility Subroutines — XDS-series of very useful subroutines to assist COBOL programmers.

  REFILE — Closes and releases files to the monitor

  DELREC — Deletes records from keyed files

GETCOM — Gets date, time, and switch settings

KEYSTART — Obtains key value of last record read

ADDSEQ — Adds a record to a keyed file being processed sequentially

PAPERCHG — Instructs operator to change paper, carriage tape or punch, or card stock

BDP$PRT — Prints reports (previously written to tapes) on forms that need alignment

BINARY SEARCH — Searches a table for a specific element in the table

References: 890598-11

- DITTO

    IMT (Information Management Technology – Fargo) — Developed as replacement for the System/360 "DEBE" program. (generalized file-to-file — card to tape, tape to printer, tape to tape, etc.) — 34 functions in all — extremely useful.

    References: 890617-11

For a more complete list, a KWIC index, together with abstracts, is available by sending a request to the:

    Secretary, XDS Users' Group
    Xerox Data Systems
    5300 W. Century Boulevard
    Los Angeles, California 90045

SECTION V

PRODUCT PERFORMANCE/COMPETITIVE INFORMATION

5.0 INTRODUCTION

The competitive information in this notebook is presented as six independent, stand-alone sections (Competitive Sales Guides) as follows:

- Sigma 9 Competitive Overview
- IBM 370/155, 360/67 Competitive Report
- Univac 1106/1108 Competitive Report
- CDC 6200/6400 Competitive Report
- Burroughs 6500 Series Competitive Report
- DMS Competition: Special Report

Copies of these Sales Guides will be distributed to each District Office to be filed in the Competitive Information Reference Library, Volume I, according to the filing instructions on each Sales Guide.

All members of Competitive Analysis contributed to each of the individual Competitive Sales Guides contained in this notebook. Specifically:

Gerry Blowey	—	Provided the analysis on Communications, Peripherals, and assisted in Price/Performance Analysis.
Errol Forkner	—	Provided the analysis on CPU's, Memory, Configuration Diagrams and assisted in Price/Performance Analysis.
Bob Kemp	—	Provided the major portions of all Software Analysis and assisted in the other analyses.

Competitive Sales Guide

25 OCTOBER 1970   COMPETITIVE ANALYSIS   File: XDS
                                                              SIGMA 9

This Sales Guide is intended to provide you with an overview of the Sigma 9 competition and market. The competitors discussed in this report and their primary software operating systems are:

MFR	HARDWARE SYSTEM	SOFTWARE SYSTEM
IBM	370/155	OS/MVT
	360/67	TSS
UNI	1106, 1108	EXEC 8
CDC	6200, 6400	KRONOS
		SCOPE
BUR	6503, 6504, 6506	MCP
GE	635	GECOS III
RCA	70/60	TDOS
	(R6)	(OS/70)
	70/61	OS/61
	(R7)	(VMOS)
DEC	PDP-10	10/50 MDM

On the next page you will find the table of contents for this report.

Distribution: Sigma 9 Announcement   Kent L. Cootes, Manager
Distribution and Competitive Infor-   Competitive Analysis
mation Reference Libraries   XDS   AFC

FOREWORD

This Sales Guide is intended to provide you with an overview of the major
competitors you will be encountering against the Sigma 9. Because of the
new product announcements by other manufacturers that are now occurring,
it is not possible to fully define what your competition will be in the near
future. Competitive Analysis plans to keep you informed as these announce-
ments occur.

Also, this overview is not complete with respect to the material presented
on the individual competitors discussed herein, nor is it intended to be —
it is an overview. For additional information on individual competitors
you should consult the Sales Guide specifically concerned with that com-
petitor. For information on a competitor not covered in these reports or
for competitive information not included, contact the appropriate individ-
ual in Competitive Analysis.

Remember, it is your job to sell solutions to problems. The purpose of this
document is to provide you with the necessary technical support to sell
XDS solutions and to provide the proof that we can indeed provide better
solutions because of the superiority of the Sigma 9 as a total system.

Good Selling!

CONTENTS

COMPANY PRIVATE

MARKET AND COMPETITIVE OVERVIEW

- ## MARKET OVERVIEW

   The market in which the Sigma 9 fits is characterized by the following attributes:

   - System Purchase Price Range: $1.5M to over $5M (multi-processor systems)

   - System Lease Price Ranges: Typically from $30K/month to $100K/month (multi-processor systems)

   - Internal System Speeds (Unit Processor Systems): 500 KIPS to 1000 KIPS

   - Main Memory Ranges (Unit Processor Systems): 64K words to 512K words

   - Identifying System: IBM 360/65, 370/155

   - System Use: Multiple concurrent use — typically scientific batch, commercial batch, and remote batch

   - Typical Operating System: Batch Multiprogramming with communications management. Trend is to add some time-sharing capability

Current market statistics for individual systems in this market class are summarized in the table on the following page. The total value of the approximately 1500 currently installed systems in this class is over 3 billion dollars. IBM has approximately 55 - 65% of this market with UNIVAC second (about 20% currently).

CURRENT MARKET STATISTICS

Mfr.	Model	No. Installed	No. On Order	Date of 1st Inst.
IBM	360/65	625	775	11/65
	360/67	50	60	6/66
	370/155	0	750	3/71
UNI	1106	20	30	12/69
	1108	125	30	9/65
CDC	6200	0	12	9/70
	6400	65	12	5/66
BUR	5500	130	5	3/63
	6500	15	50	6/69
RCA	70/60	0	15	9/70
	70/61	0	10	2/71
GE (HON)	635	75	12	5/65
	655	0	6	3/71
DEC	PDP-10	110	25	10/67
TOTALS		1215	1792	

● ABOUT THE COMPETITION

The hardware and software combinations provided for the systems which you will be encountering as competition to the Sigma 9 represent the ultimate the computer industry has to offer today. There are, of course, larger and faster hardware systems, but the software is, essentially, the same and the hardware complexity is not much greater. Therefore, you have a big job to do — learning about these systems and how to sell in this marketplace. The key to selling Sigma 9's will not be the hardware capability of the Sigma 9, although this will certainly help you. The key will not be price, although this will also help. The key will depend upon your ability to understand the prospect's problems and convince him that XDS can solve these problems. How? Through software. For the first time, XDS is providing you with a basic set of software tools that will allow you to sell against anyone, including IBM. (But, be very careful about trying to sell to current IBM 360/40, and up, users.) To give you some idea of how the Sigma 9 compares in a very gross sense against the competition, take a quick look at the chart on the next page.

A GROSS COMPARISON OF CAPABILITIES AMONG SIGMA 9 COMPETITORS

Remote Batch	Commercial Batch	Scientific Batch	Time-Sharing	Critical Real Time
IBM 370/155 (OS MVT + Options)	VG	VG	G	P
IBM 360/67 (TSS)	P	G	G	P
UNIVAC 1106/1108 (EXEC 8)	VG	E	P	P
CDC 6200/6400 (KRONOS) (SCOPE)	G F	E E	F X	P P
BURROUGHS 6500 (MCP)	G	E	X	X
RCA 70/60, 61 (TDOS) (OS/61)	G VG	F to G F to G	X VG	X X
DEC, PDP-10 (MDM)	X to P	G	VG	G
SIGMA 9 (XOS) (UTS)	VG F	E VG	G E	VG E

E = Excellent
VG = Very Good
G = Good
F = Fair
P = Poor
X = Non-existent

The results presented on this chart are, of course, not completely objective since they are based partly upon the combined experience of each Competitive Analysis analyst and on inputs from the field. There is no unanimous opinion on what factors, and what weighting should be included in making such an assessment. However, it is our belief that, upon reading the individual reports on each competitor and the material presented in this overview, that you will come to very similar, if not identical conclusions.

Before providing you with some proof of this — from a hardware, software, and price/performance basis — we shall discuss the importance of software in the Sigma 9 marketplace.

THE IMPORTANCE OF SOFTWARE IN THE SIGMA 9 COMPETITIVE ENVIRONMENT

- **INTRODUCTION**

  The Sigma 9 introduces several new and challenging dimensions for "total capability". These new dimensions have a potential never before available in XDS and are primarily based on software that will be operational upon first delivery of a Sigma 9. This, in itself, is a competitive advantage.

  These software systems were specifically designed for their intended marketplaces: UTS for the time-sharing/scientific/batch market and XOS for the general purpose/commercial/time-sharing market. This presents a total capability unexcelled in the computer industry.

- **BACKGROUND**

  In the past, XDS had the reputation for having "hot hardware" for real time applications. That reputation was expanded by adapting the real time hardware features to the time-sharing market with outstanding success in the 940 computer. The Sigma computers have proven their capability in the general-purpose, scientific marketplace with time-sharing overtones. With the Sigma 9, XDS offers a combination of hardware and software that will impact the computer market in those areas where the customers have been extremely upset with other vendors such as IBM, Univac, and CDC. It is no longer a matter of "what have these vendors done for me in the recent or ancient past" – it is "what are these vendors doing for me today to help me solve my pressing problems"?

- **WHAT PROBLEMS CAN XDS SOFTWARE SOLVE?**
  - Implementing, Evaluating and Utilizing Advanced Applications: The pressing problems referred to above are the capability to address new and challenging applications within reasonable costs. The actual cost of the computer is not insignificant, but its significance is certainly reduced when examined as only one part of the analysis, system design, implementation, evaluation and operational checkout costs over the life of a specific (new and challenging)

application that solves a very real and specific problem. Where does the total capability of the Sigma 9 hardware/software combination fit into this overall environment?

- Analysis: Modeling and statistical correlation are primary candidates for this area. The Sigma 9 offers GPDS for discrete simulation, SL-1 for analog simulation, FMPS for optimization studies, PERT for network planning and statistical packages — all usable from remote terminals — either remote batch or interactive terminals. Also, there are powerful processors, FORTRAN and COBOL, for explicit definitions of problems and reports related to analysis.

- System Design: The most significant way to evaluate a particular systems design is to try it out for size for a while and see if it really is what is wanted. If changes are required, and they usually are, then they should be capable of being made without a major overhaul of the system. The Sigma 9 offers MANAGE, Terminal Oriented MANAGE, and DMS for this type of activity as well as operational features if desired. These systems lend themselves to changes and more or less self-reorganization with very minor effort on the part of the systems designer.

● IMPLEMENTATION

Now we get to the area where the computer must become a tool to assist in the explicit definition of the advanced application! The major cost to the customer in this area is the cost of programming! What does the Sigma 9 offer the potential customer for this problem. This is the area where the Sigma 9 offers more capability than any competitive vendor! Let's discuss it as follows:

- Efficient Program Development: Facilities are offered both in UTS and XOS to combine interactive as well as batch and remote batch program development. Not only are a variety of problem oriented languages supported (FORTRAN, COBOL, etc.), but an extremely high level macro language (META-SYMBOL) highlights the system for its very superior systems development quality. These features would be almost meaningless if they were not fully supported by debugging features that are unequaled in the computer industry (Monitor Debug, FDP, DELTA, and individual language processor debug facilities)! And, fully combined into this total environment is not only the communications "real time" as referred to by other vendors, but the XDS real "real time" of fast response to external stimuli. This type of program development, for real "real time" applications, is also supported by the real time extension to our language processors. This real time extension to our operating systems (monitors) and the capability of real "real time" program development from interactive terminals, remote batch terminals as well as the normal batch environment. These characteristics offer the applications implementor a level of programmer efficiency not equaled in the computer industry. The system is easy to learn (no horrendous educational costs), easy to modify (no large systems maintenance and programming staff is absolutely necessary as is with the other

vendor's systems), and the systems offer characteristics that carry forward into the operational stages, such as the interactive nature of terminals, efficient utilization of core through memory management techniques like shared processors, non-resident programming, overlaying and non-fragmentating map handling and others, such as centralized file management, which are too numerous to mention.

These are only a few of the characteristics that contribute to solving the implementation problem, but they represent current, day-now solutions to this greatest headache (and costs) and combine these solutions with the absolute price/performance advantages of the Sigma 9 and no other vendor has successfully accomplished this very competitive total solution to the problem.

-   <u>Evaluation and Operational Checkout</u>: This is the point where the top management of our customers get deeply involved. They are usually motivated by a deadline which must be met and a successful demonstration that the system is operating as designed by that particular deadline. What does the Sigma 9 offer in this area? First, the Sigma 9 was designed with a high level of reliability and self-checking built in. This means that the system will be <u>available</u> during the implementation and program checkout phases — thus <u>providing the other</u> necessary ingredient to successful program development — that of computer and system time! Thus, with that fear resolved, what about evaluation? Built into the Sigma 9 operating systems are accounting features and installation evaluations and control features that absolutely no other manufacturer is willing to put in, much less advertise. Not only are the control parameters there for variations and evaluations; they are there so that the system can recognize its own inefficiencies and modify itself, plus provide warning messages to prevent catastrophes!

• DECIDING WHAT SOFTWARE TO SELL

It must first be resolved amongst our own offering of software, which to offer what prospects and which one will really do his job — not the job he is doing now — but the job he wants to do!

The several that are offered are:

-   RBM
-   BPM - non-symbiont, symbiont
-   BTM
-   UTS
-   XOS

This is a complex problem in itself! Throughout this section, we are restricting ourselves to a combination of BTM/UTS or XOS. It is thought that BTM will sometimes

be desired (possibly for minimum configurations), but XOS will more often be required because of its outstanding features in a batch environment. Thus, you will usually decide between UTS and XOS or some combination of the two. For example, UTS may be operational during 8:00 a.m. to 5:00 p.m. and XOS operational otherwise.

The differences between UTS and XOS are differences of degree rather than of kind. UTS was primarily designed for time-sharing with the added uses of batch/remote batch and time-sharing. XOS was primarily designed for a highly efficient batch/ remote batch with added users of real time and time-sharing. Therefore, you will have to learn to very carefully define the user's requirements and problems before deciding which of these software systems to recommend.

- When to Sell UTS: Sell UTS for time-sharing (30-40 users and up) combined with:

    a. Batch - strong scientific
    b. Remote batch - strong scientific
    c. Real time - where real time or operator can give some advanced notice, such as a few hundred milliseconds or more. Once this is given, fast real time response can be provided

- When to Sell XOS: Sell XOS for general purpose batch, both scientific and commercial, combined with:

    a. Remote batch - scientific/commercial
    b. Time-sharing - 40, or fewer, terminals
    c. Real time - fast response within a batch environment

- SELLING SOFTWARE BASED ON NEEDS AND SOLUTIONS

Software competition is very complex as XDS knows from Sigma 5/6/7 experience. It becomes increasingly more complex as the systems get larger. This is because it is very difficult to evaluate software without a great deal of effort and the potential customer does not have the time and/or resources to devote to this effort (they are too busy using their present computer). This means that the decision to buy is usually made on fairly straightforward basis — price, benchmarks and maybe the fact that another customer is able to do the same thing (much reliance is placed on the proposal cost estimate, too). Price includes training and conversion costs, if any, in addition to systems and software costs.

To deliberately set about to completely tell the prospect "all about" UTS and XOS is a waste of time. By the time that was done, the prospect might be in the 5th generation of computers. To deliberately set about to learn what the prospect wants to do is the recommended way to go! As this is done, there is some software fallout to the prospect — he learns that certain of his requirements can be met. The whole idea here is that without knowing what he wants to do is like shooting at quail in the dark. You know they are there - you can hear them - you hope that when you shoot you hit one - but there is no aiming process. Software characteristics

must be aimed at a need! Without the need, there is just a very efficient mess of nothing! For example, a prospect says he wants time-sharing! Ah - a good time to give him your full UTS pitch! Afterwards, the prospect says "I was only talking about two or three inquiry terminals"! XDS has a tendency to talk too much about what it has and not listen to what the customer needs (wants?)! Sure, it makes a good story and makes you feel good because you are on firm ground, talking about something you know. But, are you solving the prospect's problem? NO! You're part of it.

Large, complex systems take 3 to 5 years to analyze and design. A couple of more years are spent implementing. The system should stay operational for another 3 to 5 years (maybe more). We are talking about a span of 8 to 12 years. The prospect is very serious about the success of the project. He has every right to be because he is going to live with it for a while. He has to be convinced that the XDS system will satisfy his needs. This cannot be done unless XDS understands his needs.

SIGMA 9 AS A TOTAL SYSTEM

Now that you have a better understanding of the importance of software, let us address the following topics:

Sigma 9 Hardware Overview

Some Examples of Sigma 9 Price/Performance

While the discussion of these topics will not give you a complete picture of the Sigma 9 as a total system, it should provide you with some of the more important insights into why you should consider this to be the case. Let's look at some of the capabilities that Sigma 9 offers that are not offered by any of the major competitors. These include:

- Three Software Systems: BTM, UTS and XOS — that provide three varying degrees of multi-use capabilities depending upon the prospect's needs and problems.

- An I/O and Real Time Structure unmatched by any other Sigma 9 competitor.

- Real Time Software unmatched by any other Sigma 9 competitor.

- Three FORTRAN Processors: Extended FORTRAN IV-H, Extended XDS FORTRAN IV, and FLAG, each with capabilities and debug facilities that are unmatched.

- A price tag competitive with anyones.

We've talked about software to some extent, so let's take a quick look at the Sigma 9 hardware.

● SIGMA 9 HARDWARE OVERVIEW

The Sigma 9 is a match or is superior to almost all the competitors you will be facing and this is particularly true in those multi-use applications where "real" real time and time-sharing are required, regardless of where you may find these prospects. On the next page you will find a configuration diagram of the Sigma 9. Compare this diagram with those of the other major competitors that you will find in the individual reports. Look at the I/O structure and the number of independent ports to memory.

Sigma 9 also has many reliability, serviceability and maintainability features built-in. IBM is perhaps the only competitor (with the 370/155) that can claim even 1/3 of these features, which are designed to insure that a Sigma 9 system stays up and running.

What about peripherals? No, Sigma does not have as extensive a range of peripherals as IBM (although to get the lower-priced of IBM's two fast RAD's; the 2305-2, you'd have to pay four times as much as for an XDS 7212). On the other hand, the range of Sigma peripherals compares very favorably with those offered by the rest of the competition. For example, Burroughs has no removable disk drives and to get UNIVAC's removable disk drive, one currently has to get another computer (UNIVAC 9000 series), program, incompatible (of course) with the UNIVAC 1106/1108.

By the way, who offers standard Systems Interface Units along with software besides XDS?

512 KB MIN | 128 KB INCREMENTS | 256 KB INCREMENTS

E F G H I J K L M

2-12
MEMORY
ACCESS
PORTS

CENTRAL
PROCESSING UNIT

MIOP
A | B

MIOP
A | B

MIOP
A | B

MIOP
A | B

MIOP
A | B

MIOP
A | B

MIOP
A | B

MIOP
A | B
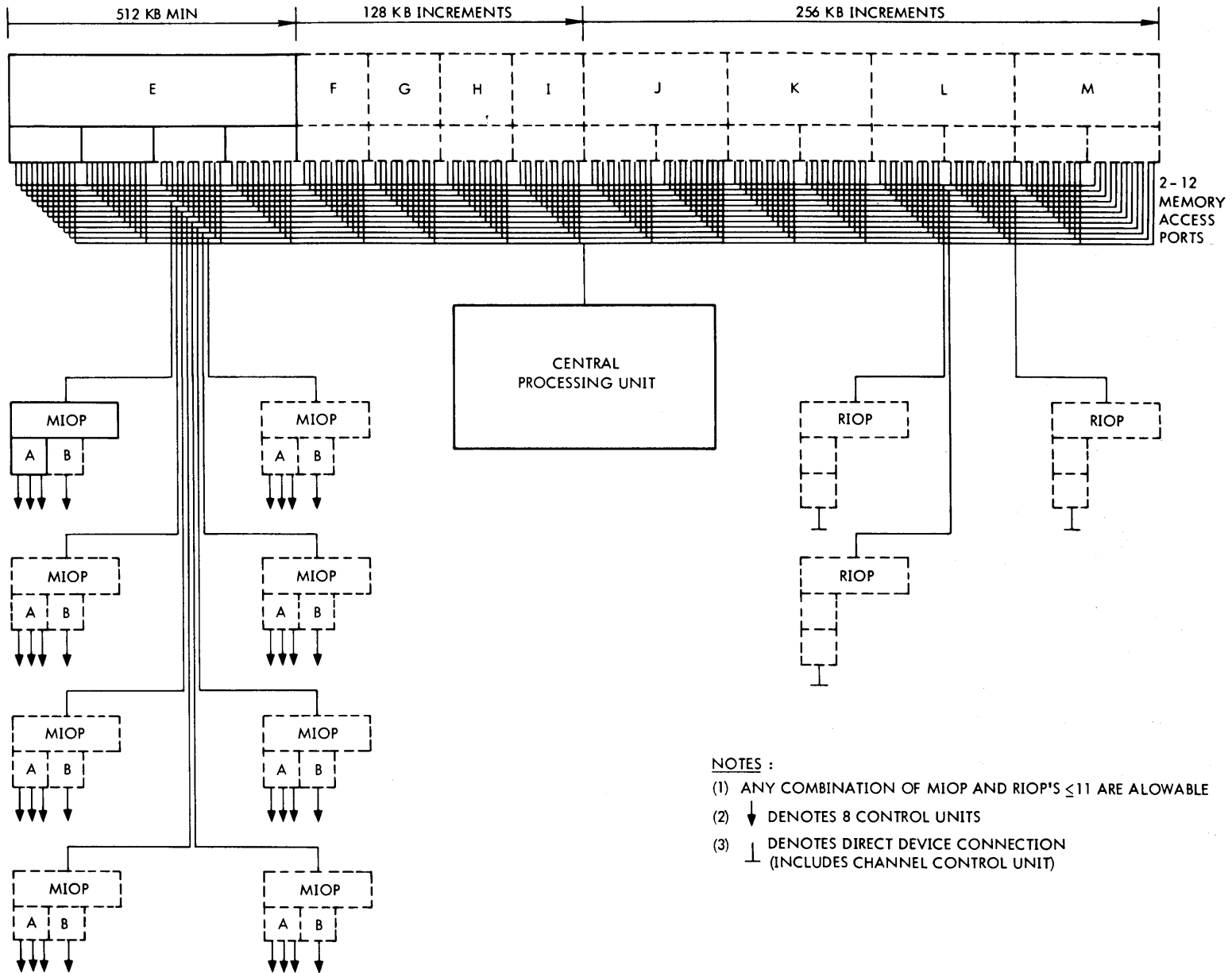
RIOP

RIOP

RIOP

NOTES :
(1) ANY COMBINATION OF MIOP AND RIOP'S ≤11 ARE ALOWABLE
(2) ↓ DENOTES 8 CONTROL UNITS
(3) ⊥ DENOTES DIRECT DEVICE CONNECTION
(INCLUDES CHANNEL CONTROL UNIT)

● EXAMPLES OF SIGMA 9 PRICE/PERFORMANCE

The Sigma 9 offers a very high level of performance, particularly in a predomi-
nantly scientific application, but also in a commercial environment where inter-
active time-sharing and remote batch are required. In fact, although the internal
commercial speed of the Sigma 9 is only about average for the machines in this class,
the I/O capability - which is at least as important in these applications - is
superior to all, and the XOS operating system provides performance superior to
IBM's OS MVT in the multi-use environment for which it is designed.

On the following page you will find a comparison of the internal operating speeds
of the Sigma 9 and our major competitors. Note that this chart reflects only raw
CPU speeds and does not take I/O, software effects, and CPU degradation due to
cycle-stealing I/O, such as is found on the IBM 370/155, into account.

INTERNAL SPEED RELATIVE TO SIGMA 9

COMPANY PRIVATE

16

```
5.0

4.5                                    X 6506

4.0

3.5                                    X 6504

3.0

2.5

            X 360-65
2.0
            X 370-155              · X 6503
1.5                      X 1108   X 6400              X 070/61/60
                         O 1108   X 6200
SIGMA 9  1.0  ⊙-6400    ⊙-6506    X 655
            O 360-65
            O 370-155   O 1106 MM  ⊙ 6200    ⊙ 6504    ⊙ 635
0.5         X 1106 MM             ⊙ 6503    X 635    O 70/61/60    O PDP10

            IBM      UNI      CDC      BUR      GE      RCA      DEC
```

O SCIENTIFIC

X COMMERCIAL
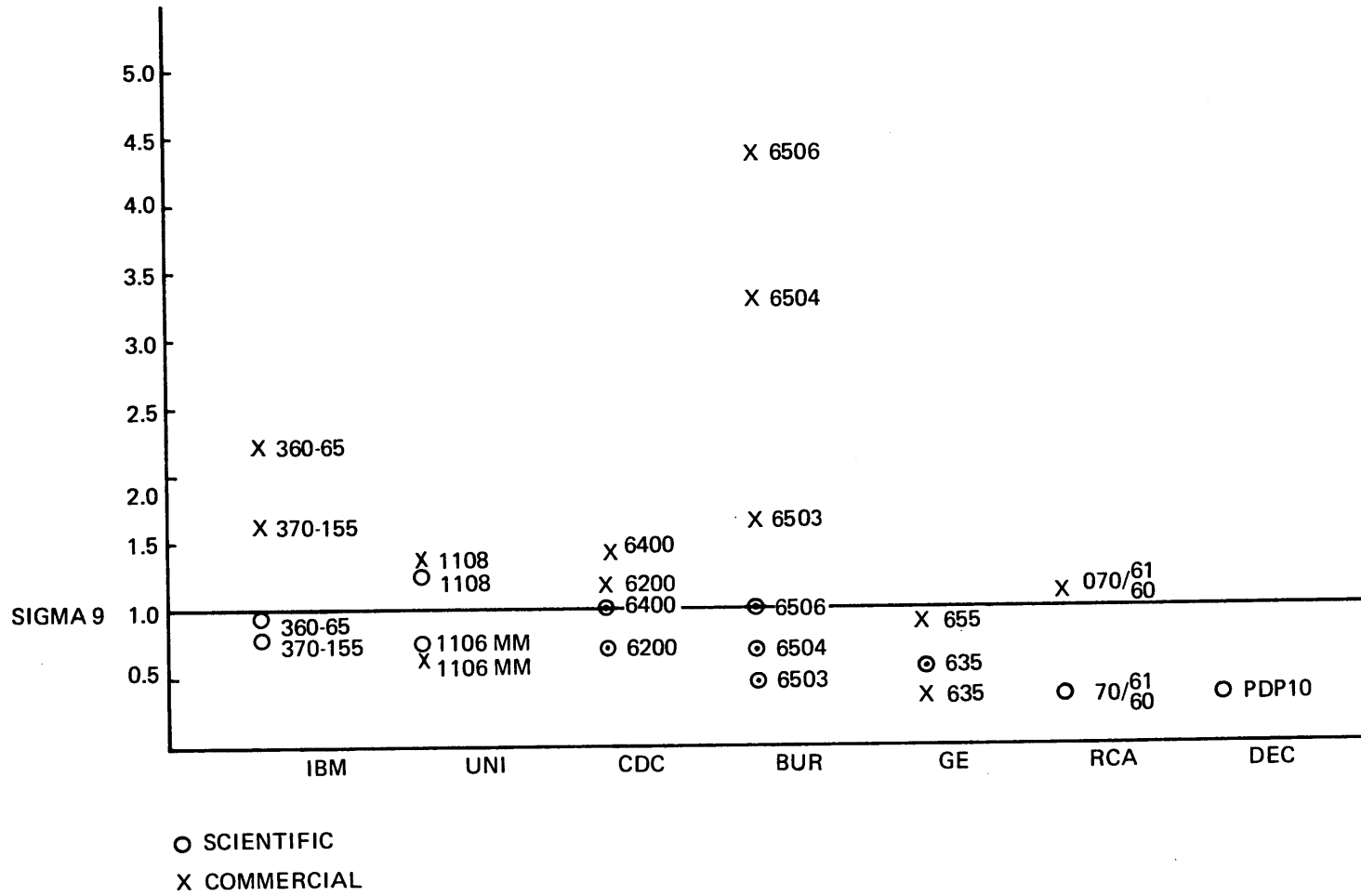
One of the key indicators of system price is the price of main memory. You will find on the following page a chart showing just how well the Sigma 9 memory price compares with that of the other competition. Not only is memory price and important indicator, but so also are the size of the memory increments a user needs in order to upgrade his system memory capacity. Smaller increments of memory are offered on the Sigma 9 because Sigma software uses memory more efficiently, therefore, the customer gets a double bonus – more usable memory to begin with and smaller incremental costs to increase his memory capacity.

AVERAGE COST PER CHARACTER (BYTE)

$ PER BYTE OR CHARACTER*

STORAGE CAPACITY
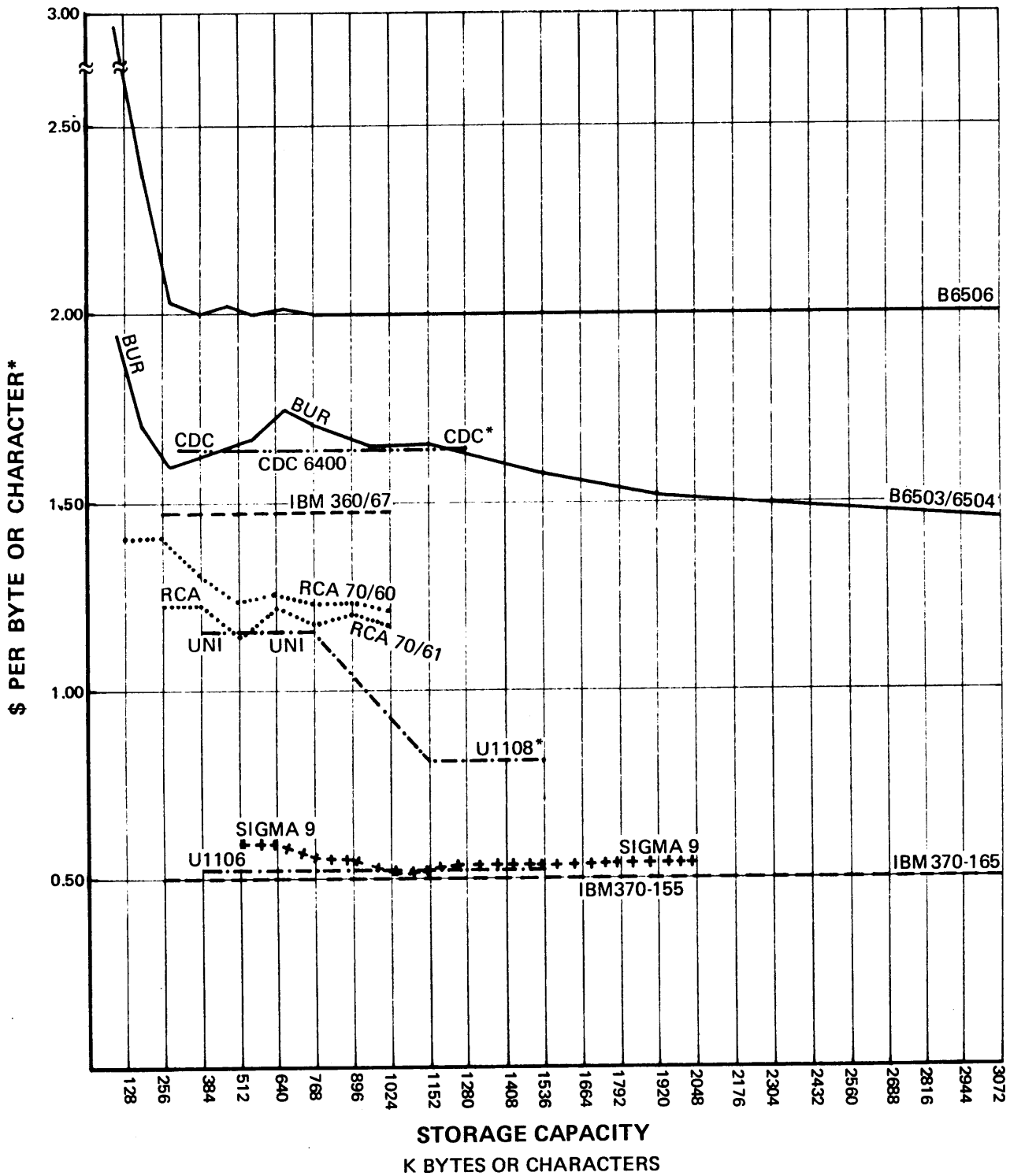
K BYTES OR CHARACTERS

The chart on the following page illustrates the increment sizes available on Sigma, compared with those available on the competitive systems.

AVAILABLE MAIN MEMORY SIZES
(8-BIT BYTES - KB OR 6-BIT CHARACTERS - KC*)

CDC 6200*	≈1.4 μS/48 BITS
BUR 6503	1.2 μS/48 BITS
RCA 7060	765 NS/64 BITS
RCA 7061	750 NS/64 BITS
CDC 6400*	1.0 μS/60 BITS
DEC PDP10*	1.0, 1.65, OR 1.8 μS/36 BITS
GE 635/655*	1.0 μS AND 500 NS/72 BITS
UNI 1108*	750 NS/36 BITS
UNI 1106*	1.5 μS/36 BITS
IBM 370155	2.1 μS/128 BITS
IBM 36067	750 NS/64 BITS
XDS Σ9	850 NS/32 BITS
BUR 6504/6506	1.2 μS AND 600 NS/48 BITS

(Memory size axis values: 128, 256, 384, 512, 640, 768, 896, 1024, 1152, 1280, 1408, 1536, 1664, 1792, 1920, 2048, 2176, 2304, 2432, 2560, 2688, 2816, 2944, 3072, 3200, 3328, 3456, 3584, 3712, 3840, 3968, 4096)

WHERE K = 1024

SOME SALES STRATEGY GUIDELINES

With the Sigma 9 there are two basic strategies to follow. These are:

- Consolidate several smaller systems into one Sigma 9 multi-use system
- Replace a smaller machine - usually the next smaller size - with a Sigma 9

Let's discuss each of these strategies briefly.

● CONSOLIDATION STRATEGIES

One of the best opportunities occurs when one of the systems at the prospective installation is a Sigma 5 or Sigma 7 and the other machines are significantly smaller than the currently-installed Sigma 5/7. In this situation, the DP management in control of the Sigma equipment probably has the most political and financial influence. The smaller machines may or may not be attached to the Sigma gear. If they are not, the controlling element may be small, scientifically-oriented, and both financially and politically weak within the organization (an example of such a situation would be a Hewlett-Packard system or an IBM 1800 installed in a laboratory environment) dedicated to some data acquisition and analysis project. With the real time capabilities and other capabilities of the Sigma 9, you can effectively counter any weak opposition from the users/management of the smaller systems while selling the powerful computation and cost-savings advantages of the Sigma 9 to the most influential group - those currently responsible for the Sigma 5/7 operations.

Furthermore, with the added capabilities of XOS, you also have a good chance to absorb an IBM 360/30 or other small 360's (360/20, 25) that are performing commercial work. To do this, you must be able to show the cost-savings that would accrue to the user and the benefits of consolidating all operations into a central system. Note that this represents a good way to attack IBM - not frontally, but on the flank. There are two keys to successfully using this strategy:

- The largest system currently installed should be Sigma and the Sigma management should have the political power within the organization

- The Sigma system currently installed should be substantially larger than any one of the IBM systems (meaning a 360/30 or smaller) and, if there is more than one IBM machine, the organizations controlling each should be separate in management and influence from the other. This way, you may be able to get one of these organizations to support the Sigma buy and, failing that, their individual weakness within the orgainzation may not enable them to generate sufficient opposition to the larger Sigma organization.

You must be aware, however, that IBM will try to use the same strategy with the 370/155. Presumably, the Sigma system was installed because there was a substantial amount of time-sharing and/or real time work. In order to make your consolidation strategy work, you will have to <u>protect</u> the currently installed Sigma equipment.

This is especially crucial if the installed IBM system is as large or larger than the installed Sigma system. This means that you must carefully manage your accounts and never miss an opportunity to extoll the virtues of Sigma for real time and time-sharing as well as scientific batch and commercial batch. When IBM tries to consolidate the systems, you should point out the inherent complexity of:

- Managing an IBM system in a commercial environment, let alone combining the two environments (OS requires heavy operator intervention and system management; the addition of even slow real time and scientific data acquisition makes this problem enormously complex)

- Conversion of the scientific/real time/time-sharing problems from Sigma to IBM, compounded by the very real possibility of having to convert upwards from DOS or OS MFT to OS MVT plus TSO plus RTM

Finally, if there is any critical real time application, it is very possible that IBM would not be able to perform at all within the response constraints and, if there is a substantial amount of high speed I/O (say data acquisition), it is possible that the IBM 370/155 (or 145) CPU performance would be severely degraded because of its cycle-stealing I/O.

Some other points about consolidation/centralization:

- Where IBM has the most influence and/or largest system at an installation, your attempt to consolidate all systems will probably not succeed. Try, instead, to consolidate all the installed scientific systems or upgrade/replace the currently installed Sigma system while protecting yourself from IBM's consolidation attempt.

- You can use this strategy against other competitors where there may not even be a Sigma computer installed as the largest system. However, you will have to be very careful when one or more of the smaller systems is an IBM system.

- The precise consolidation strategy you follow must be tailored to the situation. It is impossible to discuss each of the possible alternatives here.

- REPLACEMENT/UPGRADE STRATEGIES

With the Sigma 9, you have a very potent upgrade/replacement system because of its compatibility with smaller Sigma systems and its outstanding price/performance. In general, you should try to find opportunities where the currently installed machine has little or no compatibility with that manufacturer's next larger system. Some examples of currently-installed systems that provide good upgrade/replacement oppotunities include:

- CDC 3000 Series: These systems are old and have almost no compatibility with CDC's 6000 Series. Converting to Sigma will present little more diffi-

COMPANY PRIVATE

22

culty than converting to a CDC 6000 series system and present the prospect with many system capabilities at a lower price than CDC can offer.

- UNIVAC 490/494 Series: Virtually the same comments can be made about these systems as the CDC systems. The UNIVAC 1106 or 1108 is not upwards compatible from the 490/494.

- DEC PDP-10: There is no larger DEC system and hence no upwards compatibility. Furthermore, even if DEC did produce a larger system, past experience shows that DEC is in no way committed to upwards compatibility.

- Others: There are many other good upgrade/replacement situations that will arise but these do not generally present as advantageous or clear-cut a picture as the above. Each of these situations will have to be carefully qualified. However, currently-installed GE-400's will represent one of the best of these situations since we can provide a conversion aid, the TS-400 Simulator, and because there is only limited upwards compatibility to the very costly GE-600 series systems.

(CAUTION   —   IBM AHEAD!)

Finally, you should be very cautious about competing against IBM in a single system upgrade/replacement opportunity where the currently-installed system is an IBM 360/40, or larger, system. IBM currently provides too many upgrade paths for these users, will fight like H--- to protect these accounts, and provides an easy upgrade conversion for these users. The most important of these three items may well be the last. IBM users still carry many scars from their past conversion to 360. They are afraid to convert to any other system except under extreme duress. Since Sigma currently provides only minimal conversion tools to assist this user, any attempt to directly confront IBM is paramount to suicide in all but extremely exceptional circumstances. RCA's goal, for example, is to convert 10% of the 360/30 and below IBM users. They have no announced goal for larger IBM systems and the blood is still deep in the executive washrooms from their last attempt to so confront IBM in their stronghold. Even RCA's announced 10% IBM system conversion goal – with all their new conversion aids – is exceedingly optimistic. If RCA achieves 5%, it will probably cause chairman Sarnoff to buy a round of drinks for all the RCA field sales and marketing personnel.

In short, don't try to confront IBM head on!

- SUMMARY

There are far too many alternatives and strategies to outline here or anywhere else because of the complexities which are associated with each opportunity and each prospect's needs. It is felt that if you follow the general guidelines presented in this overview and in the other Sigma 9 Competitive Sales Guides, you will achieve the most success.

A BRIEF OVERVIEW OF XDS VS COMPETITION

The material presented in the charts which follow contain only some of the highlights of the competitor's strengths and Sigma 9 advantages. For additional information on the strengths and weaknesses of a competitive system, you should consult the appropriate Competitive Sales Guide on that competitor.

IBM 370/155 vs SIGMA 9

IBM 370/155 STRENGTHS	SIGMA 9 ADVANTAGES
OS MVT is available now	XOS provides more efficient resource allocation and scheduling
IBM provides easy upwards conversion for current IBM users	XOS allows more usable main storage in small configurations
IBM provides many diverse applications packages	XOS uses the Sigma hardware memory map for efficient storage management
The 370/155 has a buffer memory (status symbol)	UTS provides integral time-sharing and real time and supports more terminals
The 370/155 internal commercial capabilities are superior to Sigma 9's	UTS uses the Sigma hardware memory map
The 370/155 random access storage capabilities are superior (the IBM 2305-2 and the 3330)	XDS Extended FORTRAN IV is superior to IBM's FORTRAN — in both user features and debug capabilities
IBM offers higher quality line printers	XDS applications software, though more limited than IBM's, is superior to comparable IBM software, in most cases
IBM provides 1600 bpi magnetic tapes with speeds up to 320 KB	Sigma 9 internal scientific speed is superior to IBM's
IBM provides a wide variety of communications equipment	Sigma 9 has a hardware memory map and uses it!
IBM support (real or imagined)	Sigma 9 I/O structure (up to 12 ports to memory and independent IOP's) is superior and can support I/O rates far in excess of the 370/155
	Sigma 9 real time hardware and SIU's
	Sigma 9 prices are lower!

UNIVAC 1106/1108 STRENGTHS	SIGMA 9 ADVANTAGES
Exec 8 is proven	XOS has all the capability of Exec 8 plus real time and time-sharing
Exec 8 handles more programs in core	
Exec 8 provides extensive proven communications capability (software)	In a time-sharing environment, UTS handles 10 times the number of terminals that Exec 8 can support
Univac 1108 scientific processing speed is superior	Both XOS and UTS use the Sigma hardware memory map for efficient core management
Univac provides a wide range of communications products — DCT 500, 1000, 2000, etc.	XDS Extended FORTRAN IV is superior to Univac FORTRAN V, both in capabilities and in debug facilities
A slightly wider range of mass storage devices than XDS, including FASTRAND	Sigma internal processing speeds are superior to all below:
	a. Univac 1106 Scientific b. Univac 1106 Commercial c. Univac 1108 Commercial
	More ports to memory on Sigma 9
	32-bit word, 8-bit bytes versus Univac's 36-bit word, 6-bit bytes (ours is industry-standard)
	Sigma 9 reliability and availability features
	Sigma 9 offers upwards compatibility from lower numbered Sigmas, Univac 1106/1108 have no path from lower systems
	Sigma 9 prices are lower

CDC 6200/6400 STRENGTHS	SIGMA 9 ADVANTAGES
SCOPE provides proven batch multiprogramming	XOS offers superior facilities over SCOPE, including time-sharing and real time
KRONOS will support (CDC claims) up to 384 active terminal users and concurrent batch multiprogramming	UTS and XOS time-sharing languages are fully conversational, KRONOS languages most likely are not
CDC provides ALGOL as a standard language	XDS time-sharing experience supports XOS and UTS. CDC has no past experience
CDC 6400 provides fast internal scientific processing	XDS commercial capabilities, both hardware and software, are superior to CDC's (CDC does not even have decimal hardware)
CDC has bulk secondary core storage (option)	
CDC IOP's (PPU's) are intelligent	Sigma 9 internal scientific performance is superior to the CDC 6200
	Sigma 9 has hardware real time capability, the CDC 6200/6400 have virtually none
	Sigma 9 prices are low compared to either CDC system

BURROUGHS 6500 SERIES STRENGTHS	SIGMA 9 ADVANTAGES
MCP provides proven batch multiprogramming	XOS and UTS both provide time-sharing and real time capabilities, MCP has neither
Burroughs has ALGOL as a standard product	UTS provides extensive systems management and accounting statistics as well as rate/charge structure
Good communications capabilities in both hardware and software	
Burrough's efficient internal CPU structure uses dynamic, automatic stack concept	Internal scientific processing of Sigma 9 is superior to B6503 and B6504. Be sure you know which 6500 Series machine you are competing against!
Superior internal scientific processing speeds on the B6506 (only)	Sigma 9 has extensive reliability, maintainability, and availability features — Burroughs does not even have power fail-safe
	Sigma 9 has high speed I/O rates, up to 6MB/ memory access port. The highest transfer rate device Burroughs supports is 395KB
	Sigma 9 hardware real time is far superior. Burroughs has practically none
	Sigma 9 prices are lower

RCA STRENGTHS	SIGMA 9 ADVANTAGES
TDOS is a proven multiprogramming system	XOS provides time-sharing and real time capability not found in TDOS (OS/61 has no real time but does provide time-sharing)
OS/61 has very good facilities for time-sharing users	
RCA uses the virtual memory concept to provide addressable program/data storage on secondary storage devices	UTS offers extensive performance management and accounting statistics not found in either of the RCA systems
RCA provides extensive IBM compatibility	XDS FORTRAN and FLAG languages are superior to RCA's
RCA offers an extensive range of commercial applications software	Sigma 9 internal scientific CPU speeds have a big edge over the RCA 70/61 and RCA 7
RCA has extensive communications capabilities, hardware/software (new front-end communications processor on RCA 2/3/6/7)	Sigma has a proven, accepted time-sharing reputation. RCA has encountered significant hardware/software problems with the 70/46 and is just beginning to install 70/61's
RCA has higher speed magnetic tape drives	Sigma 9 reliability, maintainability and availability features
New prices on RCA 6/7 are attractive but not necessarily lower than Sigmas	Sigma multi-port memory
RCA's new educational discounts (up to 30%) are attractive	Sigma real time hardware and priority interrupt features
	Sigma SIU's
	Sigma's approach to program-swapping is well-proven. RCA's demand-paging (page-swapping) generates high overhead and results in low response in a short job environment (OS/61 and new VMOS)
	Sigma RAD's provide better performance than RCA swapping devices
	Sigma prices are lower than older Spectra series systems and should be at least as low as new RCA 6/7 prices

GE 600 SERIES STRENGTHS	SIGMA 9 ADVANTAGES
Proven multiprogramming Applications software Good communications capabilities GE has good time-sharing reputation	XOS and UTS both provide real time as well as time-sharing UTS time-sharing capabilities (languages, facilities, etc.) are superior to those offered under GECOS III Performance management and accounting under UTS XDS Extended FORTRAN IV is superior to GE's FORTRAN XDS has FLAG, GE does not Sigma 9 internal processing speeds superior to all GE 600's except GE-655 Sigma 9 has decimal instructions. GE does not Sigma 9 reliability, maintainability and availability features XDS RAD's are superior to GE swapping devices Sigma 9 real time hardware and SIU's Sigma 9 prices are much lower than GE's

DEC STRENGTHS	SIGMA 9 ADVANTAGES
Mature and operational time-sharing operating system	Multi-use capability with outstanding batch stream
Good FORTRAN code generation for large instruction set	Superior batch capabilities under either UTS or XOS. DEC has poor batch
Word size of 36 bits vs 32 bits on Sigma 9. Provide 8 decimal digits of significance compared to between 6 and 7 on Sigma 7.	Strong XDS supported applications packages for Data Management, linear programming, simulation (analog and discrete)
ASCII capability - internal and on printers	XDS supports commercial applications. DEC does not!
DEC equipment is traditionally low priced due to lower performance and inadequate capability	Number of time-sharing terminals supported. UTS supports up to 128 while the PDP-10 is pushed to support 30 to 35
	Performance - Sigma 9 is 2 times the PDP-10 for scientific work
	Sigma 9 following hardware features not available on PDP-10
	a. Memory map b. Double precision instructions c. Decimal instructions d. Watchdog timer e. And many more
	Removable disk pack support
	XDS provides standard leasing policies with full nationwide maintenance support

Competitive Sales Guide

OCTOBER 1970   COMPETITIVE ANALYSIS   File: IBM
370/155
360/67

Contained in this Sales Guide is an assessment of the IBM 370/155 and the 360/67 computer systems versus Sigma 9. You will not often encounter the 360/67 since it has been placed in a "restricted marketing" class by IBM. However, if backed into a corner, IBM may propose a 360/67 to compete against a Sigma 9 UTS system rather than the 370/155 with OS/MVT-TSO.

CONTENTS

Please refer to the Sigma 9 Overview Sales Guide (File: XDS) to compare the configuration diagrams of the IBM 370/155 and 360/67 to Sigma 9, and also refer to the other Sales Guides on Sigma 9 competitors for additional information.

Distribution: Sigma 9 Announcement
Distribution and Competitive Infor-
mation Reference Libraries

Kent L. Cootes, Manager
Competitive Analysis
XDS   AFC

Xerox Data Systems

SYSTEM(S) SUMMARY

IBM's System/360 announced in April of 1964 and, more recently, the System/370 announced in July 1970, are two families of general purpose (general batch) computers. With the exception of the 360/44 for real time and the 360/67 for time-sharing, IBM's 360 and now the 370 have not attacked user environments where special function mainframe hardware has been a key factor.

- INTRODUCTION — 360/155

  The System/370 is an extension of IBM's System 360 family of general purpose (batch processing) computers.

  The System/370 represents a new, more competitive series of computers, both from an extended capability and a price/performance standpoint. The 370/155 is a medium to large sized (512KB - 2MB) system. The 370/155 obtains much of its speed from its high speed CPU buffer storage. The buffer storage is used to isolate, partially, CPU memory accessing from that of the I/O channels, thus creating an effect similar to that of Sigma's multiple ports to memory.

  IBM markets the 370/155 as the "natural growth system for its current 360/40 and 360/50 users. Because of larger core capacity and channel bandwidth, the 370/155 is sure to become IBM's entry point into the full service time-sharing market under TSO/OS MVT.

  IBM indicated with its 370 announcement that it intended to extend its most powerful operating system — OS MVT — to include a real time monitor capability (refer to IBM 370 CSG). Initial customer deliveries will start in April 1971.

- IBM 370/155 STRENGTHS

  - Compatibility: The 370/155 is compatible with System/360 and has a concurrent multiprogramming of DOS, 1400, and 7010 emulations under OS MVT.

  - Commercial Performance: The 370/155 has better commercial execution compute power (108.2 KIPS to 64.7 KIPS). This ratio is based on ideal conditions for their execution out of the CPU buffer memory and minimal cycle-stealing channel loading. The 370/155 obtains part of this advantage because of its ability to perform storage-to-storage operations.

  - Extended Precision Floating Point: The 370/155 can obtain greater precision (up to 34 decimal digits) with their quadruple or extended precision feature.

  - Extensive Market Penetration: In most cases IBM is the industry leader, especially in commercial application areas. IBM has an extensive line of special purpose peripherals for various commercial data processing applications.

  - The newness of the 370/155

- IBM 370/155 WEAKNESSES

    - Scientific Processing: The 370/155's scientific processing internal compute power is less than Sigma 9's (636 KIPS:763 KIPS). Because the 370/155's channels are cycle-stealing, they can degrade the 370/155's performance significantly under heavy loads.

       The nature of many scientific problems; i.e., matrix inversion, is such that the 370/155's CPU buffer memory will be hard pressed to contain the necessary data to permit efficient execution operations. All in all, it should be noted that the above 370/155 KIP rate is a best-case condition, and it is doubtful that the 370/155 will perform up to the rates designated.

    - Time-Sharing: The 370/155 does not have the necessary hardware (core management/map, multiple register blocks, low CPU load I/O processors, etc.), in order to perform efficiently in a time-sharing environment.

    - Real Time Software: IBM's new Real Time Monitor (RTM) extensions to OS MVT is difficult to program, may be quite expensive, exposes batch programs to danger, complicates system management, may cause inaccurate resource accounting, is based on a poor design philosophy, and requires a very large system OS MVT.

    - Price: The 370/155 is a very expensive system ($37K - $79K).

    - Real Time Hardware: The IBM 370/155 lacks these important features — stack instructions for efficient writing of reentrant code (note less than total memory required), multiple general purpose registers for rapid context switching, integral high priority interrupt structure, associative memory map, etc.

HARDWARE

- IBM 370/155 HARDWARE HIGHLIGHTS

   The System/370 — Model 155 — data processing system is aimed at satisfying the following major customer requirements:

    - Compatibility

    - Performance

    - Reliability, Availability, and Serviceability

    - Extended Capabilities

- COMPATIBILITY

  - Extensive 360 compatibility

  - Integrated Simulation

In response to constant demands by its users, trade associations, and governmental agencies, IBM will remain upward compatible. The key word "upward" allows IBM to continue its one-ups-manship game however, as demonstrated in the plug-to-plug compatible peripheral market.

By the 370 being compatible, IBM may make use of its sizeable software investment and avoid bucking its own 360 momentum. Because the 370 is upward compatible with the 360, it can run virtually all non machine-dependent 360 software. Through its special emulation features and under OS MVT the Model 155 increases 360 compatibility by concurrently multiprogramming under one operating system:

    Normal 360 OS jobs

    DOS jobs

    1400 and 7010 jobs

Thus, scheduling problems, etc., previously associated with stand-alone emulation are eliminated.

The most significant characteristic of the System/370 announcement is the establishment of a more competitive set of price/performance curves. IBM all but eliminated its 360/65 with the introduction of the 370/155, the exception being the unavailability of a 7090 series emulator on the 370/155.

IBM, having learned their lesson in 1964 about ir compatibility with the System/360, strenuously characterizes the System/370 as an extension of the System/360 family. For instance, the 370 instruction set is the same as the 360's with some minor enhancements. These enhancements include the ability to monitor machine and program product use by having the CPU serial number in the CPU for verification by a control program.

- PERFORMANCE AND CAPACITY

  - Estimated potential compute power
      Commercial   —   108 KIPS/90% hit ratio
      Scientific   —   636 KIPS/90% hit ratio

  - Main memory - 256KB to 2MB, 2.1 µs/16B

- Buffer memory – 8KB, 230 ns access/4B
- CPU memory access partial isolation
- CPU – 115 ns cycle/4B – Monolithic ROS – 69 ns
- I/O Channels – 5.8MB aggregate, 1 – 2* MPX and 2 – 5* BLK
- Swapping Disk – can handle one 2305-2 fixed head disk, 1.5MB transfer, 5MS average access
- Bulk High Speed Store – IBM 3330 Disk Storage Unit, 100MB capacity/drive packaged two spindles per unit, 806KB transfer rate, 30MS average seek access, 8.3MS average latency
- 2000 LPM printer – IBM 3211 Printer program carriage control
- High Speed Console – IBM 3215 console printer/keyboard (85 CPS)

Based on design specifications, the IBM 370/155 should perform very well whenever it has sufficient core to take advantage of its multiprogramming operating systems. It is even quite possible that the 155, when configured with its newer high speed peripherals, could reach IBM's goal of 3 or 4 to one compute power advantage over the 360/50.

Even though the 370/155 has only integral I'O channels, like its 360/50 brother, its performance is greatly enhanced by employing the CPU buffer storage concept. The buffer storage partially isolates CPU memory accessing from that of the I/O channels, thus reducing contention similar to Sigma 9's multiple ports on multiple memory modules. However, when multiprogramming several programs, or when any given program instruction or data accessing bounces around to any great degree, this isolation effect is defeated and, in addition, the execution speed would be greatly degraded. Of course, the traditional argument against cycle-stealing channels is still valid. Note that even IBM recognizes this problem and has independent I/O processors on its supposed "leading edge technology" machines – the 360/75 through 360/195. Total system bandwidth may be important to your prospect so, here again, Sigma's faster independent I/O processors (3.0MB to 1.5MB max), multiple memory modules with multiple ports has quite an impressive advantage (25.3MB: 5.8MB).

- RELIABILITY
    - Reliability, Availability and Serviceability (RAS)
    - Error Check and Correction Code (ECC)
    - Instruction Retry
    - Channel Retry

Some features included in the 370 announcement are, in reality, the restatement of

---

* Six channels maximum.

features originally stated for inclusion in the 360; i.e., Recoverability, Availability and Serviceability (RAS). RAS is a marketing slogan for some features that are "state-of-the-art" technology for proper system functioning. Because a system of this size and expense requires that no precious time be lost to minor malfunctions, every effort is made as is the case with Sigma 9 to, at worst case, "fail soft". ECC, instruction retry, and channel retry are some of the specific techniques to which IBM so beautifully attaches new acronyms or terminology.

- EXTENDED CAPABILITIES

    - Extended Precision Floating Point: IBM, seeking to regain some of its scientific business lost to CDC's 6000 series big word machines, has extended its floating point precision capability. This new extended precision more than doubles double precision by providing an additional two words (64 bits) of precision.

- INTRODUCTION — 360/67

IBM's most complete time-sharing system prior to 360 TSO/OS MVT had been the System/360 — Model 67 (360/67) — and its Time-Sharing System (TSS) monitor. The 360/67 was announced in April 1965 as a non-standard system (available only upon special request). In August 1965 the 360/67 became a standard IBM product only to be reclassified as available only under a "controlled marketing" basis in January 1967, due to unforeseen system design and development problems. Currently, it appears as if the 360/67 is still available only under the controlled marketing restriction, whereby, regional management must sign off on its feasibility (systems assurance). However, IBM has had its 360/67 TSS monitor and language processors completely refurbished within the last year and may be ready to try again in the use of a totally integrated time-sharing system (virtual memory, reentrant processors, operating system environmental controls, etc.)

Effectively, IBM is claiming that it is too early to tell how time-sharing systems ought to be designed and, furthermore, that they recognize the need to have their own experimental machine - 360/67. Because IBM has made several wrong turns in its 360/67 TSS development and has not met any of its performance goals, it becomes increasingly evident that in order to market this system successfully, a repackaging and/or renaming of these items would be in order. Because it can default to a 360/65 mode of operation, IBM attempts to sell the 360/67 as a standard batch processor with time-sharing hardware and software extensions.

Unique to the 360/67 is its Cambridge Monitor System (CMS). CMS has the capability to make several different terminals each appear (virtual) as if each is the systems console and time-sharing data entry device for almost any IBM operating system (including OS). One such operating system, which runs under CMS is Control Program-67 (CP/67). CP/67 is a time-sharing monitor that, when run in conjunction with CMS, provides the facilities to create, debug and execute IBM's OS Assembler (F-level), FORTRAN (G-level) and PL/I (F-level). CMS is not a Type I or standard IBM operating system.

The only differences between the 360/67 and the 360/65 are:

- Virtual memory addressing scheme and an eight-register associative memory for virtual-to-physical transaction purposes

- Dual processor CPU options

- 360/67 MARKETING STRATEGIES (TSS)

  - Time-Sharing: This system is aimed at large universities and large corporate organizations. Where an IBM customer absolutely states that he wants a large time-sharing capability, this system may be proposed if the customer will not wait for the TSO option under OS MVT.

  - Batch/Remote Batch Multiprogramming: Concurrently with doing time-sharing, TSS has all the flavor of IBM's operating system for the 360.

- 360/67 STRONGPOINTS (TSS)

  - Time-Sharing: Conversational FORTRAN IV, Assembly language, and PL/I are offered with a command language that provides terminal job entry and interactive debugging facilities. The virtual memory concept allows each program to directly address over 16MB of storage.

  - Batch: TSS provides compatibility for IBM 360 programs which include COBOL, FORTRAN, ALGOL, PL/I, and the other applications programs.

- IBM 360/67 WEAKNESSES

  IBM's problems with the 360/67 are tied to the following factors:

  - Virtual Memory/Demand Paging: Algorithms did not/do not exist for efficient CPU utilization and I/O structures. Even with multi-ports, high speed drum for paging, address translation (associative memory) etc., the scheduling problems are overwhelming and saturate the system with overhead.

  - Reentrant Programs: Because the 360/67 and all other 360 systems lack the stack instructions, the normal use of the fixed savearea for general registers does not permit routines to be reentered prior to their completion without excessive software overhead.

  - Context Switching: System/360's, and especially the 360/67, have only one set of general purpose registers; thus, every time a context switch is made between the monitor and user programs, or between user programs, or when branching to subroutines, these registers must be saved. The use of multiple blocks of general purpose registers eliminates the need to save/restore these registers.

  - Over-Specification: The 360 was facing serious competition from the GE 600 Series and from CDC's 6600. The initial 360's in this range, namely the Models 60, 62, 64, and even the Model 70, were all respecified, re-engineered and ultimately emerged as two Models — 65 and 75 — with much added capability. The Model 67 was a direct off-shoot of this re-everything.

  - TSS - Time-Sharing Systems: This system is one of a kind and represents a dedicated effort on the part of IBM to fail - softly! As a part of the original TSS de-committment, IBM renegotiated with each customer and bought them off by pledging on-site systems assistance plus a large devoted group programmer to rejuvenate TSS.

- 360/67 WEAKNESSES (TSS)

  - Demand Paging: TSS uses demand paging for bringing programs and/or program pages into and out of memory. This means that a program can refer to any location (address) in its virtual memory space and if the page where that location exists is not in core, the system will go get it, place it in actual memory, and then continue execution.

    This method creates I/O more or less for the sake of I/O and burdens the system with additional overhead.

  - Technical Difficulties: The implementation of TSS has been plagued with technical difficulties primarily due to the demand paging as discussed above. Other difficulties in the area of developing reentrant conversational compilers caused the 360/67 to be placed in a "controlled marketing" status and

much of its original, specified capability to be "de-committed". Thus, the 360/67 will seldom be encountered in a competitive situation.

- SIGMA 9 XDS STRATEGY — GENERAL

  - Time-Sharing - UTS: Stress the time-sharing scheduling methods that are integral to UTS. The fact that a deliberate attempt is made to keep 3 or 4 compute bound users in core so that the CPU can be better utilized.

    Stress the performance management reports that are instantly available and accumulative over a period of time under UTS. TSS installation managers cannot tell whether their system is running well or not.

    Stress the operator control parameters. These parameters are available to tune the system during operation. TSS does not offer these parameters.

    Stress the additional language facility of BASIC and other services.

- SIGMA 9 STRENGTHS — UTS

  - Memory Mapping Management: TSS does this under the demand paging method. The difference is that UTS has designed into the system the provision for fast response to those terminals which need fast response. At the same time, UTS provides installation controlled resource allocation to compute bound users and to the batch stream.

- SIGMA 9 STRENGTHS — XOS

  Since TSS is fundamentally for time-sharing, XOS strengths relative to batch will not be discussed.

256 KB MIN | 128 KB INCRE. | 256 KB INCREMENTS | 512 KB INCREMENTS

| H | HG | I | IH | J | JI | K |

BYTE MPX | 8 KB BUFFER | BLOCK MPX

| 1 | 2 |

| 1 | 2 | 3 | 4 | 5 |
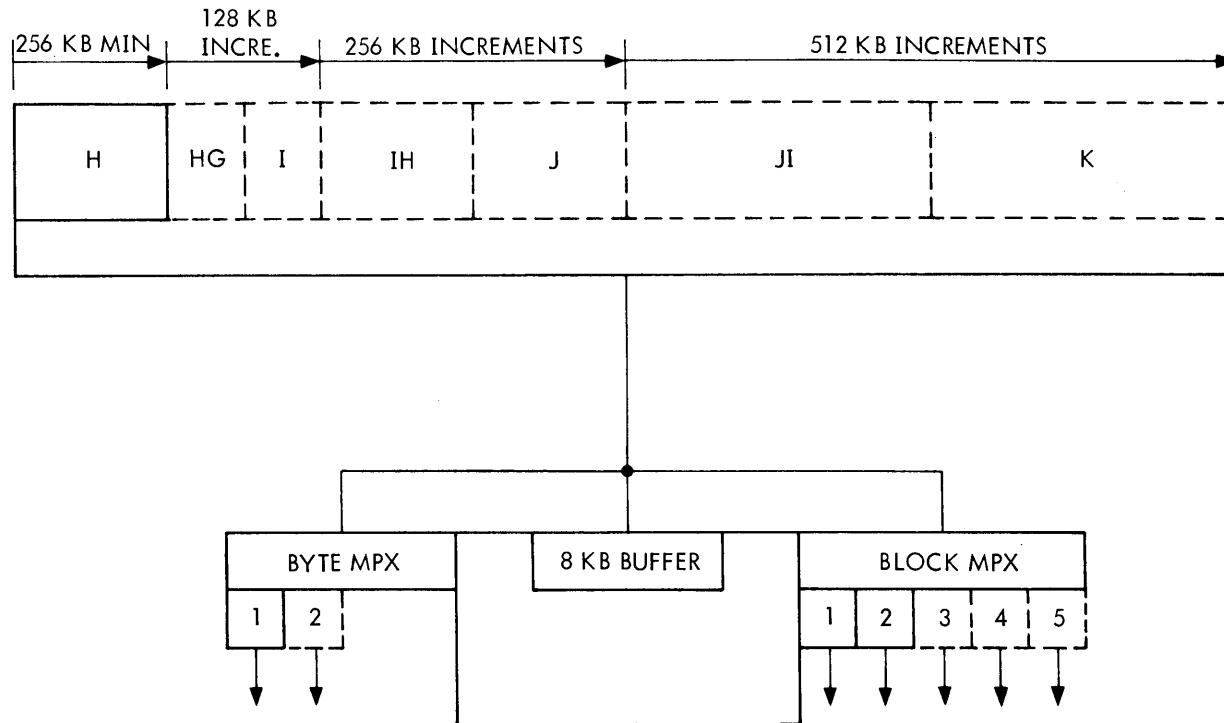
NOTES : (1) MAX.6 CHANNELS - SECOND BYTE MPX OR
FOURTH BLOCK MPX

(2) MAX.BLOCK MPX CHANNEL TRANSFER
RATE - 1.5 MB

IBM 370/155

DETAILED HARDWARE SPECIFICATION SHEET

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: SYSTEM/370

SYSTEM CHARACTERISTIC	MODEL 145	MODEL 155	XDS SIGMA 9
● MAIN MEMORY			
Size Range	112-512 KB	256K - 2M Bytes	512K - 2M Bytes
Cycle Time	Fetch 540NS Store 607.5NS	2.1 μs	850 NS
Access Width	8 Bytes	16 Bytes	4 Bytes
Interleave	NA	NA	4 Way
Access Ports	1	1	2 - 12
Map	NA	NA	Std.
Protection	Std.	Std.	1st 512KB Only
Min. Increase	48KB	128KB	128KB
● BUFFER MEMORY	NA ↓		NA ↓
Size Range		8KB	
Cycle Time		230NS	
Access Width		4 Bytes	
Interleave		2 Way ?	
Access Ports		2	
Protection		?	
● LARGE CAPACITY STORAGE	NA ↓	NA ↓	NA ↓
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Xfer Rate			
Map			
Protection			
Exec. Instr. From			
● CENTRAL PROCESSING UNIT			
CPU Clock Cycle	202.5 - 315 NS	115 NS	Instr. & Seq. Dep.
Look Ahead			
Instr. Prefetch	8 Bytes Less Current Instr.		12 Bytes Less Current Instr.
Levels of Decode	Current Instr.		One Beyond Current

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: SYSTEM/370

SYSTEM CHARACTERISTIC	MODEL 145	MODEL 155	XDS SIGMA 9
● INSTRUCTIONS			
Commercial KIPS	85.8	108.2	64.7
Scientific KIPS	330.9	636.1	763.4
Word Size	32 Bits or 4 Bytes	⟶	32 Bits or 4 Bytes
Instruction Size	2, 4, & 8 Bytes	⟶	2 or 4 Bytes
Mult. Addr. Ops.	0, 1, or 2	⟶	
Decimal	STD	⟶	Std.
Fixed Point	STD	⟶	Std.
Floating Point	Sing., Dbl. & Quad	⟶	Sing. & Dbl. Std
Stor.-to-Stor.	STD	⟶	NA
Byte/Char. Manip.	STD	⟶	Yes
Stack	NA	⟶	Yes
● ADDRESSING			
Indexing	Yes	⟶	Yes
Indirect	NA	⟶	Yes
Virtual	NA	⟶	Map - 512KB/Prog.
Page Size			512 words
No. of Pages	↓		256
Scheme	Base Displacement	⟶	Direct Reference
Size	1, 2, 4 or 8 Bytes	⟶	1, 2, 4 & 8 Byte
Max. Addr.	16MB	⟶	4MW or 16MB/ Extended Mode
● REGISTERS			
Index	Gen. Purpose	⟶	Gen. Purpose
Gen. Purpose	16	⟶	2 Blks. of 16 Std.
Mult. Gen. Reg. Blks.	NA	⟶	2 Std. & Opt.
Floating Point	Sing., Dbl. & Quad	⟶	Single & Double
Decimal	STD	⟶	Std.
● INTERVAL TIMER	3.33MS & "Time-of-Day" Clock - 1 μs	⟶	500 HZ & one of (50/60, 500, 2K, or 8K HZ) Switchable

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: SYSTEM/370

SYSTEM CHARACTERISTIC	MODEL 145	MODEL 155	XDS SIGMA 9
● INTEGRAL MULTIPLEXER	Byte Multiplexer	⟶	NA
Channels	1	1-2	
Subchans./Controllers	8/Channel	⟶	
Aggregate Xfer	?		
MPX - Max Chan Xfer	?		
Burst - Max Chan Xfer	?		
Command Chaining	Yes	⟶	
Data Chaining	Yes	⟶	
Data Path Width	1 Byte	⟶	
CPU Loading			
Multiplexing	14.3μs/Byte	?	
Burst	5.3μs/Byte	?	
Command Chain	19.2μs/Burst, 31.0μs/Byte		
Data Chain	6.25μs/Chain		
Xfer in Channel	2.0 μs/TIC		
● EXTERNAL MULTIPLEXER	NA	NA	1 MIOP Std 11 Max
Channels			1 Std. 2 Max.
Subchans./Controllers			8-32
Aggregate Xfer			1.9MB W/4 Byte & 2 Chan.
			470KB
MPX - Max Chan Xfer			
Burst - Max Chan Xfer			
Command Chaining			Yes
Data Chaining			Yes
Data Path Width			1 or 4 Byte/Device & 4 Byte to Mem.
● INTEGRAL SELECTOR	Selector & Blk MPX	Block Multiplexer	NA
Channels	1-4 or 1-2 w/Integ File	2-5 or 2-4 w/2nd MPX	
Subchan./Controllers	8-16	⟶	
Max. Xfer	820KB or 1.85MB w/chan word buffer	⟶	
Command Chaining	Yes	⟶	
Data Chaining	Yes	⟶	
Data Path Width	Byte or word w/ chan word buffer	⟶	
CPU Loading			
Xfer.	?	?	
Command Chain	?	?	
Data Chain	?	?	

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: SYSTEM/370

SYSTEM CHARACTERISTIC	MODEL 145	MODEL 155	XDS SIGMA 9
● EXTERNAL SELECTOR	NA	NA	1 - 4 RIOP's
Channels			1
Subchans./Controllers			1 Integral
Max. Xfer.			3MB
Command Chaining			Yes
Data Chaining			Yes
Data Path Width	↓	↓	4 Byte to Mem.
● SYSTEM BANDWIDTH	5.3MB	5.8MB	25.3MB
● AVAILABILITY			
Watchdog Timer	NA	⟶	Yes
Power Fail Safe	Manual Restart Required	⟶	Saves Volatile Regs. & Brings to Orderly Halt
Recoverability	Yes	⟶	Yes
Relocatible Homespace	NA	⟶	Yes
Instruction Failure			
Pre. Exec. Test	Yes	⟶	Yes
Isolation	?	?	Yes
Retry			
CPU	Yes	⟶	Yes
Channel	Yes	⟶	Yes

• IBM 360/67 HARDWARE HIGHLIGHTS

As mentioned, the time-sharing exception to the 360 standardized philosophy is the Model 67. The 360/67 employs a limited (only eight mapping registers) virtual memory scheme and has a dual CPU multi-processing capability.

The 360/67 functional system diagram and detailed hardware specification sheets contain the same information about standard 360/65 or 67 characteristics as has been previously supplied to your various district offices; i.e., Auerbach Reports and other Competitive Sales Guides.

64 KW OR 256K BYTE
MINIMUM AND INCREMENTS

| MAIN MEMORY | MAIN MEMORY | MAIN MEMORY | MAIN MEMORY |

SINGLE PORT TO MEMORY

CENTRAL PROCESSING UNIT

MULTIPLEXER   MULTIPLEXER

| M1 | S1 | S2 | S3 | M2 | S1 | S2 | S3 |

STD. MPX CHANNEL

SELECTOR   SELECTOR

| 1 | 2 | 3 | 1 | 2 | 3 |

1.3 MS MAX DATA XFER RATE/CHANNEL
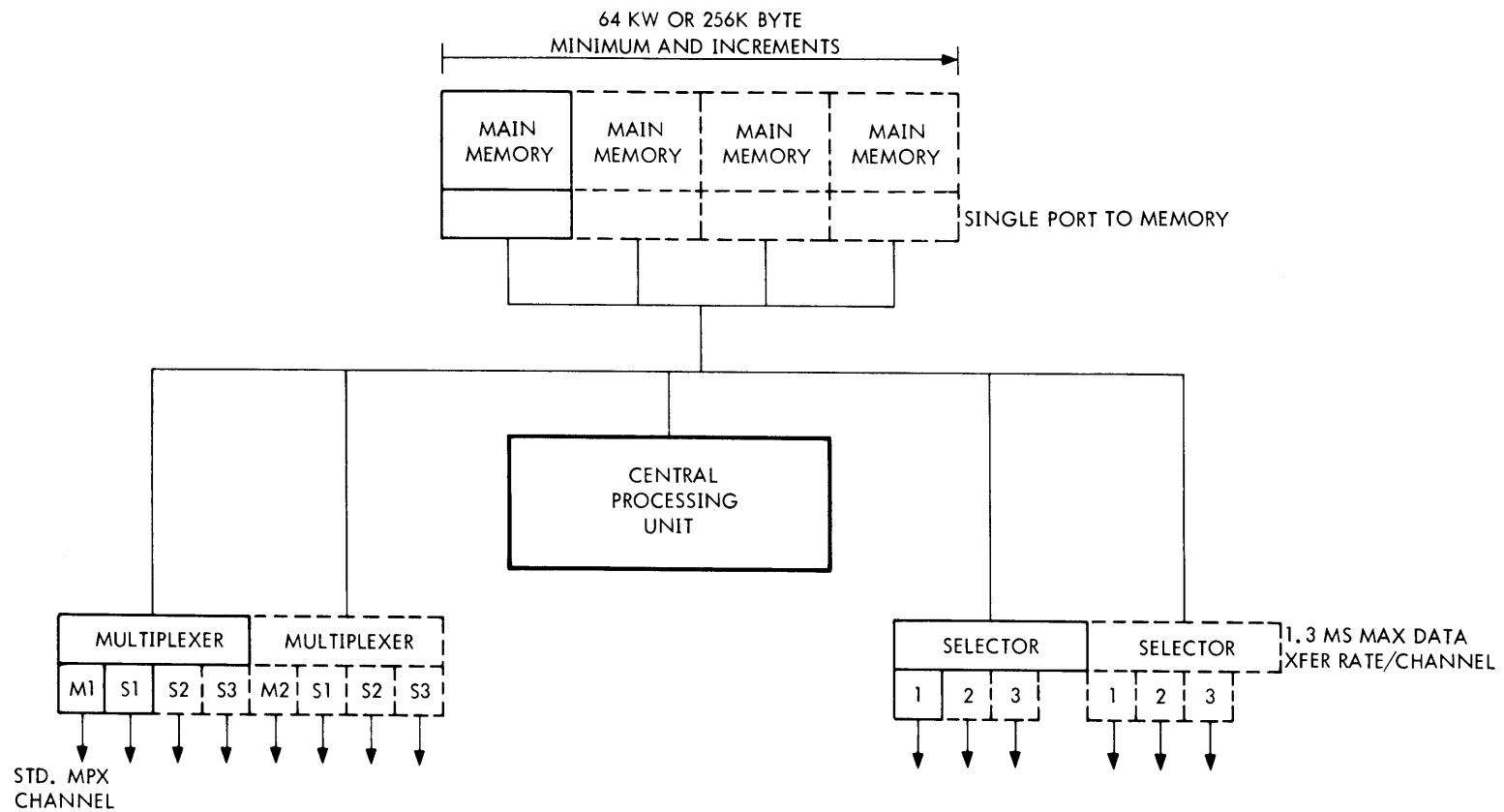
NOTES :

CHANNELS – A MAX OF SEVEN CHANNELS
(1) – 2870 MPX AND (2)– 2860-3 SELECTORS OR
(2) – 2870 MPX, (1) 2860-2, AND (1) 2860-3

MAGTAPES – TSS/360 SUPPORTS MAGTAPE ATTACHED TO THE
2870 SELECTOR SUBCHANNEL ONLY

IBM 360/67-1

512 KB MIN AND 256 KB INCREMENTS W/MAX 2MB FULL DUPLEX

256 KB MIN AND INCREMENTS
W/MAX 1MB HALF DUPLEX

3 PORTS/
HALF DUPLEX
4 PORTS/
FULL DUPLEX

CHANNEL
CONTROLLER

CHANNEL
CONTROLLER

CENTRAL
PROCESSING
UNIT

CENTRAL
PROCESSING
UNIT

MULTIPLEXER

M1 | S1 | S2 | S3

SELECTOR

1 | 2 | 3

SELECTOR

1 | 2 | 3

MULTIPLEXER

M1 | S1 | S2 | S3

SELECTOR

1 | 2 | 3

SELECTOR

1 | 2 | 3
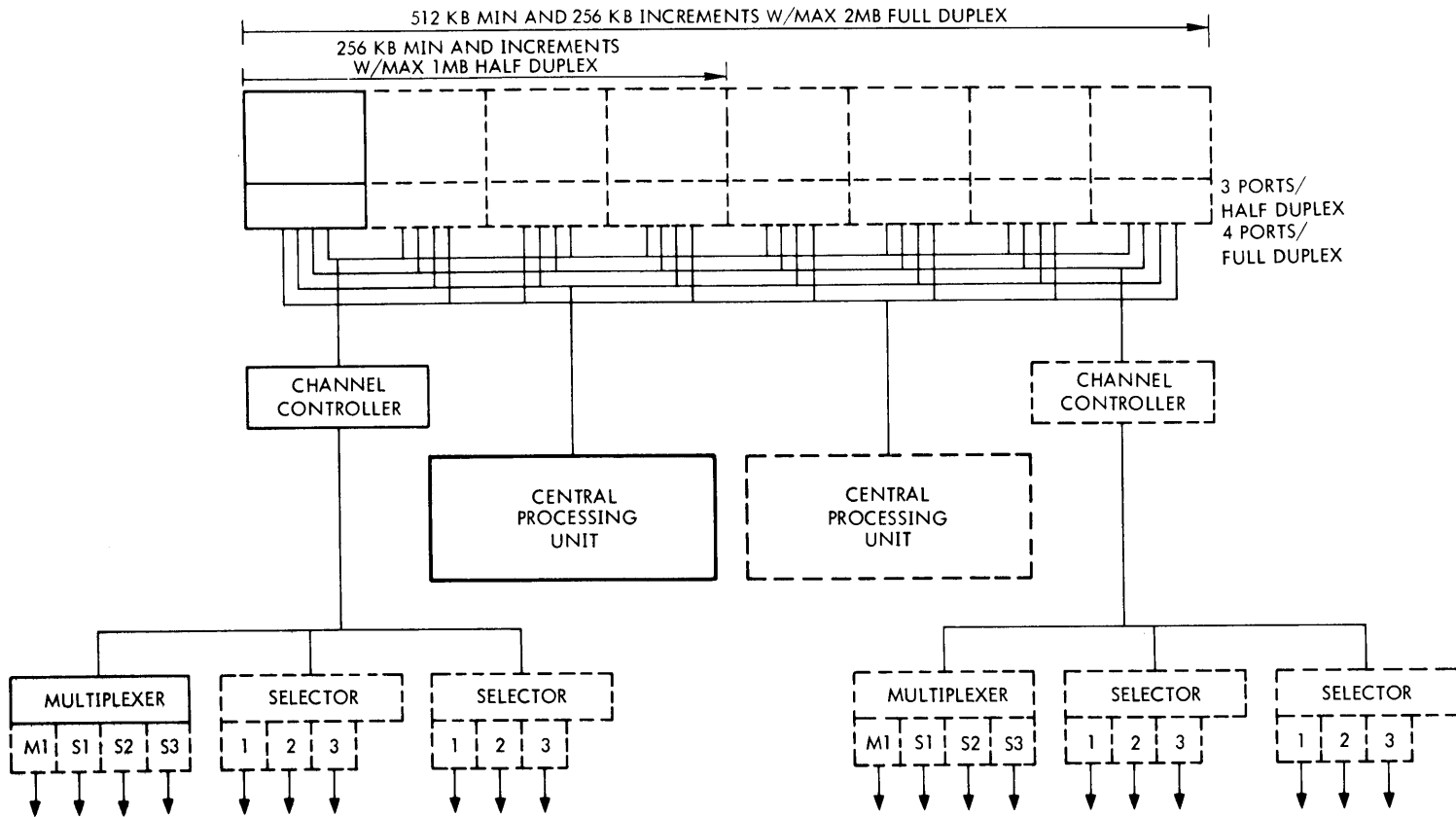
NOTES :

HALF DUPLEX - SINGLE 2067-2 CPU, 256 K B MIN – 1 MB MAX
MEMORY, AND THREE PORTS TO MEMORY
FULL DUPLEX - (2) 2067-2 CPU'S, 512 KB MIN + 2 MB MAX
MEMORY, AND FOUR PORTS TO MEMORY
CPU'S - (1) 2067-2 FOR HALF DUPLEX
(2) 2067-2 FOR FULL DUPLEX
CHANNELS - (1) 2870 MPX + UP TO (2) 2860-3'S PER 2846
CHANNEL CONTROLLER
MAGTAPES - TSS/360 SUPPORTS MAGTAPE ATTACHED TO
THE 2870 SELECTOR SUBCHANNEL ONLY

IBM 360/67-2

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/360

SYSTEM CHARACTERISTIC	MODEL 67-1	MODEL 67-2	XDS SIGMA 9
◦ MAIN MEMORY			
Size Range	256K – 1M Bytes	⟶	512K – 2M Bytes
Cycle Time	750 NS	⟶	850 NS
Access Width	8 Bytes	⟶	4 Bytes
Interleave	2 Way	⟶	4 Way
Access Ports	1	⟶	2 - 12
Map	Optional	Standard	Std.
Protection	Fetch Only	Yes	1st 512KB Only
Min. Increase	256KB	256KB	128KB
◦ BUFFER MEMORY	NA	⟶	NA
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Protection			
◦ LARGE CAPACITY STORAGE	NA	⟶	NA
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Xfer Rate			
Map			
Protection			
Exec. Instr. From			
◦ CENTRAL PROCESSING UNIT			
CPU Clock Cycle	200 NS	⟶	Instr. & Seq. Dep.
Look Ahead			
Instr. Prefetch			12 Bytes Less Current Instr.
Levels of Decode			One Beyond Current

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/360

SYSTEM CHARACTERISTIC	MODEL 67-1	MODEL 67-2	XDS SIGMA 9
● <u>INSTRUCTIONS</u>			
Commercial KIPS	Approx 140	⟶	64.7
Scientific KIPS	Approx 710	⟶	763.4
Word Size	32 Bits or 4 Bytes	⟶	32 Bits or 4 Bytes
Instruction Size	2, 4, & 8 Bytes	⟶	2 or 4 Bytes
Mult. Addr. Ops.	0, 1, or 2	⟶	
Decimal	Standard	⟶	Std.
Fixed Point	Standard	⟶	Std.
Floating Point	Standard	⟶	Sing. & Dbl. Std
Stor.-to-Stor.	Standard	⟶	NA
Byte/Char. Manip.	Standard	⟶	Yes
Stack	NA	⟶	Yes
● <u>ADDRESSING</u>			
Indexing	Yes	⟶	Yes
Indirect	NA	⟶	Yes
<u>Virtual</u>	Yes	⟶	Map - 512KB/Prog.
Page Size	4KB	⟶	512KB
No. of Pages	256 - only 8 Regs	⟶	
Scheme	Base Displacement	⟶	Direct Reference
Size	1, 2, 4 or 8 Bytes	⟶	1, 2, 4 & 8 Byte
Max. Addr.	16MB/Physical & 4BB/Virtual	⟶	4MW or 16MB/ Extended Mode
● <u>REGISTERS</u>			
Index	General Purpose	⟶	General Purpose
Gen. Purpose	16	⟶	2 Blks. of 16 Std.
Mult. Gen. Reg. Blks.	NA	⟶	2 Std. & Opt.
Floating Point	4 - 64 Bit	⟶	Single & Double
Decimal	Standard	⟶	Std.
● <u>INTERVAL TIMER</u>	13 μs	⟶	500 HZ & one of (50/60, 500, 2K, or 8K HZ) Switchable

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/360

SYSTEM / CHARACTERISTIC	MODEL 67-1	MODEL 67-2	XDS SIGMA 9
● INTEGRAL MULTIPLEXER	NA	NA	NA
Channels			
Subchans./Controllers			
Aggregate Xfer			
MPX - Max Chan Xfer			
Burst - Max Chan Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Multiplexing			
Burst			
Command Chain			
Data Chain			
Xfer in Channel			
● EXTERNAL MULTIPLEXER	2 Multiplexor	⟶	1 MIOP Std 1 Max
Channels	2 Basic & 6 Selector	2 Basic & 8 Selector	1 Std. 2 Max.
Subchans./Controllers	8/Chan	⟶	8-32
Aggregate Xfer	450KB	⟶	1.9MB W/4 Byte & 2 Chan.
MPX - Max Chan Xfer	50-110KB	⟶	470KB
Burst - Max Chan Xfer	100KB/Selec	⟶	
Command Chaining	Yes	⟶	Yes
Data Chaining	Yes	⟶	Yes
Data Path Width	1 Byte	⟶	1 or 4 Byte/Dev & 4 Byte to Mem.
● INTEGRAL SELECTOR	NA	NA	NA
Channels			
Subchan./Controllers			
Max. Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Xfer.			
Command Chain.			
Data Chain.			

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/360

SYSTEM CHARACTERISTIC	MODEL 67-1	MODEL 67-2	XDS SIGMA 9
● EXTERNAL SELECTOR	Selector	Selector	1 - 4 RIOP's
Channels	0-5 or 6	0-12	1
Subchans./Controllers	8/Channel	⟶	1 Integral
Max. Xfer.	1.3MB/Channel	⟶	3MB
Command Chaining	Yes	⟶	Yes
Data Chaining	Yes	⟶	Yes
Data Path Width			4 Byte to Mem.
● SYSTEM BANDWIDTH	6.4MB	⟶	25.3MB
● AVAILABILITY			
Watchdog Timer			Yes
Power Fail Safe	Manual Restart	⟶	Saves Volatile Regs. & Brings to Orderly Halt
Recoverability			Yes
Relocatible Homespace	Opt-Float.Stor.Addr	Float.Stor.Adr. Opt-Partition Sense	Yes
<u>Instruction Failure</u>			
Pre. Exec. Test	NA	⟶	Yes
Isolation			Yes
<u>Retry</u>			
CPU			Yes
Channel			Yes

● PERIPHERALS

With few exceptions, the 360/67 and 370/155 can use all of the standard 360 peripherals. For a more thorough listing of IBM's 360 peripherals, including characteristics and specific pricing, consult your Sigma 6 notebook. More specific information regarding the 2000 LPM IBM 3211 Printer can be found in your IBM 370 announcement CSG. IBM has made no mention of enhancing TSS to allow it (360/67) to use the new IBM 3211 Printer.

In the last year or so IBM has announced three new significant peripheral products:

- Printer - 2000 line/minute and program carriage control

- High Speed Disk - A fixed head disk with an average seek access of 5.0MS, 1.5MB transfer rate, up to 11.2MB capacity, Rotational Position Sensing (RPS), and a command queuing controller.

- Bulk Disk - A modular bulk store disk with a 30MS average seek access, 806KB transfer rate, maximum capacity of 800MB per facility (controller), RPS, and command queuing.

PERIPHERALS - W/PRICE FOR COMPARISON

MANUFACTURER   IBM

MACHINE SERIES   SYSTEM 360 and 370

DEVICE		MODEL NO.	SPEED	PURCHASE	MAINT.	1 YR. LEASE
CARD	READ	2520-B2	500cpm	35,000	132	775
		2501-B1	600cpm	14,590	52	260
		2501-B2	1,000cpm	14,820	56	320
	PUNCH	1442-N2	91cpm	18,185	71	365
		2520-B3	300cpm	34,715	106	600
	READ/ PUNCH	1442-N1	400cpm 91cpm	25,460	76	510
		2520-B1	500cpm 300cpm	39,520	40	875
		2540-1 W/2821-6	1000cpm 300cpm	(1) 47,890	(1)205	(1)1,100
PRINTER	120 pos.	1443-N1	240lpm	42,945	88	850
	132 pos.	1403-2	600lpm	(1) 60,080	(1)209	(1) 1,350
	120 or 132 pos.	1403-3 or N1	1,100lpm	(1) 72,800	(1)216	(1) 1,647
	120 pos.	3211-1	2,000lpm	96,750	670	2,800
MAGNETIC TAPES	7-Track	2415-2	15KC	57,590	181	1,300
		(2) 2401-1	30KC	93,530	269	2,040
		(2) 2401-2	60KC	121,010	301	2,640
		(2) 2401-3	90KC	175,550	365	3,840
	9-Track 800 BPI	2415-2	15KB	59,615	184	1,300
		(2) 2401-1	30KB	91,270	268	1,990
		(2) 2401-2	60KB	118,750	300	2,590
		(2) 2401-3	90KB	173,290	364	3,790
	9-Track 1600 BPI P.E.	2415-5	30KB	73,370	215	1,590
		2401-4	60KB	107,375	321	2,340
		2401-5	120KB	134,855	353	2,940
		2420-5	160KB	170,760	475	3,400
		2401-6	180KB	189,395	417	4,140
		2420-7	320KB	265,580	515	5,220

NOTES: (1) Combination control units are available for a 2540 card.
Read/Punch and one printer - reduces price $70/month
(2) Same tape drives 7 or 9 track or combination adaption in control unit.

• MASS RANDOM ACCESS MEMORY

The 360/67 has only a 1.3MB maximum channel bandwidth which is insufficient to handle the slower (1.5MB) fixed head disk storage drive available at the high end of the 360 line Models 85 and 195. In addition, the IBM 3330 removable disk storage units have not been made available to the 360/67.

Because of its CPU buffer memory, the 370/155 can support a system bandwidth several times that of its older 370/50 brother (5.8:1.0) but, in comparison to the Sigma 9 (5.8:25.3), it is very poor. The 370/155 block multiplexing channels have sufficient bandwidth (1.8MB) to support transfer from a single IBM 2305-2 fixed head disk drive (1.5MB), and up to four of IBM's 3330 (removable) disk file storage modules. IBM's 2301 (1.2MB) drum storage is not available on the 370/155.

MANUFACTURER   IBM

MACHINE SERIES System/360 and 370

MODEL NO.	CAPACITY PER MODULE	KB RATE	AVE. ACCESS	CAPACITY	PURCHASE	MAINT.	1 YR. LEASE
2301	4.1MB	1.2MB	8.6Msec	4.1MB 8.2MB 16.4MB	$ 198,780 291,900 478,140	$ 510 945 1815	$ 4,480 6,660 11,020
2303	3.9MB	304KB	8.75Msec	3.9MB 7.8MB	147,010 250,510	463 863	3,425 5,925
FILE STORAGE (REMOVABLE)							
2314	233MB	312KB	87.5Msec	233.4MB	237,105	615	5,250
2314A	29.2MB	312KB	72.5Msec	58.4MB 116.7MB 233.4MB	108,475 145,720 224,605	195 300 540	2,400 3,225 4,970

NOTE: 2314 contains eight drives plus a spare
2314-A is modularly packaged. Prices are for capacities indicated with no spare.

The above listed devices are almost obsolete, especially where the System/370 is concerned. These models are now, or soon will be, no longer in production. Other devices, 2311, 2302 and 2321 in particular, have been out of production for some-time and so were not included.

RANDOM ACCESS DEVICES

MANUFACTURER ___IBM___

MACHINE SERIES ___System 360 and 370___

MODEL NO.	CAPACITY PER MODULE	KB RATE	AVE. ACCESS	CAPACITY	PURCHASE	MAINT.	1 YR. LEASE
SYSTEM STORAGE AND SWAPPING RADS							
2305-1	5.4MB	3.0MB	2.5 μsec	5.4MB 10.8MB	$ 371,300 601,600	$1110 1695	$ 7,900 12,800
2305-2	11.2MB	1.5 MB	5.0 μsec	11.2MB 22.4MB	300,800 484,100	1000 1555	6,400 10,300
FILE STORAGE (REMOVABLE)							
3330	100 MB	806 KB	46.7 μsec	200 MB 400 MB 800 MB	173,900 235,000 357,200	370 570 970	3,700 5,000 7,600

• COMMUNICATIONS

IBM has, primarily, three multi-line communications controllers.

- IBM 2701 Data Adapter Unit: The 2701 provides for the attachment of remote and local I/O devices (communications gear, plotters, etc.). Through any of the I/O channels, the 2701 provides for attachment of four start/stop lines, four synchronous lines, or data acquisition systems/devices. Specific adapters enable the 2701 to provide CPU to CPU connection over communications lines. With appropriate attachments, the 2701 can operate in a binary sync mode at speeds up to 230,400 bits/second.

- IBM 2702 Transmission Control: The 2702 attaches to a multiplexer channel and, with appropriate adapters, provides for connection for up to 31 half duplex lines at speeds of 180 bits/second or 600 bits/second. The 2702 is used to interface IBM's 2712 Remote multiplexor and other terminals.

- IBM 2703 Transmission Control: The 2703 is used when attaching large numbers (up to 176) of asynchronous lines at speeds up to 165 bits/second, up to 72 asynchronous lines up to 600 bits/second, up to 48 synchronous lines at speeds up to 2400 bits/second, or up to 24 synchronous lines at speeds up to 4800 bits/second.

None of IBM's communications controls have memory or have programmable control capabilities. In addition, none of IBM's controls have a rate sensing capability.

MANUFACTURER IBM

MACHINE SERIES System/370 and System/360

SYSTEM CHARACTERISTIC	2701	2702	2703
Function	Multi line contr.	Multi line contr.	Multi line contr.
Programmable contr.	No	No	No
Memory	None	None	None
Data path interface	MPXR or SBL C.H.	MPXR C.H.	MPXR C.H.
Number of lines	up to 4@ 600 bps, or 2 H.S.	up to 31@ 200 bps, 15@ 600 bps	up to 176@ 134.5 bps
Max. bit rate/line	230,000	200 bps (600 opt)	up to 4800 bps
Aggregate bit rate	460,000 bps	around 9000 bps	around 11,500 bps
Rate sensing	No	No	No
Mode(s)			
HDX, FDX, ECHPLX	HDX	H DX	HDX
ASYNC, SYNC	Both	ASYNC	Both
BISYNC	Yes	Not applicable	Yes
Data format	5, 6, 7, 8	6 or 8	5, 6, 7 or 8
Code translation	None	None	None
SYNC Pattern	Fixed	Not applicable	Fixed
Control code recog.	Fixed – in adapter	Fixed – in adapter	Fixed – in adapter
Buffer size	1 to 6 bytes	1 byte	4 bytes
Interrupts	Message	Message	Message
Auto dial	OPT	OPT	OPT
Auto answer	Yes	Yes	Yes
Parallel data input	Yes	No	No
Voice answer back	No	No	No
Polling	Yes	Yes	Yes
Error Check	Over run	LRC, VRC	LRC, VRC, CRC
NOTES:	LRC, VRC, CRC		

REAL TIME

IBM has completely ignored critical real time throughout its major computer systems prior to the July 1970 announcement of the Real Time Monitor (RTM) extensions to OS MVT. In an effort to re-direct the "lambs back to the fold", IBM has used the phrase "real time" to mean even the hand-carrying of reports to management in sufficient time for them to impact events prior to their forecasted occurrence. Just recently, with the announcement of the 370, IBM stresses how their system's processing power has been extended to handle the needs of today's "real time" world.

Make sure that your prospect is well aware of real real-time differences:

- Priority interrupts

- Rapid context switching, register blocks, etc.

- Reentrant programming support

- High throughput I/O processing

- Real Time Clock repertoire

- Real Time Core Management/Map, etc., etc.

IBM would have to attach special channels and front-end gear in order to obtain the real time priority level capability that is found on Sigma. The newly announced Real-Time Monitor (RTM) option of OS MVT has many drawbacks:

- In order to get "normal" monitor services (I/O, scheduling of other foreground tasks which are dependent upon the interrupt-driven task) must go to OS and big systems

- Time required for "handshaking" between RTM and OS

- Software "filter" required to ascertain whether interrupts belong to RTM or OS (32 instruction - subroutines) — Suspect hardware interrupts not properly used.

- In order to get XDS equivalent of hardware interrupts requires at least a 2909 channel plus 32 levels of interrupt (these items must be RPQ'd) or a separate, front-end 1800 or 360/44 system (these systems must be programmed separately from the central computer.

- Program checkout can be very lengthy and complex because: Programmer must take steps to protect background environment (RTM can get access to all 370 "privileged" instructions).

- Management of all system resources is made more complex and difficult because of the additional demands placed on the system by RTM.

Questions to raise:

- How does the system (in batch environment) <u>account</u> for RTM resource usage? Is it charged to batch user? or what?

- Resource allocation – The RTM user is satisfied first. What happens to the batch user who needs the resources? Batch throughput declines and batch jobs are delayed (say, payroll).

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/360

SYSTEM CHARACTERISTIC	MODEL 67-1	MODEL 67-2	XDS SIGMA 9
● TIME CRITICAL REAL-TIME		NA	
Priority Interrupts			
Levels	1		224
Branch Address	256 Unique		PSD SWAP
Arm/Disarm	NA		Yes
Enable/Disable	Yes		Yes
Triggerable	NA		Yes
Clocks	13 µs	→	2-STD, 2-OPT (500HZ and 50/60, 500, 2K, or 8K HZ, or Other - Switchable)
Context Switch			
Exec. Time to Change Control	/LPSW	→	XPSD 5.43 µs
Mult. Reg. Blks. or Load/Store Mult.			Mult. Reg. Blks.
Instrumentation	DAC Center RPQ	→	Extensive List of Standard Products in Addition to Separate Systems Division Product Line.

MANUFACTURER: INTERNATIONAL BUSINESS MACHINES

MACHINE SERIES: IBM SYSTEM/370

SYSTEM CHARACTERISTIC	MODEL 145	MODEL 155	XDS SIGMA 9
● TIME CRITICAL REAL-TIME			
Priority Interrupts			
Levels	1	⟶	224
Branch Address	256 Unique	⟶	PSD SWAP
Arm/Disarm	NA	⟶	Yes
Enable/Disable	Yes	⟶	Yes
Triggerable	NA	⟶	Yes
Clocks	Interval Timer	⟶	2-STD, 2-OPT (500HZ and 50/60, 500, 2K, or 8K HZ, or Other - Switchable)
Context Switch			
Exec. Time to Change Control	8.145μs	?	XPSD 5.43 μs
Mult. Reg. Blks. or Load/Store Mult.	17.78μs/15 reg	?	Mult. Reg. Blks.
Instrumentation	DAC Center RPQ	⟶	Extensive List of Standard Products in Addition to Separate Systems Division Product Line.

SOFTWARE

- IBM SOFTWARE MARKETING STRATEGIES FOR OS MVT ON 370/155

    IBM will stress the following items about OS MVT:

    Compatibility
    Applications
    Time-Sharing
    File and Data Management Support for large data bases
    Real Time
    Communications Support

    - Compatibility: The 370/155 will execute 360 programs without change. Additional capability is provided for combining DOS with OS MVT and also providing emulation at the same time. Thus, a present 360 user can run all his programs in a multiprogramming environment.

    - Applications: This support is provided for a variety of cross-industry applications as well as within individual industries.

    - Time-Sharing: BASIC and PL/I are provided under ITF. FORTRAN and other support is provided under TSO of which ITF can be a subset.

    - File/Data Management: This is tied directly to the new 3330 disk file capability of 800MB. Also, the 2305 disks are supported. IMS was enhanced.

    - Real Time: This is a newly offered capability which provides many real time features.

    - Communications: OS MVT supports a wide variety of slow and high speed terminals.

- OS MVT — STRONGPOINTS

    OS MVT offers a wide variety of capability as indicated above. The following items are considered the strongest features of OS MVT:

    - Mature: OS MVT has been in operation for over two years.

    - Flexible: The system can be expanded/modified to fit varying situations.

    - Languages: A variety of languages are available — FORTRAN, COBOL, ALGOL, PL/I, and others.

    - Production: OS MVT is strongly oriented to production.

- OS MVT — WEAKNESSES

The major weakness of OS MVT is the shear size of it, including the core residency requirement of 100KB or more. Other weaknesses are as follows:

- Complicated: The job control language (for use as job control cards) is extremely complex and unforgiving. The size of the system and all its elements is a nightmare for systems generation.

- Scheduling/Resource Allocation: OS MVT is a priority driven system. This means that the top job priority gets all the CPU time and other resources it needs and other jobs take what is left over (unless they already had tapes/disk files when the high priority job came in — core may be given up).

- Real Time Support: This was an after-thought for OS MVT. The response is very poor due to the centrally connected interrupts (a 32-instruction routine decides between RTM and OS). Program checkout is dangerous due to possible damage to operating system (RTM has access to all privileged instructions).

- XDS STRATEGY

Time-sharing, real time and the multi-use ability of XDS are the primary areas to stress when facing OS MVT/TSO.

Otherwise, XDS should be in the picture when consideration is being made to use OS MVT (or OS MFT) for the first time. Or, at least the consideration is that some of the previously indicated strengths of OS are negated. This means that some of the usual conversion problems do not exist such as emulation/simulation of another IBM system; package application program use is to be minimum (ours will suffice); development of programs is emphasized rather than production (for the case of UTS). In specific cases where the main consideration is in the area of a specific OS strength (such as a particular communication device), it is advised to get the Data System group in early to provide special engineering and programming services.

Time-sharing and real time are very strong.

- SIGMA 9 STRENGTHS — UTS

- Time-Sharing Capability: OS MVT has no interactive/conversational time-sharing at this time. ITF, just presently becoming available under DOS, has not been placed under OS MVT. TSO, an option under OS MVT, is scheduled for 1Q71.

- UTS VS ITF

ITF provides two languages, BASIC and PL/I, and, as such, is such a minimum capability it offers no direct competition to UTS.

- UTS VS TSO

  UTS provides all the following unique features:

  - Shared Processors: OS MVT/TSO provides no capability along this line. TSO functions much like BTM — in a dedicated area of core. Only one user is there at a time. Of course, more than one area can be devoted to TSO, but no processors are shared.

  - Mapped Memory: There is no capability under OS MVT/TSO to fragment memory. All programs must reside in contiguous space; thus, there is no way to selectively remove a program from core and then return it to any alternate location!

  - Performance Management: IBM's philosophy has always been that they will not provide dynamic computer measurement features. This is considered a customer function and to be arrived at as he sees fit. This outstanding feature of UTS is an industry leading feature.

- SIGMA 9 STRENGTHS — XOS

  - Memory Management:

    Monitor — OS MVT devotes a fixed area to transient monitor routines. Alternately, an installation can dedicate an additional area for resident monitor routine. Other routines, such as Open/Close, which normally would reside in the user areas, can be selected for residency in the linkpack area — a dedicated area in upper core. This is contrasted to the mapping of the monitor under XOS.

    First of all, it must be remembered that the XOS logical I/O routines are a part of the monitor and thus are not required in each user area as they are under OS MVT (except for an installation selection for residency – generally, the most used routines are made resident, but the larger ones are not). Transient monitor routines are provided mapped core storage on an as-needed basis under XOS. This means that the core space for them is not completely taken away from the user except on an as-required basis. They occupy core space on a dynamic basis and in areas currently not in use by the user programs. These routines become resident because of usage and not because of overt dedication. This results in a dynamic allocation of monitor used space and, as such, optimizes core usage. Contrast this to the OS MVT operating characteristic that either the installation management must select the dedicated monitor area, or the operator must look ahead for each run — ususlly two or more hours — and halt the system, re-boot the monitor to place specific routines in resident areas and then continue operation. XOS takes 64KB (16KW) for dedicated space versus at least 100KB (25KW) for OS MVT.

User Area — OS MVT requires a region (contiguous) of space in core for each user. This size is specified in the job card and reserves that much core for the duration of a job. Consider that a particular job is compile, load and go. The region size is selected to be large enough for the compiler and a corresponding loader is selected. The execution phase may only occupy 1/4 or 1/3 of the region and may well take more time for execution. Thus, during execution, 2/3 to 3/4 of the region lays fallow (un-used). Suppose that 3/4 of the jobs do this and execution time equals compile and load time. This means that 1/2 of core goes completely unused for 1/2 plus the time!

XOS does not let this situation occur. Each step of a job uses only what core it needs — specifically the execution step discussed above would not reserve the whole region; thus, other jobs/job steps have access to the unused core space. Next, OS MVT dynamically loads into user space any of the logical access methods (ISAM, etc.). These routines are large and oftentimes repetitious within the user area. At a given time, there may be many copies of identical routines occupying space.

XOS does not let this happen because they are built into the monitor. Similar routines are either encased in the monitor for use by all programs or one copy occupies space and all users can use that routine. A job in XOS can also dynamically extend its size without being aborted. With sufficient priority, OS MVT allows a program to roll out another program (or region). The rolled out program must then be returned to the identical core space that it occupied before being rolled out. XOS never over allocates. Should a program request core above the limit specified, and core not be available, the program waits until it is available.

Scheduling Techniques — OS MVT takes the top job off the priority list and accumulates core for it. After the job gets core, it then sets about requesting files and/or devices — it must accumulate these until its needs are satisfied (no other job can get started during the process of accumulation). Once it does get all necessary resources, it hogs the system if it has highest priority. In other words, it takes all CPU time it needs - if it is compute bound, it gets all the time. Contrast this to the variety of control that XOS offers. If you wish to run with a scheduling algorithm as just discussed, you can with a reduced efficiency of system utilization. XOS provides other scheduling methods under system generation and/or operator control. For example, in the case above, if the system determined that another job, lower priority, could be completed before the necessary resources were released, the other job would be started and be completed in time to prevent holding up the higher priority job. Many other scheduling features such as time-slicing are available to be selected for a particular installation.

- OTHER XOS STRENGTHS

  - Utility Routines: XOS supports these within the resident monitor structure (parallel jobs). OS MVT places these in the job stream to occupy user core space.

  - Symbiont Functions: XOS has these built into the monitor. OS MVT has reader/interpretors (up to 3) that may be resident or transient (between jobs). During residency (and at least one is usually resident), they occupy either 30KB (7.5KW) or 44KB (11KW). Optionally, OS MVT may use HASP (about 100KB) for a combined spooling, control stream entry, and handling remote job entry (RJE). Remote job entry takes another 80KB when it is resident.

    In summary, over 200KB (50KW) could be devoted to symbionts and RJE by using normal OS support and, usually, the customer will use HASP for 100KB. Remember, XOS does all these things as a part of the monitor including the utilities.

  - Real Time: OS MVT has weakness under RTM in the real time area, as previously noted. XOS provides its own external interrupts for real time. Resident and non-resident real time are provided as monitor functions under XOS with complete facilities for developing and checking out real time programs from batch, remote batch and time-sharing terminals. Additionally, real time programs can be initiated by the operator, a job in the batch/remote batch stream, or a terminal. RTM programs must be present (at least partially) all the time.

TIME-SHARING SOFTWARE COMPARISON

	UTS	IBM 360/370 OS MVT/TSO
Operational Modes —		
• Dedicated Time-sharing (max no. of terminals)	Yes (128)	Yes (variable)
• Time-sharing/Batch	Yes	Yes
• Time-sharing/Batch/Remote Batch	Yes	Yes
• Real time (combined with other modes)	Yes	RTM
Installation Control Features —		
• System control		
– Performance Management*	Extensive	Very limited
• User control		
– Accounting per user*	Yes	Yes
– Central account authorization*	Yes	Yes
– Processor authorization by account*	Yes	Yes
– Terminal Control		
Passwords	Yes	Yes
Automatic connection to processors* (Restricted use)	Yes	Yes
Automatic connection to charge structure* (Unique charges based on usage)	Yes	No
Shared processors (Reentrant)*	Yes	No
– Batch control		
Controlled resource usage (core, priority, devices, etc.)*	Yes	Yes
Batch time quantum control	Yes	Limited
Percentage of CPU for batch	Yes	No
– Remote batch control	Same as batch	Good
– Real time control	Yes	RTM
• Operator Control		
– Core size for terminal user	Yes	No
– Number of terminal users	Yes	Yes
– File space	Yes	Yes
– Number of tapes/user	Yes	Yes
– Percentage of CPU for batch	Yes	No
– Terminal time quantum	Yes	No
– Compute bound quantum	Yes	No
– Batch priority relation to terminal	Yes	No

	UTS	IBM 360/370 OS MVT/TSO
Program/Application Development Features —		
• Terminals		
– Conversational processors	Basic	Editor, Basic
	Extended XDS	FORTRAN
	FORTRAN IV	PL/I Subset
	EDIT	Syntax Checkers
	META-SYMBOL	
– Interactive Languages	PCL	Yes
– Media conversion	Yes	Yes
– TJE to batch stream	Yes	Yes
– Processor compatibility with batch	Yes	Yes
– File interchangeability		
– Terminals can be logged on for particular		
processor	Yes	Yes
– Sharable processors*	Yes	No
– Debug facilities (conversational)	FDP, DELTA	Limited
• Batch/Remote Batch		
– Language Processors		
Standard FORTRAN, COBOL – others	Yes	Several versions of each
– Other Processors	SL-1, FMPS,	Many
• Real Time	DMS, GPDS	
– Languages	FORTRAN, MS	RTM
– Other processors	SL-1	
– Terminal program development	Yes	
• Terminal		
– Terminal job entry (under priority control)	Yes	Yes
– Compiling not required for each run (load		
modules maintained)	Yes	Yes
– Output optionally permanent	Yes	Yes
– Terminal must stay connected	Yes –	Yes –
	time-sharing	time-sharing
	No – batch	No – batch
• Batch/Remote Batch		
– Priority scheduling	Yes	Yes
– Standard system files available	Yes	Yes
– Non-standard files available	Yes	Yes
– Printer forms control	Limited	Yes
– Checkpoint/Restart	No	Yes
• Real Time		
– Initiated by terminal	Optional	No
– Initiated by batch/remote batch	Optional	No

	UTS	IBM 360/370 OS MVT/TSO
– Initiated by interrupt	Optional	Yes
– Initiated by operator	Optional	Unknown
– Preempts core	Possible	Possible
– Highest priority	Yes	Not really

---

* UTS Reference Manual covers these topics in detail. Additional information is contained in UTS Specification No. 702489.

Manufacturer: IBM
Operating System: OS MVT/TSO
Computer System: IBM 360/370
Operating System Classification: General Purpose/Time-Sharing

	XOS	OS MVT/TSO
Operational Modes –		
• Batch/Remote Batch	Yes	Yes
• Dedicated Time-Sharing (max no. of terminals)	Yes (variable)	Yes (variable)
• Batch/Remote Batch/Time-Sharing	Yes	Yes
• Real Time	Yes	RTM
Batch/Remote Batch Multiprogramming –		
• Minimum System Requirements	48KW	128KW
• Number of Concurrent Programs	7 (R.T. included)	15 max
• Remote Batch is Part of Monitor	Optional	No (acts as program)
• Symbionts Are Part of Monitor	Yes	No (takes user space)
• Scheduling Criteria	Priority, Avail Resources	Priority, Core
• Checkpoint/Restart	Yes	Yes
• Disk Pack Support	Yes	Yes
• Memory Management		
– System	Mapped	Dedicated
– User	Mapped	Variable size, contiguous space
• Processor Support	FLAG, FORTRAN, COBOL, SORT/MERGE	Extensive
• Methods of Job Entry	CR, Tape, Disk, RBT	CR, Tape, Disk, RBT
• CPU Time Allocation	Priority	Priority
Communications Support –	Character and Msg oriented	Extensive
Time-Sharing –		
• Languages	All Standard	See T/S Report
• Performance Management	Yes	
Real Time Support –	Full	RTM
Applications –	DMS, GPDS, SL-1, FMPS	Wide variety

SYSTEM PRICES AND POLICIES

- **PRICES**

    - IBM's basic rental agreement provides 176 metered hours of usage. Overtime charges average 10% of basic monthly rental per hour of extra usage.

    - IBM includes maintenance in the basic monthly rental charge. For installations paying extra usage charges round-the-clock, emergency maintenance support is included in the extra usage charges.

    - IBM has NO long-term lease agreement.

    - IBM allows a purchase option credit of 50+% of first year's monthly rental. For Federal, State and Local Governments, a credit is allowed of 50% of the first two year's monthly rental.

    - IBM has gone to separate pricing of training, systems analyst support, and program products.

- **POLICIES**

    - IBM is using hardware on the System/370 to enforce their program product's licensing agreement. Built into the CPU and I/O channels are unique ID numbers which, in conjunction with Read ID instructions, will limit program operation to a specific machine.

    - The System/370 announcement included a unique one-year warrenty including parts and labor on all purchased System/370 equipment. All maintenance performed under the warrenty must be accomplished on standard shift. They will charge for emergency maintenance performed on an extra shift.

PURCHASE PRICE PROCESSOR, MEMORY AND I/O (MIOP)

RENTAL PRICE PROCESSOR, MEMORY AND I/O (MIOP)

STORAGE CAPACITY
K BYTES
K = 1024
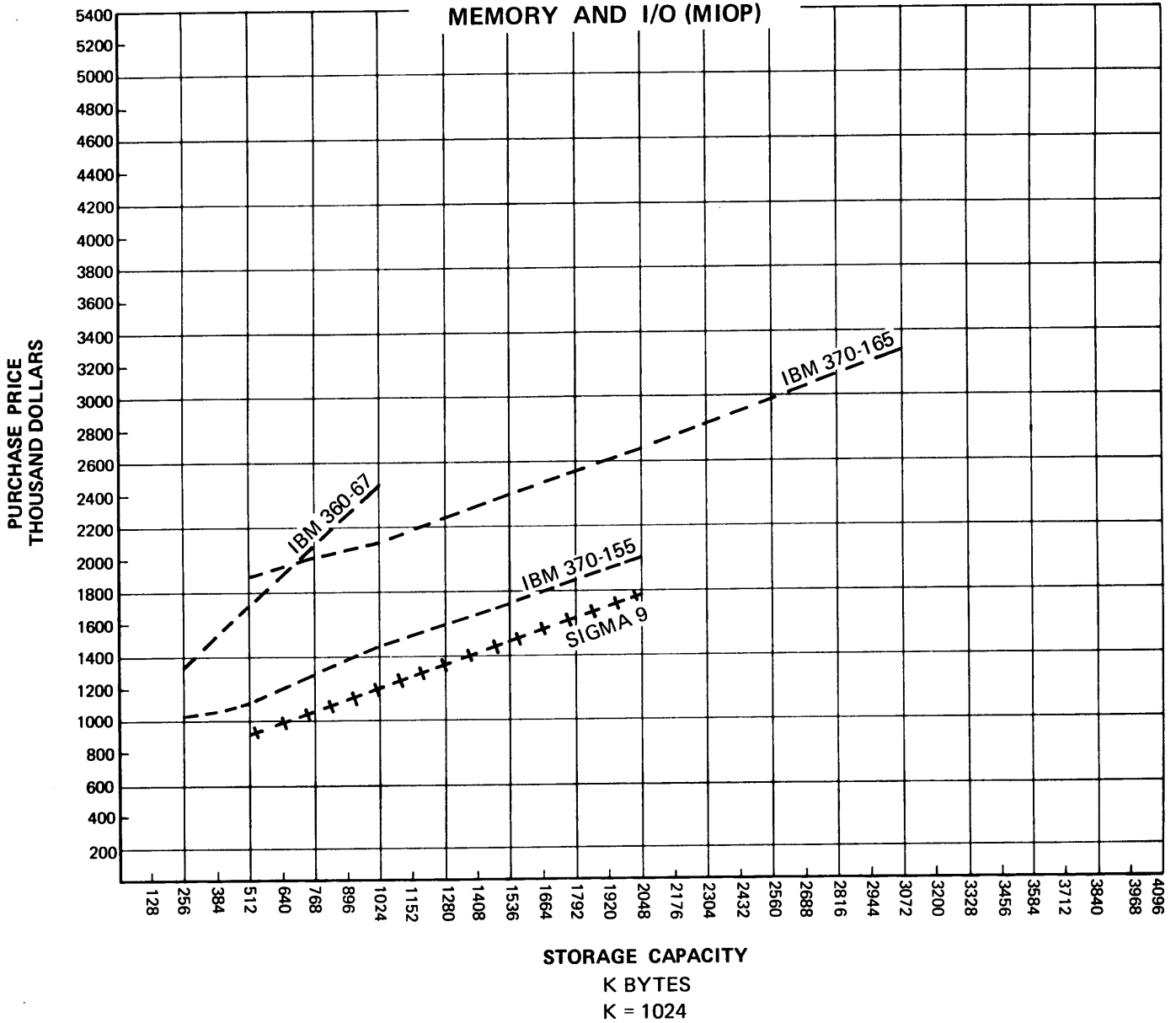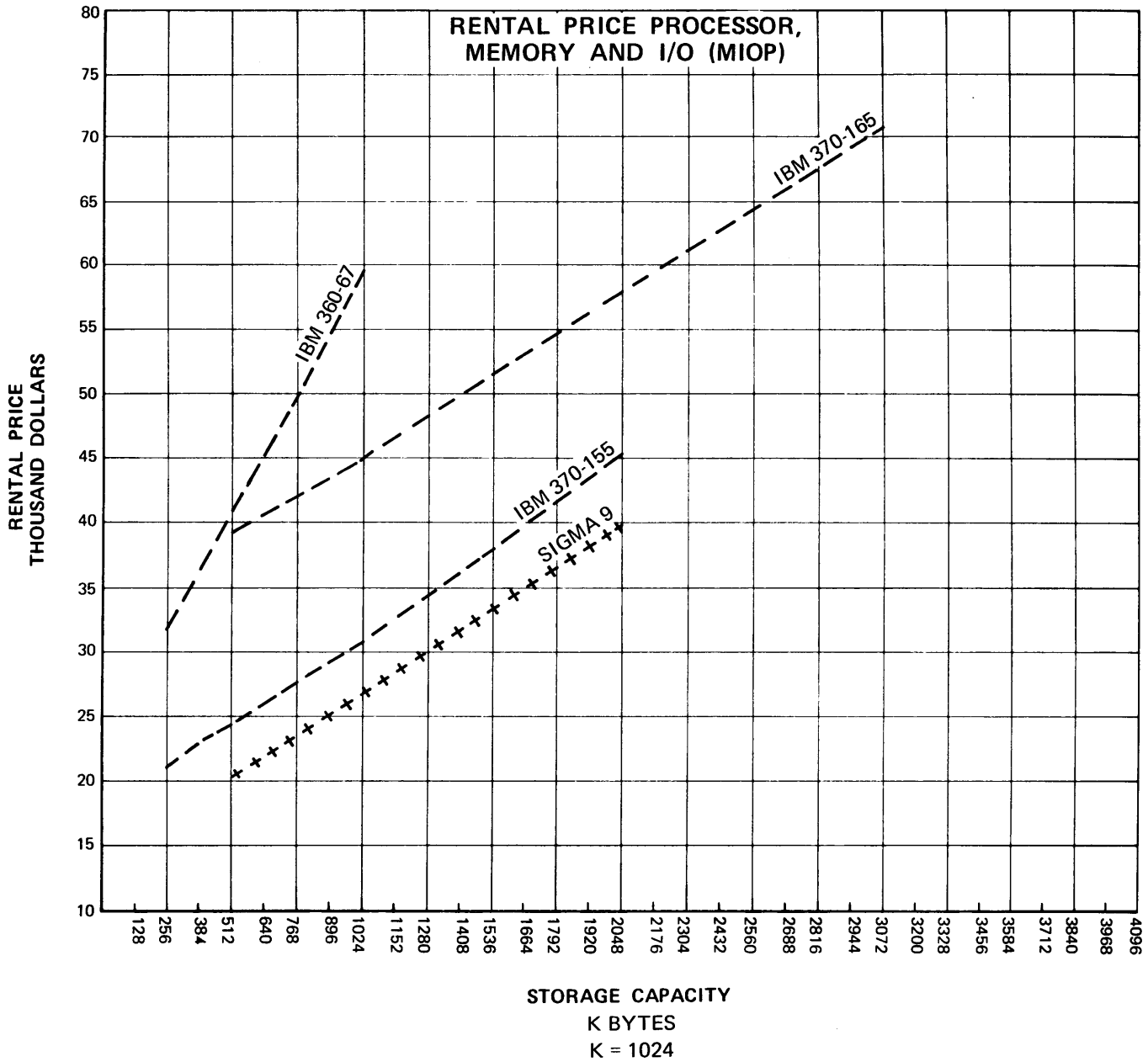
Competitive Sales Guide

OCTOBER 1970   COMPETITIVE ANALYSIS   File: UNIVAC
1106/1108

Contained in this Sales Guide is an assessment of the UNIVAC 1106 and 1108 computer systems versus Sigma 9. The most important of these two competitors will be the 1106.

CONTENTS

Please refer to the Sigma 9 Overview (File: XDS) Sales Guide to compare the configuration diagrams of the UNIVAC 1106 and 1108 to Sigma 9, and also refer to the other Sales Guides on Sigma 9 competitors for additional information.

Distribution: Sigma 9 Announcement
Distribution and Competitive Infor-
mation Reference Libraries

Kent L. Cootes, Manager
Competitive Analysis
XDS   AFC

COMPANY PRIVATE

Xerox Data Systems

UNIVAC 1106/1108

SYSTEM SUMMARY

- INTRODUCTION

    The Univac 1108 was introduced in 1965 as a substantially improved version of the 1107. Modifications have been made to the 1108 since its announcement. These include multiprocessor capability (up to three processors); independent I/O processors (1 or 2 extended I/O controllers with up to 16 simultaneously active channels per IOC); bulk core availability; and (in mid 1969) the availability of the Exec 8 operating system. The 1108 is a very expensive but also very powerful 36-bit binary machine, primarily designed for the scientific marketplace.

    The Univac 1106 (first announced in March 1969) is a slowed down version of the 1108, designed to extend the life and the market of the 1108. A CE can upgrade an 1106 to an 1108 in a matter of minutes. The 1106 is included in this study because, with modular memory, its price/performance ratio places it mid-way between the Sigma 7 and the Sigma 9. Any references to the 1106 in this document apply only to the 1106 with modular memory - 1106 with unitized memory has not been considered.

    This report deals primarily with the 1108. For specific details on the 1106 not covered by this report, please refer to the Univac 1106 Competitive Sales Guide, December 8, 1969.

    Univac attempts to sell both of these machines in time-sharing environments but hardware and software limitations make both machines very weak in time-sharing. The 1106 with modular memory can handle four to six terminals; the 1108 with internal I/O channels only (no IOC's) can handle a maximum of 10 to 12 terminals and still process some batch or remote batch. With independent IOC's, or with the shared processor system (announced in January 1970) which features an 1108 processor dedicated to I/O handling, the time-sharing capabilities of the 1108 may be enhanced. Unfortunately, theer is currently no information available on either of these latter two configurations.

    Essential differences between the 1106/1108 are listed below:

Model	Multiprocessor Capability	Independent I/O Controllers	Memory Cycle Time
1106	No	No	1.5 μsec
1108	Up to 3	1 or 2	0.75 μsec

- **UNIVAC 1106/1108 STRENGTHS**

  - Multiprogramming: Exec 8 is a powerful multiprogramming system of the same class as IBM's OS/MVT. It handles a variable number of batch and remote batch jobs. Operator scheduling is kept to a minimum.

  - Scientific Processing: The 1108, despite its age, is a very powerful machine in a strictly scientific environment.

  - Random Access Devices: The wide range of mass storage devices available must be considered a plus for Univac. However, software support for the removable disk pack is questionable at this time.

  - Application Packages: Univac has a wide range of scientific application packages available.

- **UNIVAC 1106/1108 WEAKNESSES**

  - The 1106/1108 represent a second generation architecture. They are not industry compatible (6-bit character rather than 8-bit byte). The 1108 is a virtually obsolete machine; the 1106 was Univac's attempt to extend its life one or two more years.

  - Time-sharing is strictly an add-on to both the hardware and the software of the 1106/1108. Because of poor design, a very few terminals (4 to 6 on the 1106 – less than 15 on the 1108) will completely swamp the CPU. The reasons for this are the exceptional amount of operating system overhead and the cycle-stealing of the standard I/O channels. (The effects of the independent IOC's is not known. We have no reports of successful installations.)

  - There is no critical real time hardware capability. Only one external interrupt level is available.

  - There is very poor storage utilization. The lack of a memory map requires excessive shuffling of core. Programs are typically large batch programs and language processors are not reentrant as on Sigma.

HARDWARE

Some of the highlights of the Univac 1106/1108 hardware features include:

- **MEMORY**

  - Memory size ranges from 384K to 1.5M characters (6 bits/character)

  - Cycle time 1.5 μsec for the 1106
                .75 μsec for the 1108

  - Access width is two words, one operand and one instruction if stored in different modules.

  - The 1106 does not permit interleaving
    The 1108 can have 2-way interleaving

- The 1106 has one port to memory
  The 1108 can have up to 5 ports to memory

- Memory protection is provided by two sets of Base and Limit registers, one for programs and one for data (thus permitting programs and data to be stored in different storage modules to take advantage of Fetch execute overlap.

- **CPU**

  - 36-bit word

  - No decimal or byte string instructions

  - Fixed and floating point arithmetic is standard

  - Special processor modes permit operating on quarter, half or third of a word. This is not the equivalent character addressing. There is no carry between the partial words.

  - Two sets of general registers (16 each), one reserved for the operating system and the other for the user plus 15 separate – user or system – accessible index registers.

  - Only 47 of the 128 total registers are program-accessible.

  - Overlap of operand fetch and next instruction is automatic if they are stored in different storage modules.

- **INPUT/OUTPUT**

  - I/O System: Integral I/O standard on both the 1106 and the 1108. Independent I/O controllers (maximum 2) are optional on the 1108.

  - I/O capacity: Four to 16 simultaneous channels on the 1106. Eight to 16 simultaneous channels on the 1108. Each independent IOC on the 1108 can have 4 to 16 simultaneous channels.

  - Standard integral I/O channels cycle-steal (and heavily) from the CPU.

  - The optional (and expensive) independent IOC's each have a separate port to memory.

  - I/O bandwidth:

1106 standard channels	333K words per second per channel. Total rate 667K words per second
1108 standard channels	440K words per second per channel. Total rate 1.33M words per second
1108 external I/O controllers	250K words per second per channel. Total rate 1.33M words per second

  - Total system bandwidth:

    1106 – 667K words per second
    1108 – 1.33M words per second

64 KW OR 384 K CHAR MINIMUM AND INCREMENTS

1.5 µS CYCLE TIME
FOR 72-BITS
(DOUBLEWORD)
MAX MEMORY SIZE
256 KW OR 1536 K
CHAR MEMORY ALSO
AVAILABLE IN
UNITIZED 256 KW
SINGLE MODULE

SINGLE PORT
TO MEMORY

MAX XFER RATE 667 KW
OR 4 MCHAR PER SEC

INTEGRAL I/O CONTROLLER

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

CENTRAL
PROCESSING
UNIT

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
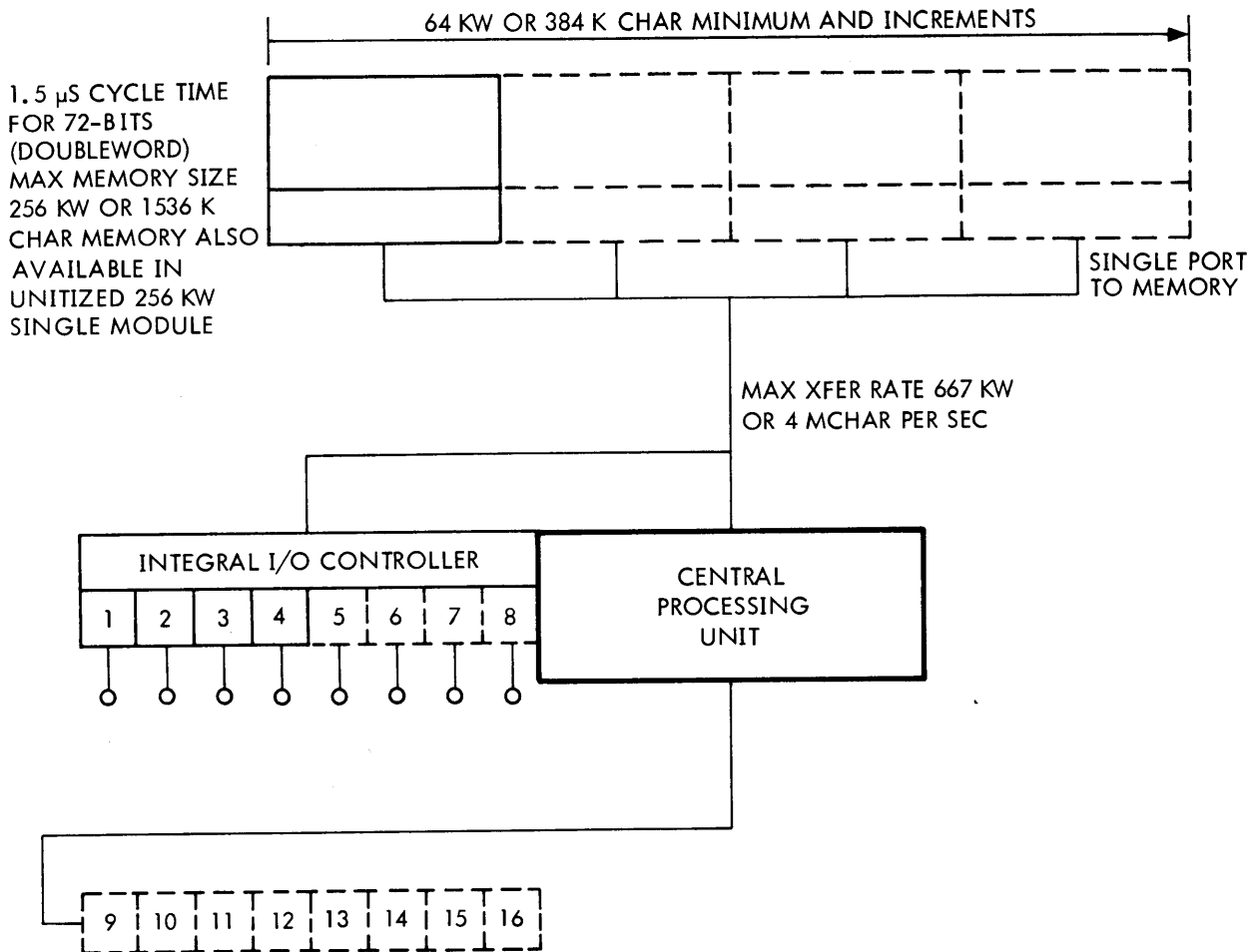
NOTES :

(1) THE INTEGRAL I/O CONTROLLER HAS
4, 8, 12 OR 16 CHANNELS

(2) CHANNELS CAN DATA CHAIN

(3) ONE CHANNEL IS DEDICATED TO SYSTEM CONSOLE

UNIVAC 1106

6

65 KW OR 384 K CHAR MINIMUM AND INCREMENTS

750 NS CYCLE TIME
NON OR TWO-WAY
INTERLEAVED

1, 5 OR 7 PORTS
TO MEMORY

AVAILABILITY
CONTROL
UNIT

CENTRAL
PROCESSING
UNIT

INTEGRAL I/O CONTR.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

CENTRAL
PROCESSING
UNIT

INTEGRAL I/O CONTR.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

CENTRAL
PROCESSING
UNIT

INTEGRAL I/O CONTR.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

CENTRAL
PROCESSING
UNIT

INTEGRAL I/O CONTR.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |

| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

EXTERNAL I/O CONTROLLER

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

EXTERNAL I/O CONTROLLER

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

EXTERNAL I/O CONTROLLER

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

NOTES :
(1) ALL I/O CONTROLLERS HAVE 8, 12 OR 16 CHANNELS
(2) EACH CPU USES ONE CHANNEL FOR CONSOLE
(3) ALL I/O CONTROLLERS CAN DATA CHAIN

UNIVAC 1108 MULTI-PROCESSOR

DETAILED HARDWARE SPECIFICATION SHEET

MANUFACTURER: UNIVAC/SPERRY RAND CORP.

MACHINE SERIES: UNI 1100

SYSTEM CHARACTERISTIC	1106	1108	XDS SIGMA 9
MAIN MEMORY			
Size Range	384K – 1.5M Char	⟶	512K – 2M Bytes
Cycle Time	1.5µs	750 ns	850 NS
Access Width	2 36-bit words(12 char)	⟶	4 Bytes
Interleave	No	2-way Opt.	4 Way
Access Ports	1	1 – 5	2 – 12
Map	NA	NA	Std.
Protection	BAR/LAR – 2 sets		1st 512KB Only
Min. Increase	192KC	384KC	128KB
BUFFER MEMORY	NA	⟶	NA
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Protection			
LARGE CAPACITY STORAGE	Unitized Channel Storage	⟶	NA
Size Range	256K – 1MW	⟶	
Cycle Time	2.3ns	⟶	
Access Width	Word – 36 bits	⟶	
Interleave	NA	⟶	
Access Ports	1	⟶	
Xfer Rate	2.66MC	⟶	
Map	NA	⟶	
Protection	NA	⟶	
Exec. Instr. From	I/O Dev. only	⟶	
CENTRAL PROCESSING UNIT			
CPU Clock Cycle	166 ns		Instr. & Seq. Dep.
Look Ahead			12 Bytes Less Current Instr.
Instr. Prefetch	2-word – current instr	⟶	One Beyond Current
Levels of Decode	Current Instr	⟶	

COMPANY PRIVATE

MANUFACTURER: UNIVAC/SPERRY RAND CORP.

MACHINE SERIES: UNI 1100

SYSTEM CHARACTERISTIC	1106	1108	XDS SIGMA 9
● INSTRUCTIONS			
Commercial KIPS	44.6	89.2	64.7
Scientific KIPS	576.4	1011.1	763.4
Word Size	36 bits - 6 char	⟶	32 Bits or 4 Bytes
Instruction Size	36 bits	⟶	2 or 4 Bytes
Mult. Addr. Ops.	1	⟶	
Decimal	NA	⟶	Std.
Fixed Point	Std	⟶	Std.
Floating Point	Std	⟶	Sing. & Dbl. Std
Stor.-to-Stor.	NA	⟶	NA
Byte/Char. Manip.		⟶	Yes
Stack	NA	⟶	Yes
● ADDRESSING			
Indexing	Yes	⟶	Yes
Indirect	Yes - to any level	⟶	Yes
Virtual	NA	⟶	Map - 512KB/Prog.
Page Size			512KB
No. of Pages			
Scheme	Base displacement	⟶	Direct Reference
Size	6,9,12,18 or 36 bits	⟶	1, 2, 4 & 8 Byte
Max. Addr.	256KW	⟶	4MW or 16MB/ Extended Mode
● REGISTERS			
Index	15	⟶	General Purpose
Gen. Purpose	16 Arithmetic	⟶	2 Blks. of 16 Std.
Mult. Gen. Reg. Blks.	2-16/sys, 16/user	⟶	2 Std. & Opt.
Floating Point	Yes?	⟶	Single & Double
Decimal	NA	⟶	Std.
● INTERVAL TIMER	Time of day 600 ms	⟶	500 HZ & one of (50/60, 500, 2K, or 8K HZ) Switchable

MANUFACTURER: UNIVAC/SPERRY RAND CORP.

MACHINE SERIES: UNI 1100

SYSTEM CHARACTERISTIC	1106	1108	XDS SIGMA 9
● INTEGRAL MULTIPLEXER	Integ I/O Control	——→	NA
Channels	4-16	8-16	
Subchans./Controllers	1/channel	——→	
Aggregate Xfer	667KW	1.33MW	
MPX - Max Chan Xfer	?	?	
Burst - Max Chan Xfer	333KW	440KW	
Command Chaining	NA	——→	
Data Chaining	NA	——→	
Data Path Width	1 36-bit word	——→	
CPU Loading			
Multiplexing	?	——→	
Burst	?	——→	
Command Chain	?	——→	
Data Chain	?	——→	
Xfer in Channel	?	——→	↓
● EXTERNAL MULTIPLEXER	NA	Ext I/O Control	1 MIOP Std 11 Max
Channels		8-16	1 Std. 2 Max.
Subchans./Controllers		1/channel	8-32
Aggregate Xfer		1.33 MW	1.9MB W/4 Byte & 2 Chan.
MPX - Max Chan Xfer		?	470KB
Burst - Max Chan Xfer		250KW	
Command Chaining		NA	Yes
Data Chaining		Yes	Yes
Data Path Width	↓	1 36-bit word	1 or 4 Byte/Dev & 4 Byte to Mem.
● INTEGRAL SELECTOR	Capabilities/Integ I/O Control (See multiplexor)	——→	NA
Channels			
Subchan./Controllers			
Max. Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Xfer.			
Command Chain.			
Data Chain.	↓		↓

MANUFACTURER: UNIVAC/SPERRY RAND CORP.

MACHINE SERIES: UNI 1100

SYSTEM CHARACTERISTIC	1106	1108	XDS SIGMA 9
● EXTERNAL SELECTOR	NA	Capabilities/Ext I/O Control (see multiplexor)	1 - 4 RIOP's
Channels			1
Subchans./Controllers			1 Integral
Max. Xfer.			3MB
Command Chaining			Yes
Data Chaining	↓	↓	Yes
Data Path Width			4 Byte to Mem.
● SYSTEM BANDWIDTH			25.3MB
● AVAILABILITY			
Watchdog Timer	?		.Yes
Power Fail Safe	NA	⟶	Saves Volatile Regs. & Brings to Orderly Halt
Recoverability	NA	⟶	Yes
Relocatible Homespace	NA	⟶	Yes
Instruction Failure			
Pre. Exec. Test	NA	⟶	Yes
Isolation	NA	⟶	Yes
Retry	NA	⟶	
CPU			Yes
Channel			Yes

● PERIPHERALS

For low speed peripherals – card reader, punch and printer – Univac normally con-
figures a 9200 or 9300 Peripheral Processor Subsystem. There are higher speed periph-
erals available but at a much higher cost. Univac has a limited number of magnetic
tapes available. Their lower speed tapes are expensive compared with Sigma.

MANUFACTURER _____UNIVAC_____

MACHINE SERIES _____1106/1108_____

DEVICE		MODEL NO.	SPEED	PURCHASE	MAINT.	1 YR. LEASE
CARD READER	(1)	0711-02(9300)	600 cpm	$ 6,315	$ 60	$ 205
		0706-97	900 cpm	44,005	390	1,400
CARD PUNCH	(1)	0603-04(9300)	75 cpm	6,315	60	205
	(1)	0604-00(9300)	200 cpm	9,920	90	320
	(1)	0604-99(9300)	250 cpm	15,660	90	450
		0600-00	300 cpm	50,180	495	1,635
CARD READ/PUNCH		0706/0600	900/ 300 cpm	65,565	620	2,115
PRINTER	(1)	3030-02(9300)	600 lpm	64,860	255	1,740
		0758-00	1200/ 1600 lpm	73,515	505	2,135
MAGNETIC TAPE 7 track		UNISERVO VIC	34.2KC	79,830	360	2,190
		UNISERVO VIIIC	96 KC	193,370	560	4,990
9 track 800 BPI		UNISERVO VIC	34.2KB	81,790	365	2,240
		UNISERVO VIIIC	96 KB	197,950	565	5,100
	(2)	UNISERVO 12	34.2KB	82,220	415	2,305
	(2)	UNISERVO	96 KB	155,630	555	4,135
9 track 1600BPI	(2)	UNISERVO12	68.4KB	77,435	400	2,180
		UNISERVO16	192 KB	150,945	540	4,010

NOTES: All prices include controller price where required.
      Magnetic tape prices include price of controller and 4 drives.

(1) 9300 subsystem includes CPU, 8K memory, interface to 1106/1108 and a 600 lpm printer.
    Card reader and punches can be added – Subsystem as listed under printer is a prerequisite.
(2) UNISERVO 12 and 16 controller prices are not available. These prices are as listed for
    9400 System. Actual prices for 1106/1108 may be slightly higher.

- MASS RANDOM ACCESS STORAGE

Univac has a wide range of mass storage devices available, including a recently announced bulk core (unitized channel storage). Univac recommends the FH-432 Drum as a system storage and swapping device. Note the high cost and low transfer rate compared with our 7212.

Fastrand has been the standard file storage device for Univac since the days of the original RANDEX drum. In late 1969 Univac announced the 8414 Removable Disk pack (equivalent to our 7242). However, the current availability of software support for this device on the 1106/1108 is very doubtful.

RANDOM ACCESS DEVICES

MANUFACTURER   Univac

MACHINE SERIES   1106/1108

MODEL NO.	CAPACITY PER MODULE	KB RATE	AVE. ACCESS	CAPACITY	PURCHASE	MAINT.	1 YR. LEASE
SYSTEM STORAGE AND SWAPPING RADS							
FH-432	1.6MC	1.44MC	4.33Msec	4.7MC	$ 209,820	$ 560	$ 5,355
				9.4MC	337,125	860	8,565
				14.2MC	464,430	1160	11,775
FH-1782	12.6MC	1.44MC	17Msec	12.6MC	199,725	520	5,085
				25.2MC	316,935	780	8,025
				50.3MC	551,355	1300	13,905
				100.7MC	1,020,195	2340	25,665
FH-880	4.7MC	180MC	17Msec	4.7MC	143,965	380	3,700
				9.4MC	229,130	570	5,850
				18.9MC	399,460.	950	10,150
				37.7MC	740,120	1710	18,750
BULK CORE							
UCS	1.5MC	2.6MC	2.3μsec	1.5MB	270,240	(2)	5,630(2)
				3.0MB	540,480		11,260
				4.6MB	810,720		16,890
				6.1MB	1,080,960		22,580
FILE STORAGE (REMOVABLE)							
Fastrand II	132MC	154KC	92Msec	132MC	218,050	415	5,385
				264MC	382,690	715	9,435
				396MC	547,330	1015	13,485
				793MC	1,041,250	1915	25,635
				1056MC	1,370,530	2515	33,735
Fastrand III	198MC	230KC	92Msec	198MC	263,020	485	6,530
				396MC	463,820	835	11,495
				793MC	865,420	1535	21,425
				1586MC	1,668,620	2935	41,285
8414(1)	29.2MB	312KB	72.5Msec	29.2MB	55,200	180	1,330
				58.3MB	84,000	240	1,990
				116.7MB	138,720	350	3,240
				233.4MB	235,680	610	5,520
Multi-Subsystem Adapter (see NOTE)							
0961-02					25,680	65	600

NOTE: (1) 8414 can be attached to U1106 or 1108 in two ways, either via a 9300II peripheral subsystem or via a Multi-Subsystem adapter (MSA) direct to the U1106/1108. Software support is available for the 9300 but is questionable for the direct connection via MSA. Therefore, above prices are for 8414 if attached to 9300II. For direct connection add price of MSA as listed above.

(2) Maintenance prices are not known for Univac's Unitized Channel Storage (UCS). One year lease prices do not include maintenance.

• COMMUNICATIONS

Univac has two single line controllers and a multi-line controller in their product line. The single line controllers are synchronous only. The multi-line controller handles a maximum of 32 half or full duplex lines. There is limited control code recognition capability in the multi-line controller. Univac does not currently have a communications processor similar to the XDS systems CIOP. For very large communications networks a 418 III front-end system (approximate price equal to or greater than Sigma 5) may be required.

MANUFACTURER Univac

MACHINE SERIES 1106/1108

SYSTEM CHARACTERISTIC	WORD TERM SYNC	COMM TERM MODULE CONTROL	COMM TERM SYNC
Function	Single line contr.	Multi line contr.	Single line contr.
Programmable contr.	No	No	No
Memory	None	None	None
Data path interface	IØ CH	IØ CH	IØ CH
Number of lines	1 (2 opt)	32 HDX, FDX	1 (up to 6 per cabi-net)
Max. bit rate/line	up to 230,400 bps	up to 4800 bps	up to 40,800 bps
Aggregate bit rate	230,400 bps	?	40,800 bps
Rate sensing	No	No	No
Mode(s)			
HDX, FDX, ECHPLX	HDX, FDX	HDX, FDX	HDX, FDX
ASYNC, SYNC	SYNC	Both	SYNC
BISYNC	No	No	No
Data format	6 bit CHAR only	4 – 8 level	5 – 8 level
Code translation	None	None	None
SYNC pattern	2 to 10 CHAR Patch Bd Wiring	Fixed in adapter	?
Control code recog.	Patch Bd Wiring	Limited	?
Buffer size	36 bits	1 Char	1 Char
Interrupts	Message	Message	Message
Auto dial	Opt	Opt	Opt
Auto answer	Opt	Yes	Opt
Parallel data input	No	Yes	No
Voice answer back	No	No	No
Polling	No	No	No
Error check	Parity, BCC	BCC	Parity
NOTES:	Each WTS independent, i.e., no MPX-ing.		Each CTS is independent, i.e., no MPXing is performed

- REAL TIME

  Univac talks about multiple interrupt levels, real time capability, etc. However, they mean communications handling when they say real time. There is only one external interrupt level. Multiple real time interrupts must be handled <u>via software.</u>

MANUFACTURER: UNIVAC/SPERRY RAND CORP.

MACHINE SERIES: UNI 1100

SYSTEM CHARACTERISTIC	1106	1108	XDS SIGMA 9
● TIME CRITICAL REAL-TIME			
Priority Interrupts			
Levels	1?	⟶	224
Branch Address	Single LPS?	⟶	PSD SWAP
Arm/Disarm	NA?	⟶	Yes
Enable/Disable	Yes?	⟶	Yes
Triggerable	NA?	⟶	Yes
Clocks	R-T clock 200 μs	⟶	2-STD, 2-OPT (500HZ and 50/60, 500, 2K, or 8K HZ, or Other - Switchable)
Context Switch			
Exec. Time to Change Control	LPS + LSL + ER ⟶ 5.33 μs	LPS + LSL + ER ⟶ 2.85μs	XPSD 5.43 μs
Mult. Reg. Blks. or Load/Store Mult.	Up to six double Load/Store? - 16μs	Up to six double Load/Store? - 9.0μs	Mult. Reg. Blks.
Instrumentation	RPQ	⟶	Extensive List of Standard Products in Addition to Separate Systems Division Product Line.

SOFTWARE

- **EXEC 8**

  Exec 8 is the primary operating system for the Univac 1106/1108. Exec 8 provides batch multiprogramming for a variable number of jobs (number in a system generation) – specializing in scientific processing and wide support for remote batch terminals. The system has a very sophisticated scheduling method based on dynamic allocation of resources to fit the job stream. Utilities are integrated into the system and applications packages, free to customer, are oriented toward the scientific customer.

  Real time in the form of communications support is provided with highest priority within the monitor. Time-sharing is referred to as "demand processing" and as such is handled either for remote job entry much like a remote batch terminal or an interactive manner when connected to an interactive language processor. The time-sharing language processors are reentrant and swapping has a high priority within the system.

- **UNIVAC SOFTWARE STRENGTHS**

  - <u>High Speed Processing/Turnaround:</u> This is basically a quality of the computer and not the operating system. However, Exec 8 provides a very intricate scheduling mechanism which allows the user to provide a "deadline" time which identifies when the job must be completed.

  - <u>Efficient Multiprogramming:</u> Univac stresses the variable number of jobs that can be processed concurrently. The intention of this is to give the idea that an installation can throw any combination of jobs at it and Exec 8 figures out the best way to do them. Operator scheduling is kept to a minimum.

  - <u>Batch Scheduling:</u> Exec 8 provides very unusual techniques for control of job processing. Presumably, this results from Univac attempting to overcome IBM's OS scheduling inadequacies. For example, each job is given a unique quantum of time that is a function of many variables; resources required, deadline time, how long it has been in the system, its usage of CPU so far, etc., etc. At particular times, new quantums may be established dynamically to force a particular situation.

  - <u>Communications:</u> Remote batch terminals are the main form of communication with support for other types of terminals also.

  - <u>Scientific Processing:</u> Univac provides an extended version of FORTRAN called FORTRAN IV. Many features are supported that are similar to those of XDS FORTRAN IV. Other languages, ALGOL and JOVIAL, are scientific oriented plus sets of routines called Math Pack and STATPACK. Applications programs support the scientific processing through APT, LP, GPSS II and Similar.

- EXEC 8 WEAK POINTS

  - Time-Sharing: Due to the structure of the batch scheduling, when time-sharing scheduling was retro-fitted, apparently there was severe interference imposed. Demand scheduling (as the time-sharing users are referred to) imposes another level of priority (real time is highest, demand users are next, then comes batch) and as such controls the system. Queues for swapping are established and queues for programs that are currently resident are established. CPU time is allocated on the basis of minimum and maximum percentage parameters. Thus, an installation can say I always want 15% of CPU for TS but never to go above 60%. This should provide for a fairly substantial batch stream. Apparently it does not! Several reports have come from across the country relating that in the 1106, batch dries up when 6 to 10 terminals are active and the 1108 batch dries up when under 15 terminals are active. The problem is suspected to be in two areas: dynamic scheduling overhead increases drastically due to the imposition of more users (more active programs that must be handled); core space cannot remain allocated to a particular batch program long enough to make substantial headway — this is because TS takes on a higher priority.

  - Commercial Data Processing: Fundamentally, the 1106/1108 are not designed for commercial DP — due to the lack of decimal instructions, etc. (Von Newmann machine) Beware of pointing fingers at this inadequacy because, due to the inherent processing powers of the system, their COBOL compiler is very fast and due to the long-term stability of the system (the basic instruction set has not changed for 10 to 15 years) a very adequate and competitive execution capability exists. This means that they have very efficiently organized decimal interpretive subroutines that do their work well in combination with the compiler.

- SIGMA 9 STRENGTHS — UTS

  All of the time-sharing features of UTS are superior to Exec 8 time-sharing. It is expected that Univac will improve on their time-sharing capability over the next years, but even then they would be forced to create a new operating system to come anywhere near to the Sigma 9 and UTS capability. Some of the unique strengths of UTS over Exec 8 in a time-sharing environment include:

  - Up to 128 users with UTS. The best Univac has been able to do with Exec 8 on the 1108 is about 12 users (3 or 4 on the 1106)

  - UTS overhead is much lower than Exec 8's, providing superior response and operating out of a much smaller resident space (80KW vs 131KW plus with Exec 8)

  - UTS allows the system manager to have a higher degree of control over the operating system parameters than is possible in Exec 8

  - UTS provides a financial charge rate structure which allows precise customer/department billing and accounting

- UTS supports critical real time; Exec 8 cannot support real real-time

- SIGMA 9 STRENGTHS — XOS

  This is the primary batch-oriented competitor to Exec 8 that we offer on Sigma 9.

  - Time-Sharing: XOS offers a substantially better TS capability due to the inherent design of the system and the lack of competition for resources between batch and time-sharing.

  - Batch:

    a. Memory Management — The programs in Exec 8 must be contained in contiguous core. Real time programs are assigned a resident location during their execution. Batch programs enter the system, are assigned contiguous core wherever available, and then activation is under control of the dynamic allocator. Should a program be considered for execution and core is available but not contiguous, a compression cycle is made to move all programs together and provide the space. Contrast this to XOS memory management by paging so that no compression cycle is necessary and programs do not require contiguous space — neither batch or real time!

    b. Scheduling/Resource Allocation — In XOS, priority of execution is determined by external interrupts which are under the control of the scheduler. This is a very clean way plus there is little·overhead involved. Contrast this to the complicated formula allocation and quantum allocation of Exec 8 which is cumbersome and adds overhead. Exec 8 provides dynamic allocation of resources according to system generation parameters and job control cards. Since this is dynamic, Exec 8 is continuously re-examining itself to see if dynamic allocation is necessary which again contributes to overhead. XOS provides a very clean allocation by priority by class of job and by resources available under installation established constraints (no. of tapes, etc.) most of which are under operator control. When a job needs the resources they are there, otherwise they are used by other jobs.

- OTHER SIGMA 9 STRENGTHS — XOS

  Many of the features in XOS and Exec 8 are similar since they are basically similar to Exec II (BPM and Exec II have very similar features). Embedded into XOS are the parallel processing functions which Exec 8 does not have as such. Symbionts are similar for the two systems.

	UTS	UNIVAC 1106/ 1108 – Exec 8
Operational Modes –		
. Dedicated time-sharing (max no. of terminals)	Yes (128)	Possible (unknown)
. Time-sharing/Batch	Yes	Yes
. Time-sharing/Batch/Remote Batch	Yes	Yes
. Real time (combined with other modes)	Yes	Yes
Installation Control Features –		
. System Control		
– Performance Management*	Extensive	Good
. User control		
– Accounting per user*	Yes	Yes
– Central account authorization*	Yes	Yes
– Processor authorization by account*	Yes	No
– Terminal Control		
Passwords	Yes	Yes
Automatic connection to processors* (Restricted use)	Yes	No
Automatic connection to charge structures* (Unique charges based on usage)	Yes	No
Shared processors (Reentrant)*	Yes	Unknown
– Batch control		
Controlled resource usage (core, priority, devices, etc.)*	Yes	Some
Batch time quantum control	Yes	Extensive
Percentage of CPU for batch	Yes	Yes
– Remote Batch Control	Same as batch	Good
– Real time control	Yes	Yes
. Operator Control		
– Core size for terminal users	Yes	No
– Number of terminal users	Yes	Yes
– File space	Yes	No
– Number of tapes/user	Yes	Yes
– Percentage of CPU for batch	Yes	Yes
– Terminal time quantum	Yes	Restricted
– Compute bound quantum	Yes	Restricted
– Batch priority relation to terminal	Yes	Yes
Program/Application Development Features –		
. Terminals		
– Conversational processors	Basic Extended XDS FORTRAN IV EDIT	FORTRAN

	UTS	UNIVAC 1106/ 1108 – Exec 8
- Interactive Languages	META-SYMBOL	Unknown
- Media conversion	PCL	Yes
- TJE to batch stream	Yes	Yes
- Processor compatibility with batch	Yes	Yes
- File interchangeability	Yes	Yes
- Terminals can be logged on for particular processors	Yes	No
- Sharable processors*	Yes	Yes
- Debug facilities (conversational)	FDP, DELTA	Some
. Batch/Remote Batch		
- Language Processors Standard FORTRAN, COBOL	Yes	Yes
Others		ALGOL, JOVIAL
- Other processors	SL-1, FMPS, DMS, GPDS	Many
. Real Time		Yes
- Languages	FORTRAN, MS	Unknown
- Other processors	SL-1	Unknown
- Terminal program development	Yes	Yes
. Terminal		
- Terminal job entry (under priority control)	Yes	Yes
- Compiling not required for each run (load modules maintained)	Yes	Yes
- Output optionally permanent	Yes	Yes
- Terminal must stay connected	Yes – time-sharing No – Batch	Unknown
. Batch/Remote Batch		
- Priority scheduling	Yes	Yes
- Standard system files available	Yes	Yes
- Non-standard files available	Yes	Yes
- Printer forms control	Limited	Yes
- Checkpoint/Restart	No	Yes
. Real time		
- Initiated by terminal	Optional	Optional
- Initiated by batch/remote batch	Optional	Optional
- Initiated by interrupt	Optional	Yes
- Initiated by operator	Optional	Yes
- Preempts core	Possible	Possible
- Highest priority	Yes	Yes

---

* UTS Reference Manual covers these topics in detail. Additional information is contained in UTS Specification No. 702489.

Manufacturer: UNIVAC
Operating System: Exec 8
Computer System: U1106/1108
Operating System Classification: Multi-Use

	XOS	Exec 8
Operational Modes —		
. Batch/Remote Batch	Yes	Yes
. Dedicated Time-Sharing (max no. of terminals)	Yes (variable)	Yes (Unknown)
. Batch/Remote Batch/Time-Sharing	Yes	Yes
. Real Time	Yes	Yes
Batch/Remote Batch Multiprogramming —		
. Minimum System Requirements	48KW	96KW(1106),128KW(1108)
. Number of Concurrent Programs	7 (R.T. included)	Variable
. Remote Batch is Part of Monitor	Optional	Yes
. Symbionts Are Part of Monitor	Yes	Yes
. Scheduling Criteria	Priority, Avail Resources	Very complex
. Checkpoint/Restart	Yes	Yes
. Disk Pack Support	Yes	Unknown
. Memory Management		
- System	Mapped	Dedicated
- User	Mapped	Contiguous (Compressable)
. Processor Support	FLAG, FORTRAN, COBOL, SORT/MERGE	FORTRAN, COBOL ALGOL, JOVIAL
. Methods of Job Entry	CR, Tape, Disk, RBT	CR, Tape, Disk, RBT
. CPU Time Allocation	Priority	Complex
Communications Support —	Character and Msg oriented	Same
Time-Sharing —		
. Languages	All Standard	See T/S Comparison (preceding pages)
. Performance Management	Yes	
Real Time Support —	Full	Yes
Applications —	DMS, GPDS, SL-1, FMPS	See T/S Comparison (preceding pages)

SYSTEM PRICES AND POLICIES

- PRICES

    The Univac 1106 is priced between the Sigma 7 and the Sigma 9. Note, however,
    that the performance potential of the 1106 is 70 to 75% that of the Sigma 9. The
    1108 is considerably more expensive than Sigma 9. Univac has been heavily dis-
    counting the 1108, particularly to universities. They permit both the long-term lease
    and the educational discount to apply which amounts to a 40% reduction.

● POLICIES

- Univac's basic lease contract provides for unlimited use of equipment.

- Univac excludes maintenance from basic rental charges. To get total monthly rental, maintenance charges must be added to the basic rental price (all quoted prices include maintenance).

- For the 1106/1108, Univac has two long-term lease plans. Quoted discounts apply only to basic rental price, exclusive of maintenance:

  1. Level payment plan:

3-year lease	10% discount
4-year lease	17.5% discount
5-year lease	25% discount

  2. Reducing payment plan: Averages out to same as level plan:

	Payment Rates (% of Monthly Rent)		
Payment Schedule	3-Year Lease	4-Year Lease	5-Year Lease
1st Year	95%	90%	85%
2nd Year	90%	85%	80%
3rd Year	85%	80%	75%
4th Year	N/A	75%	70%
5th Year	N/A	N/A	65%

- For the 9200 or 9300 peripheral subsystem:

  1. Level payment plan:

4-year lease	12.5% discount
5-year lease	15% discount

  2. Reducing payment plan:

Payment Schedule	4-Year* Lease	5-Year Lease
1st Year	95%	95%
2nd Year	90%	90%
3rd Year	85%	85%
4th Year	80%	80%
5th Year	N/A	75%

---

* Applies only to equipment which has been on an "on rent" status for 12 or more months.

- Educational Discount: Univac allows an educational allowance which is in addition to any discount obtained via long-term lease agreement.

  | 1106/1108 | 15% discount |
  | 9200/9300 | 10% discount |

- Purchase Option: Univac allows a purchase option credit of 75% of the 1st year's monthly lease price (exclusive of maintenance). This option may be exercised at any time providing the customer retains title to the equipment for a period of one year after exercize of the option.

  Univac has not gone to separate pricing of software program products or analyst support. They will, as a matter of policy, provide full systems engineering for the design through to the flow charting (but not program coding) of the customer's application.

PURCHASE PRICE PROCESSOR, MEMORY AND I/O (MIOP)

A chart titled "PURCHASE PRICE PROCESSOR, MEMORY AND I/O (MIOP)". The vertical axis is labeled "PURCHASE PRICE THOUSAND DOLLARS" ranging from 200 to 5400 in increments of 200. The horizontal axis is labeled "STORAGE CAPACITY K BYTES, K = 1024" ranging from 128 to 4096.

Curves shown:
- U1108 WIOC-4 CHANNELS
- U1108
- SIGMA 9
- U1106

RENTAL PRICE PROCESSOR,
MEMORY AND I/O (MIOP)

STORAGE CAPACITY
K BYTES
K = 1024

30
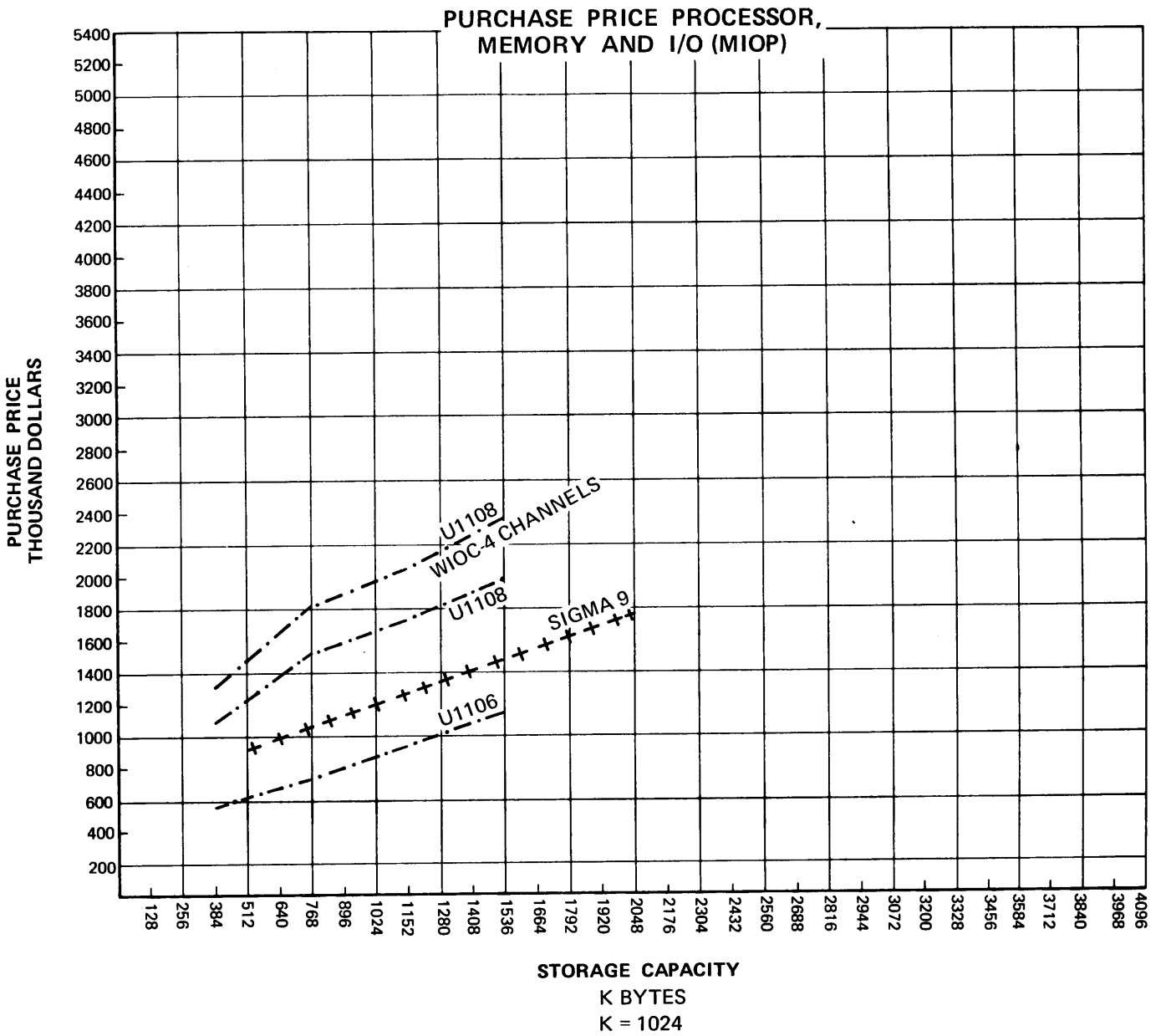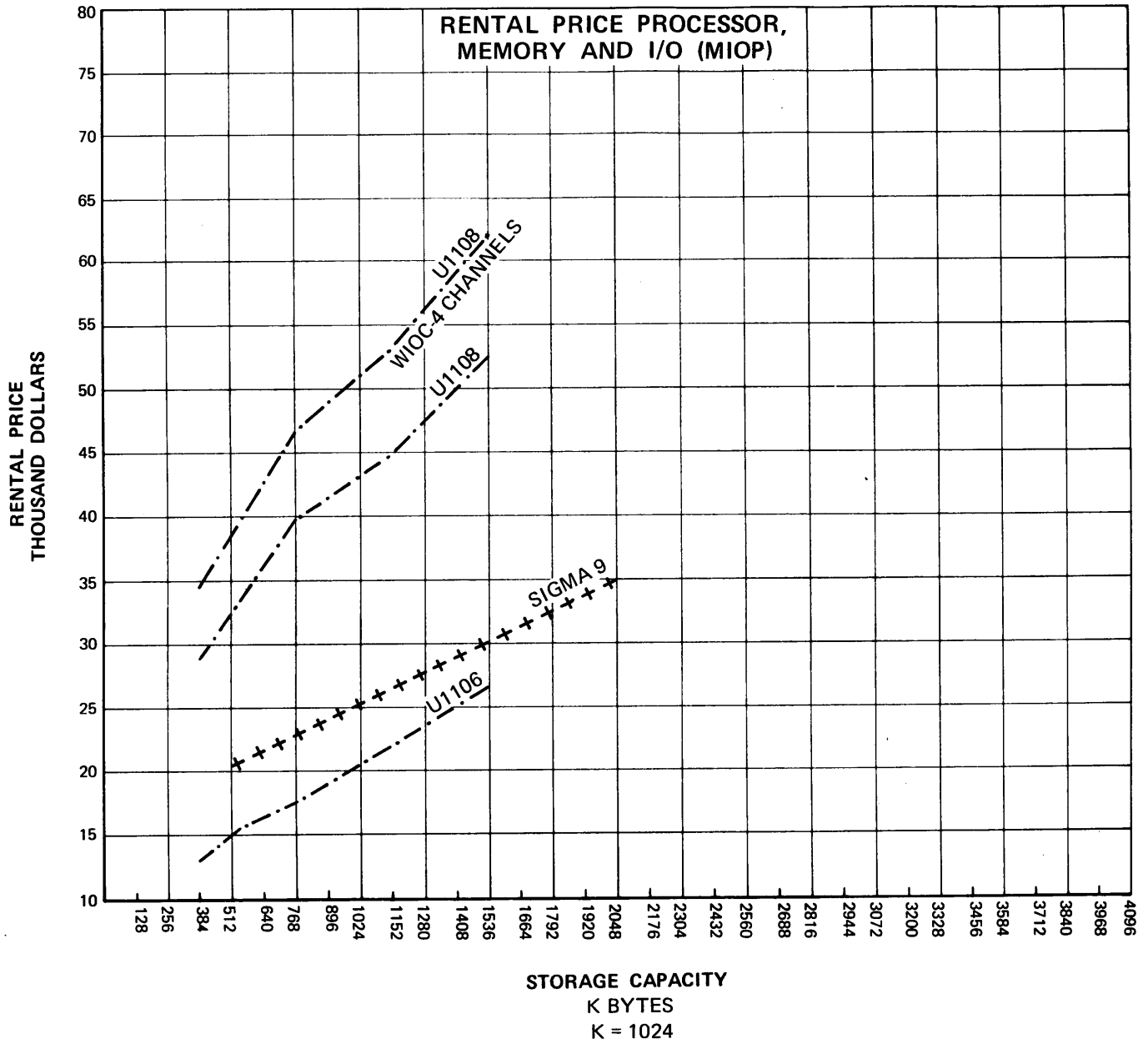
Competitive Sales Guide

OCTOBER 1970   COMPETITIVE ANALYSIS   File: CDC
                                                      6200/6400

Contained in this Sales Guide is an assessment of the CDC 6200 and 6400 computer systems versus Sigma 9. The CDC 6600 should not be considered a competitor to the Sigma 9 at this time because of its size.

CONTENTS

Please refer to the Sigma 9 Overview (File: XDS) Sales Guide to compare the configuration diagrams of the CDC 6200 and 6400 to Sigma 9 and also refer to the other Sales Guides on Sigma 9 competitors for additional information.

Distribution: Sigma 9 Announcement   Kent L. Cootes, Manager
Distribution and Competitive Infor-   Competitive Analysis
mation Reference Libraries   XDS   AFC

COMPANY PRIVATE

Xerox Data Systems

SYSTEM SUMMARY

- ## INTRODUCTION

  The CDC 6200/6400 are the two smaller members of the 6000 Series, a Series of four machines designed primarily for scientific number crunching. The 6600, the first member of the family announced, was designed around the computational needs of the atomic energy industry. As such, it features high speed extended precision (60-bit word) floating point operations, with 10 arithmetic functional units for parallel executions, and 10 peripheral and control processors which handle control of the system as well as all I/O activities. The 6200 and 6400 both have sequential arithmetic functional units, unlike the 6600. The 6600, with first delivery in August 1964, is nearing obsolescence. The 6400 and, more recently, the 6200 are attempts to extend the market (and the life) of the 6000 Series.

  Key differences between the family members are:

Model No.	1st Delivery	CPU Cycle Time	Memory Size	Memory Cycle Time	CPU Func. Units	Peripheral Processes	Data Channels
6600	August 1964	100 Msec	138 KW	1.0μs	10	10	12
6400	May 1966	100 nsec	32-188KW	1.0μs	1	7-10	9-12
6500	Dec. 1967	100 nsec	65-128KW	1.0μs	2-6400 CPU's	10	12
6200	1970	140 nsec	32-64KW	1.4μs	1	7-10	9-12

  While the 6600 is almost exclusively a scientific number cruncher, CDC is attempting to market the 6200 and the 6400 in mixed commercial/scientific environments.

- **CDC 6200/6400 STRENGTHS**

    - Scientific Processing Power: The CDC 6200 and 6400 offer extremely fast floating point operations plus a long precision word 60 bits. These systems will take heavy advantage of the reputation of the 6600 in scientific environments. This strength is considerably weakened by the lack of more general purpose instructions in the instruction repertoire, for example fixed point multiply divide, decimal and byte string instructions, radix conversion, search or table look up facilities, etc.

    - Concept/Design: The most unique feature of the CDC 6200/6400 is the inclusion in a basic system of from 7 to 10 peripheral and control processors. These processors function as high speed I/O Buffers, Monitor Control processors, etc. Effectively they free the CPU from the mundane work permitting it to concentrate on that which it was designed for – i.e., high speed scientific calculations.

    - Multi-Programming: The normal mode of operation on the 6200/6400 is multi programmed batch using the SCOPE operating system. Scope can handle 7 batch/remote batch jobs.

    - Time Sharing: The KRONOS Time Sharing operating system will handle up to 384 active TTY terminals plus up to 16 RBT's plus local batch. However, there is some question as to the degree of interaction actually provided by their "Time Sharing" system. Examples included in CDC's literature seem to indicate that except for a Text Editor, all processors actually function in a Remote Batch Mode. KRONOS handles a variable number of jobs up to 32 maximum in core at one time.

    - High Speed Memory Swap: CDC pushes the use of Bulk Core as a Swap device for time sharing. Bulk core direct to Main Memory data transfers are 480-bits (8 words) wide, and with full interleaving have a transfer rate of up to 10.4 Megacharacters per second. Parity protection is optionally available for Bulk Core only. Note that Bulk Core functions as a file device only, no instruction execution is permitted.

    - High Speed Context Switching: According to Auerbach, CDC can perform a complete register exchange in less than 5 μsec. We have no verification of this so question it if it becomes a point.

- **CDC 6200/6400 WEAKNESSES**

    - Time Sharing Software: KRONOS is a recently announced operating system with no record of success to date and little information available. We suspect that it has limited interactive/conversational capability.

- Reliability: The overall reliability of the system is suspect. There is no parity checking on any data transfers into, out of or around main memory. KRONOS/SCOPE are required to verify check sums continuously due to the lack of parity checking within the system. Power Fail Safe is doubtful. There are no recoverability features available, no specificial facilities to improve the reliability, availability of the systems.

- No Decimal/Byte String Instructions: In fact, there are few "general purpose" instructions available. The system is designed for floating point arithmetic and little else is provided.

- No Real Time: There are no interrupts. Real Time probably required a 1700 front end for real time data acquisition.

- Costs: These systems are extremely expensive compared to the Sigma 9.

HARDWARE

Some of the highlights of the CDC 6200/6400 hardware features include:

- MEMORY

    - 1.0 microsecond memory cycle on the 6400.
      1.4 microsecond memory cycle on the 6200.
    - 32, 65 or 128KW memory on the 6400.
      32 or 65KW memory on the 6200.
    - Word Size 60 bits.
    - 8 way interleaving with 32KW
      10 way interleaving with 64 or 128KW
    - Memory Protection provided by Base and Limit address Register.
    - No parity.

- CPU

    - Sequential Arithmetic Functional Unit
    - No Decimal instructions.
    - No Fixed Point Multiply or Divide.
    - High Speed Float Point operations. Extended precision – 11 bit exponent, 48 bit coefficient operands.
    - No interrupts, the peripheral and control processors control the system by monitoring I/O and CPU activity. See Peripheral processor highlights for further discussion.
    - CPU contains eight operand registers, eight address registers and eight index registers.
    - Two 60-bit instruction registers for overlap of fetch and execution.
    - Instruction size is either 15 or 30 bits depending on the type of instruction thus 2, 3 or 4 instructions are contained in an instruction register.

6200/
6400

32 KW MIN AND INCREMENTS | 64 KW INCREMENT | 6400 ONLY

MEMORY -

6200 ≈ 1.4 μS CYCLE FOR
60 BIT WORD ACCESS
MAX CORE SIZE
64 KW OR 640 KC

6400 = 1.0 μS CYCLE FOR
60 BIT WORD ACCESS
MAX CORE SIZE
128 KW OR 1.3 MC

◄— MODULARITY - MEMORY IS BROKEN
UP INTO 4 KW BANKS EACH WITH

◄— ACCESS PORTS - 10 MEMORY ACCESS
PORTS PER 4 KW BANK

MEMORY ACCESS
SEQUENCE SWITCH

◄— SINGLE ADDRESS RESOLUTION SWITCH SEQUENCES
10 MINOR CYCLE ACCESSES ≈ OVERLAPPED

o ---- o ◄— EXTENDED CORE STORAGE

CENTRAL
PROCESSING
UNIT

PPU   PPU   PPU   PPU   PPU   PPU   PPU   PPU   PPU   PPU

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | DATA CHANNELS

NOTES :

PPU   1) PERIPHERAL PROCESSING UNIT - 4 KW (12-BIT WORD)
COMPUTER BUFFERS ALL I/O ACCESSES TO AND FROM
MEMORY. USED FOR CONTROL PURPOSES ALSO

CDC 6200/6400

MANUFACTURER: CONTROL DATA CORP

MACHINE SERIES: CDC 6000

SYSTEM CHARACTERISTIC	MODEL 6200	MODEL 6400	XDS SIGMA 9
MAIN MEMORY			
Size Range	32-128KW/320K-1.3MChar ⟶		512K - 2M Bytes
Cycle Time	1.4 μs	1.0 μs	850 NS
Access Width	1-60 Bit Word ⟶		4 Bytes
Interleave	8 way/32KW & 10 way/64K ⟶		4 Way
Access Ports	10 ⟶		2 - 12
Map	N/A ⟶		Std.
Protection	Std ⟶		1st 512KB Only
Min. Increase	320 KC ⟶		128KB
BUFFER MEMORY	N/A	N/A	NA
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Protection			
LARGE CAPACITY STORAGE	Extended Core Storage	⟶	NA
Size Range	250K - 2M Words	⟶	
Cycle Time	3.2 μs	⟶	
Access Width	80 Char	⟶	
Interleave	4-Way	⟶	
Access Ports	4	⟶	
Xfer Rate	1 to 10.4 MC	⟶	
Map	N/A	⟶	
Protection	Parity-Opt/Protection-Std	⟶	
Exec. Instr. From	I/O(Swap) Device only	⟶	
CENTRAL PROCESSING UNIT			
CPU Clock Cycle	140 NS	100 NS	Instr. & Seq. Dep.
Look Ahead			
Instr. Prefetch	1 Word(2, 3, or 4 Instr)	⟶	12 Bytes Less Current Instr.
Levels of Decode	Current Instr Only	⟶	One Beyond Current
Operands	N/A	⟶	

MANUFACTURER: **CONTROL DATA CORP**

MACHINE SERIES: **CDC 6000**

SYSTEM CHARACTERISTIC	MODEL 6200	MODEL 6400	XDS SIGMA 9
● INSTRUCTIONS			
Commercial KIPS	71.8	95.7	64.7
Scientific KIPS	594.9	793.0	763.4
Word Size	60 Bits or 10 Char		32 Bits or 4 Bytes
Instruction Size	15 & 30 Bit	⟶	2 or 4 Bytes
Mult. Addr. Ops.	3-OPS (Regs only)	⟶	
Decimal	NA	⟶	Std.
Fixed Point	Add/Sub Only	⟶	Std.
Floating Point	Std	⟶	Sing. & Dbl. Std
Stor.-to-Stor.	NA	⟶	NA
Byte/Char. Manip.	NA	⟶	Yes
Stack	NA	⟶	Yes
● ADDRESSING			
Indexing	Yes	⟶	Yes
Indirect	NA	⟶	Yes
Virtual	NA	⟶	Map - 512KB/Prog.
Page Size			512KB
No. of Pages			
Scheme	BAR/LAR	⟶	Direct Reference
Size	Half & Word	⟶	1, 2, 4 & 8 Byte
Max. Addr.	256 KW or 2.6 MC	⟶	4MW or 16MB/ Extended Mode
● REGISTERS			
Index	8	⟶	General Purpose
Gen. Purpose	8(Addr) & 8(Operand)	⟶	2 Blks. of 16 Std.
Mult. Gen. Reg. Blks.	NA	⟶	2 Std. & Opt.
Floating Point	1-98 Bit Reg	⟶	Single & Double
Decimal	NA	⟶	Std.
● INTERVAL TIMER	External Adaption per I/O Chan Bus	⟶	500 HZ & one of (50/60, 500, 2K, or 8K HZ) Switchable

MANUFACTURER: CONTROL DATA CORP

MACHINE SERIES: CDC 6000

SYSTEM CHARACTERISTIC	MODEL 6200	MODEL 6400	XDS SIGMA 9
● INTEGRAL MULTIPLEXER	NA	NA	NA
Channels			
Subchans./Controllers			
Aggregate Xfer			
MPX - Max Chan Xfer			
Burst - Max Chan Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Multiplexing			
Burst			
Command Chain			
Data Chain			
Xfer in Channel			
● EXTERNAL MULTIPLEXER	7-10 Periph Proc Units		1 MIOP Std 11 Max
Channels	12 Floating		1 Std. 2 Max.
Subchans./Controllers	1/Chan		8-32
Aggregate Xfer	12 MC		1.9MB W/4 Byte & 2 Chan.
MPX - Max Chan Xfer			470KB
Burst - Max Chan Xfer	2 MC	⟶	
Command Chaining	Yes	⟶	Yes
Data Chaining	Yes	⟶	Yes
Data Path Width	12 Bit/Dev & Mem	⟶	1 or 4 Byte/Dev & 4 Byte to Mem.
● INTEGRAL SELECTOR	NA	NA	NA
Channels			
Subchan./Controllers			
Max. Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Xfer.			
Command Chain.			
Data Chain.			

MANUFACTURER: CONTROL DATA CORP

MACHINE SERIES: CDC 6000

SYSTEM CHARACTERISTIC	MODEL 6200	MODEL 6400	XDS SIGMA 9
o EXTERNAL SELECTOR	See External Multiplexer		1 - 4 RIOP's
Channels			1
Subchans./Controllers			1 Integral
Max. Xfer.			3MB
Command Chaining			Yes
Data Chaining			Yes
Data Path Width			4 Byte to Mem.
o SYSTEM BANDWIDTH			25.3MB
o AVAILABILITY			
Watchdog Timer	Yes	⟶	Yes
Power Fail Safe	?	?	Saves Volatile Regs. & Brings to Orderly Halt
Recoverability	?	?	Yes
Relocatible Homespace	NA	No	Yes
Instruction Failure			
Pre. Exec. Test	NA	⟶	Yes
Isolation	NA	⟹	Yes
Retry	NA	⟹	
CPU		⟶	Yes
Channel			Yes

● I/O AND PERIPHERAL PROCESSORS

- From 7 to 10 Peripheral Processing Units (PPU's) control all I/O activity, perform executive and monitor control services for entire system, perform time consuming operations such as file searching, array manipulations, and schedule and exchange operating programs in the CPU.

- One Peripheral Processor is reserved for the monitor, one for control of the operator's display console.

- 4K (12-bit) words of memory per CPU. 1 μsec cycle time.

- PPU's act as a buffer for the CPU but are not themselves buffered, i.e, each PPU can be accepting data from an I/O channel, operating on a program, or transmitting to main memory. One and only one operation per PPU.

- System control is effected via an exchange jump instruction executed by a PPU. This causes a CPU interrupt and, CPU registers to be exchanged.

- Exchange jump exchanges all registers to switch from one program to another in approximately 5 μsec.

- I/O System: Twelve I/O channels accessible by any PPU.

- I/O Bandwidth: 1 megacycle rate per channel (2 megacharacters), all channels may be active simultaneously.

- No parity checking on data transfers.

● PERIPHERALS

There are relatively few peripheral devices available for the CDC 6000 Series. Most which are available are devices designed for the 3000 Series and require a data channel converter for the 6000 Series. As a result, peripheral prices for the 6000 Series are extremely high compared to the Sigma 9 (and, in fact, compared to almost all other computers). The following chart summarizes the peripheral products and provides prices. Controller prices are included where required. Magnetic tape price is the price of a controller and four drives.

● MASS RANDOM ACCESS MEMORY

CDC offers a range of random access devices. Again, they are characterized by comparatively high prices. Bulk core has been included for comparison purposes. CDC recommend the use of bulk core as a swapping device for time sharing systems.

PERIPHERALS - W/PRICE FOR COMPARISON

MANUFACTURER   **CDC**

MACHINE SERIES   **6200/6400/6600**

DEVICE		MODEL NO.	SPEED	PURCHASE	MAINT.	1 YR. LEASE
CARD READER		405	1200 cpm	$ 37,630	$ 120	$ 680
CARD PUNCH		415	250 cpm	44,520	125	815
PRINTER	Train	512	1200 lpm	79,500	270	1,760
DATA CHANNEL CONVERTER (See Note)		6681	N/A	14,840	40	340
MAGNETIC TAPE	7 track	608	30 KC	101,230	537	2,347
		604	60 KC	155,820	637	3,417
		607	120 KC	225,780	717	4,637
	9 track 800 BPI	659-1	30 KB	130,515	537	2,827
		659-2	60 KB	161,255	657	3,487
		659-3	90 KB	208,955	797	4,527
		659-4	120 KB	231,855	837	5,067
	9 track 1600 BPI	659-1	60 KB	137,800	552	3,007
		659-2	120 KB	168,540	672	3,667
		659-3	180 KB	216,240	812	4,707
		659-4	240 KB	239,140	852	5,247

NOTE: All prices include controller price where required.
Magnetic tape prices include price of controller and 4 drives (and Data Channel Converter)

Model 6681 or Model 6684 Data Channel Converter is required for all Series 3000 peripherals (which includes almost all devices available for 6000 Series). Cost of converter has been included in price of tape. However, card reader, punch and printer are buffered and standard configuration is one 6681 per the three low speed devices. Price of 6681 was <u>not</u> included for these devices.

MANUFACTURER   CDC

MACHINE SERIES   6000 Series

MODEL NO.	CAPACITY PER MODULE	KB RATE	AVE. ACCESS	CAPACITY	PURCHASE	MAINT.	1 YR. LEASE
SYSTEM STORAGE AND SWAPPING RADS							
865	8MC	1MC	16.7Msec	8MC	$ 205,110	$ 555	$ 4,145
				16MC	358,810	975	7,095
				33MC	666,210	1815	12,995
				66MC	1,281,010	3495	24,795
6638-2	84MC	1.68MC	162Msec	84MC	233,200	708	4,415
				168MC	466,400	1416	8,830
BULK CORE							
663X	1.3MC	25MC	3μsec	1.3MC	371,000	960	9,325
		50MC		2.5MC	593,000	1345	14,715
		100MC		5.0MC	1,038,800	2020	25,610
		100MC		10.1MC	1,929,200	3020	47,410
FILE STORAGE (NON REMOVABLE)							
821-1	419MC	420KC	117Msec	419MC	193,980	611	4,275
				838MC	377,080	1081	7,380
821-2	838MC	420KC	117Msec	838MC	315,880	941	6,860
FILE STORAGE (REMOVABLE)							
841	35MC	416KC	87.5Msec	107MC[1]	143,100	410	3,315
				214MC	216,240	560	4,995
				286MC	254,400	660	5,850

NOTES: 1. MINIMUM configuration is three drives plus a spare.
2. Above prices do not include cost of Data Channel converter as it is not known which devices require it. For your information prices are included on preceding peripheral chart.

- COMMUNICATIONS

  The standard communications equipment for the CDC 6000 Series is characterized by a lack of flexibility. They have three multiplexers, one handles 2 or 4 Telpak (40.8K bps) lines, one handles a mix TTY and CRT Display terminal and one handles only Model 33 and 35 TTY. The standard communications equipment available for the 3000 Series can be interfaced to the 6000 Series via the data channel converter. This equipment has considerably more flexibility in line speed options, terminals supported, etc., however, it is extremely expensive and not competitive price wise with any competitive equipment including Sigma.

- CRITICAL REAL TIME

  CDC does not have standard real time support capability. A to D converters available for the 3000 Series can be interfaced via a Data Channel converter. However, because of the complete lack of interrupts, priority or otherwise, one PCP would probably have to be dedicated to each line required – a fairly expensive way to handle real time. CDC's probable approach to real time requirements would be to provide a 1700 front end for real time data acquisition and use the 6000 Series for data reduction.

MANUFACTURER CDC

MACHINE SERIES 6200/6400

SYSTEM CHARACTERISTIC	6673/74	6671	6676
Function	Multi line contr.	Multi line contr.	Multi line contr.
Programmable contr.	No	No	No
Memory	None·	None	None
Data path interface	I/O CH	I/O CH	I/O CH
Number of lines	2 or 4	16	64
Max. bit rate/line	40.8K bps	110 bps or 2400 bps	110 bps
Aggregate bit rate	?	?	?
Rate sensing	No	No	No
Mode(s)			
HDX, FDX, ECHPLX	HDX	HDX, FDX	?
ASYNC, SYNC	SYNC	Either	ASYNC
BISYNC	No	No	Not applicable
Data format	?	8 bit	8 bit
Code Translation	None	?	None
SYNC pattern	?	Output – progr. generated Input – 2 char ANSCII	Not applicable
Control code recog.	Limited	Limited	None
Buffer size	12 bits	2 – 12 bits	8 bits
Interrupts	Message	CPU SCAN	?
Auto dial	No	No	?
Auto answer	No	No	?
Parallel data input	No	No	?
Voice answer back	No	No	?
Polling	No	No	?
Error check	BCC	Overrun, parity	?
NOTES:	Adapters are in-cluded. For computer to computer transmission over telpak	Adapters are in-cluded	For TTY Mod 33, 35 only. Adapters are included.

14

MANUFACTURER: CONTROL DATA CORP

MACHINE SERIES: CDC 6000

SYSTEM CHARACTERISTIC	MODEL 6200	MODEL 6400	XDS SIGMA 9
● TIME CRITICAL REAL-TIME			
Priority Interrupts	NA	NA	
Levels			224
Branch Address	↓	↓	PSD SWAP
Arm/Disarm			Yes
Enable/Disable			Yes
Triggerable			Yes
Clocks	4.096 MS	⟶	2-STD, 2-OPT (500HZ and 50/60, 500, 2K, or 8K HZ, or Other - Switchable)
Context Switch			
Exec. Time to Change Control	?	5 µs	XPSD 5.43 µs
Mult. Reg. Blks. or Load/Store Mult.	Yes-all	Yes-all	Mult. Reg. Blks.
Instrumentation	NA	⟶	Extensive List of Standard Products in Addition to Separate Systems Division Product Line.

SOFTWARE

- INTRODUCTION

CDC's primary operating systems for the 6000 Series are SCOPE and KRONOS. The operating philosophy of the 6000 Series is to assign the first peripheral processor (PP) the monitor job of keeping track of what programs are active and causing the CP (central processor) to switch amongst these programs. Additionally, another PP is assigned to operator communication and driving the scope displays.

- SCOPE

SCOPE is a multiprogrammed batch/remote batch system with high and low speed communication support. Seven jobs can be in the execution process simultaneously. Usually, one of the jobs is used to control the input/output of job control streams. CPU time is allocated one second at a time in a round robin fashion. Jobs are selected by priority and if core is not available they wait, otherwise the core is compressed to make room.

- KRONOS

This is a recent addition and should not be confused with KRONOS III for the CDC 3000 Series which is a much more respectable system for time-sharing. KRONOS for the 6000 Series is much like SCOPE except it can have a variable number of jobs in process of execution (not just 7) and it is advertized to handle up to 214 Remote Batch terminals and 384 communication lines (multi-drop, if required). Time-sharing capability consists of BASIC and FORTRAN with an EDITOR. This appears to be more "batch oriented" time-sharing than interaction and conversational because a program must be fully entered before diagnostics and/or errors are provided.

● CDC 6200/6400 MARKETING STRATEGIES – SCOPE/KRONOS

These systems will be discussed together because of the limited information on
KRONOS.

— Multi-Hundred Active Terminal System — KRONOS: The accent here is on data
centers and large corporate users. KRONOS will handle up to 384 active time-
sharing terminals or up to 16 remote batch processing terminals and a smaller
number (exact number not known – superset 256) of time-sharing terminals.

— Scientific Processing: This is usually an implied reference to the power of the
6600 rather than that of the 6200/6400. CDC has traditionally been in the Sci-
entific market and consequently has oriented their language processors and appli-
cations programs in that direction.

— Time-Sharing: CDC is emphasizing this added capability to lengthen the life
of the 6000 series.

● KRONOS/SCOPE STRONG POINTS

— Batch/Remote Batch Multiprogramming: KRONOS handles a variable number of
jobs (up to 32 maximum) while SCOPE is limited to 7. Two or possibly more
of the jobs are handling system functions such as operator communication and
symbionts.

— Scientific Processing: This is primarily a function of the hardware and not the
software because of the 60-bit word. Standard single precision on the CDC 6000
series is almost twice the single precision on Sigma 9.

● KRONOS/SCOPE WEAK POINTS

— Time-Sharing: SCOPE offers no time-sharing capability. KRONOS is advertized
to offer 384 interactive terminals. Admittedly, the terminals may be interactive,
but a suspicion surrounds their "conversational" ability. The suspicion is sup-
ported by the following:

- CDC 6000 series KRONOS never uses the term "conversational" — the
KRONOS III for 3000 series explicitly states the conversational ability.

- The KRONOS 6000 literature implies a batch-like compiling. For example,
they state that after the program is entered and a RUN command is given,
the user receives diagnostics — these could be compiler as well as run-time
diagnostics.

- The literature emphasizes the following:

"Compile and save binary code"

"Compile from source and execute immediately"

"Standard product compiler — for compatibility purposes"

Attempts have been made to verify this suspicion with no success at this time.

Additionally, KRONOS appears to have no terminal batch entry. This comes from the way CDC refers to the "interactive" ALGOL compiler. They say that the ALGOL program must be submitted through a remote batch terminal on the local batch stream for compilation. Once this is done, the interactive terminal user can call in the ALGOL program and execute it interactively. This is a severe restriction on the interactive terminal user.

– Context Switching Time: As previously stated, the 6000 series can actually switch from one program to another in a very short time – 5 microseconds. The point of contention is that given the philosophy of their operating system and the fact that the central processor is un-interruptible, except by the peripheral processor, then the peripheral processor which acts as the monitor (MTR) must actually cause the CP (central processor) to switch from one program to the next. This is done by the MTR checking a particular core location periodically (the time of this cycle is in question). Under these circumstances, each active program must be checked and each PP must be checked because MTR is the controlling element. Our reports indicate that the MTR cycle time for checking the needs of the programs and the PP's is something like 200 microseconds, half of which would be the average. Thus the MTR does not know if the CP is waiting for something like one hundred microseconds. Then, depending on requirements (priority) it has to decide which program gets the CP. Compound this with time-sharing requirements, remote batch requirements, symbiont requirements and, potentially, the system is spending a large amount of time in context switching. Due to the architecture of the 6000 series, it is difficult to see how this could be overcome.

– Commercial D.P.: Primarily due to lack of decimal instructions, etc., the CDC 6000 series is deficient in commercial capability. Due to the pure power of the CPU – and a number of years to improve on software – compilation and execution are respectable. Much checking and verification must be done due to the lack of memory parity checking. This is a reminder of 2nd generation equipment.

- XDS STRATEGIES — GENERAL

  - Time-Sharing: XDS has a listing of proven time-sharing capability. CDC 6000 series is a recent and unproven entry into the time-sharing area.

  - Multi-Use: XDS offers batch/remote batch with time-sharing and real time. Real time is an unknown quantity in the CDC 6000 series and the facilities offered under time-sharing are very minimal.

- SIGMA 9 STRENGTHS — UTS

  Outstanding amongst UTS strengths are:

  - Mapping: The KRONOS and SCOPE assigns programs to contiguous core and then must shuffle them around to make room for programs. UTS provides mapping and, thus, a greater core utilization.

  - Reentrant Processors: SCOPE does not use reentrant or shared processors. KRONOS probably uses reentrant processors for time-sharing. However, without mapping and/or a virtual memory it is doubtful that reentrant processors are very effective.

- OTHER SIGMA 9 STRENGTHS — UTS

  The following UTS features are outstanding advantages over KRONOS:

  - Performance Management: This is extremely limited or non-existent in KRONOS.
  - Automatic connection to processors
  - Automatic connection to charge structure
  - Real time
  - Operator control parameters

- SIGMA 9 STRENGTHS — XOS

  - XOS vs SCOPE: These two systems are matched relative to batch — SCOPE handles 7 programs with one being some symbiont activity; XOS handles 6 with symbionts built into the monitor. XOS additionally supports real time. XDS advantages lie in the areas of memory management and scheduling. SCOPE has fixed partitions for contiguous assignment of core. XOS provides mapping of core and dynamic partition sizes dependent on installation parameters and individual job requirements. In XOS, scheduling is by external priorities combined with class or type of job, resources required, etc. CPU allocation in SCOPE is accomplished by giving each program a fixed quantum of time in a round robin fashion. Scheduling in SCOPE is by priority — no consideration is given to other system resources; i.e., magnetic tapes, etc.

  - XOS vs KRONOS: KRONOS does a better job of memory management than does SCOPE by providing variable size partitions which can be under control of the monitor, compressed to make room for another program. This compression is performed by the PPU independent of actual processing done by the CPU. Also,

KRONOS can have up to 32 jobs active. Again, mapping in XOS is a much more efficient ~~manner to use~~ for memory management. Scheduling for KRONOS appears very similar to SCOPE in that it is by job priority and does not incorporate the installation's overall requirements as does XOS.

TIME-SHARING SOFTWARE COMPARISON

	UTS	CDC 6000 KRONOS
Operational Modes —		
. Dedicated time-sharing (max no. of terminals)	Yes (128)	Yes (384)
. Time-sharing/Batch	Yes	Yes
. Time-sharing/Batch/Remote batch	Yes	Yes
. Real time (combined with other modes)	Yes	No
Installation Control Features —		
. System control		
— Performance Management*	Extensive	Very limited
. User control		
— Accounting per user*	Yes	Yes
— Central account authorization*	Yes	Yes
— Processor authorization by account*	Yes	No
— T/S Terminal control		
Passwords	Yes	Yes
Automatic connection to processors* (Restricted use)	Yes	No
Automatic connection to charge structure* (Unique charges based on usage)	Yes	No
Shared processors (Reentrant)*	Yes	No
— Batch control		
Controlled resource usage (core, priority, Devices, etc.)*	Yes	No
Batch time quantum control	Yes	Yes
Percentage of CPU for batch	Yes	No
— Remote batch control	Same as batch	Similar to batch
— Real time control	Yes	None
. Operator Control		
— Core size for terminal users	Yes	No
— Number of terminal users	Yes	Yes
— File space	Yes	All available
— Number of tapes/user	Yes	No
— Percentage of CPU for batch	Yes	No
— Terminal time quantum	Yes	Unknown
— Compute bound quantum	Yes	Unknown
— Batch priority relation to terminal	Yes	Unknown
Program/Application Development Feature —		
. T/S terminals		
— Conversational processors (line by line interactive)	Basic Extended XDS FORTRAN IV EDIT	None Except Edit

	UTS	CDC 6000 KRONOS
- Interactive languages (non-conversational compile and/or execute only via terminal)	META-SYMBOL	FORTRAN, Basic ALGOL-Exec only
- Media conversion	PCL	Under EDIT
- TJE to batch stream	Yes	No
- Processor compatibility with batch	Yes	Yes
- File interchangeability	Yes	Yes
- Terminals can be logged on for particular processor	Yes	No
- Sharable processors*	Yes	No
- Debug facilities (conversational)	FDP, DELTA	Very limited
. Batch/Remote Batch		
- Language processors		
Standard FORTRAN, COBOL	Yes	Yes
Others		ALGOL, Basic
- Other processors	SL-1, FMPS, DMS, FLAG, GPDS	PERT/time, APT, Optima (L.P.)
. Real time		
- Languages	FORTRAN, MS	None
- Other processors	SL-1	
- Terminal program development	Yes	
Production Features —		
. T/S terminal		
- Terminal job insertion into batch (under priority control)	Yes	No
- Compiling not required for each run (load modules maintained)	Yes	Yes
- Output optionally permanent	Yes	Yes
- Terminal must stay connected	Yes (T/S) No - Batch	Yes (T/S)
. Batch/Remote Batch		
- Priority scheduling	Yes	Yes
- Standard system files available	Yes	Yes
- Non-standard files available	Yes	Unknown
- Printer forms control	Limited	Yes
- Checkpoint/restart	No	Yes
. Real time		None
- Initiated by T/S terminal	Optional	
- Initiated by batch/remote batch	Optional	
- Initiated by interrupt	Optional	

	UTS	CDC 6000 KRONOS
– Initiated by operator	Optional	
– Preempts core	Possible	
– Highest priority	Yes	

---

* UTS Reference Manual covers these topics in detail. Additional information is contained in UTS Specification No. 702489.

Manufacturer: CDC
Operating System: KRONOS
Computer System: CDC 6200/6400
Operating System Classification: Batch/Remote Batch Multiprogramming and Time-Sharing

	XOS	KRONOS
Operational Modes —		
. Batch/Remote Batch	Yes	Yes
. Dedicated Time-Sharing (max no. of terminals)	Yes (variable)	Yes (384)
. Batch/Remote Batch/Time-Sharing	Yes	Yes
. Real Time	Yes	N/A
Batch/Remote Batch Multiprogramming —		
. Minimum System Requirements	48KW	64KW
. Number of Concurrent Programs	7 (R.T. included)	Variable
. Remote Batch is Part of Monitor	Optional	No (takes program space)
. Symbionts Are Part of Monitor	Yes	No (takes program space)
. Scheduling Criteria	Priority, Avail Resources	Priority, Core
. Checkpoint/Restart	Yes	Yes
. Disk Pack Support	Yes	?(Suspect)
. Memory Management		
- System	Mapped	Dedicated
- User	Mapped	Contiguous (Compressable)
. Processor Support	FLAG, FORTRAN, COBOL, SORT/ MERGE	FORTRAN, COBOL, ALGOL
. Methods of Job Entry	CR, Tape, Disk, RBT	CR, Tape, Disk, RBT
. CPU Time Allocation	Priority	Priority
Communications Support —	Character and Msg oriented	Unknown
Time-Sharing —		
. Languages	All Standard	See T/S comparison (preceding pages)
. Performance Management	Yes	
Real Time Support —	Full	N/A
Applications —	DMS, GPDS, SL-1, FMPS	See T/S comparison (preceding pages)

Manufacturer: CDC
Operating System: SCOPE
Computer System: CDC 6200/6400
Operating System Classification: Batch/Remote Batch Multiprogramming

	XOS	SCOPE
Operational Modes —		
. Batch/Remote Batch	Yes	Yes
. Dedicated Time-Sharing (max no. of terminals)	Yes (variable)	No
. Batch/Remote Batch/Time-Sharing	Yes	No
. Real Time	Yes	No
Batch/Remote Batch Multiprogramming —		
. Minimum System Requirements	48KW	32KW
. Number of Concurrent Programs	7 (R.T. included)	7 (1 is Symbiont)
. Remote Batch is Part of Monitor	Optional	No (acts as program)
. Symbionts Are Part of Monitor	Yes	Somewhat (takes one program location)
. Scheduling Criteria	Priority, Avail Resources	Round Robin
. Checkpoint/Restart	Yes	Yes
. Disk Pack Support	Yes	?(Suspect)
. Memory Management		
- System	Mapped	Dedicated
- User	Mapped	Fixed Partition
. Processor Support	FLAG, FORTRAN, COBOL, SORT/MERGE	FORTRAN, COBOL, ALGOL
. Methods of Job Entry	CR, Tape, Disk, RBT	CR, RBT, Tape
. CPU Time Allocation	Priority	Time Slice
Communications Support —	Character and Msg oriented	Same
Time-Sharing —		
. Languages	All Standard	N/A
. Performance Management	Yes	N/A
Real Time Support —	Full	N/A
Applications —	DMS, GPDS, SL-1, FMPS	APT, PERT/Time Math Pack, Stat Pack

SYSTEM PRICES AND POLICIES

- **PRICES**

  As can be seen from the processor and memory price comparison graphs and the peripherals comparison charts on the following pages, the CDC 6200 and 6400 are considerably more expensive than a comparably configured Sigma 9. Because of the age of the CDC equipment you can expect CDC to alter the pricing downward in the coming months. Significant discounts have already been experienced in opportunities where CDC has bid the CDC 6400.
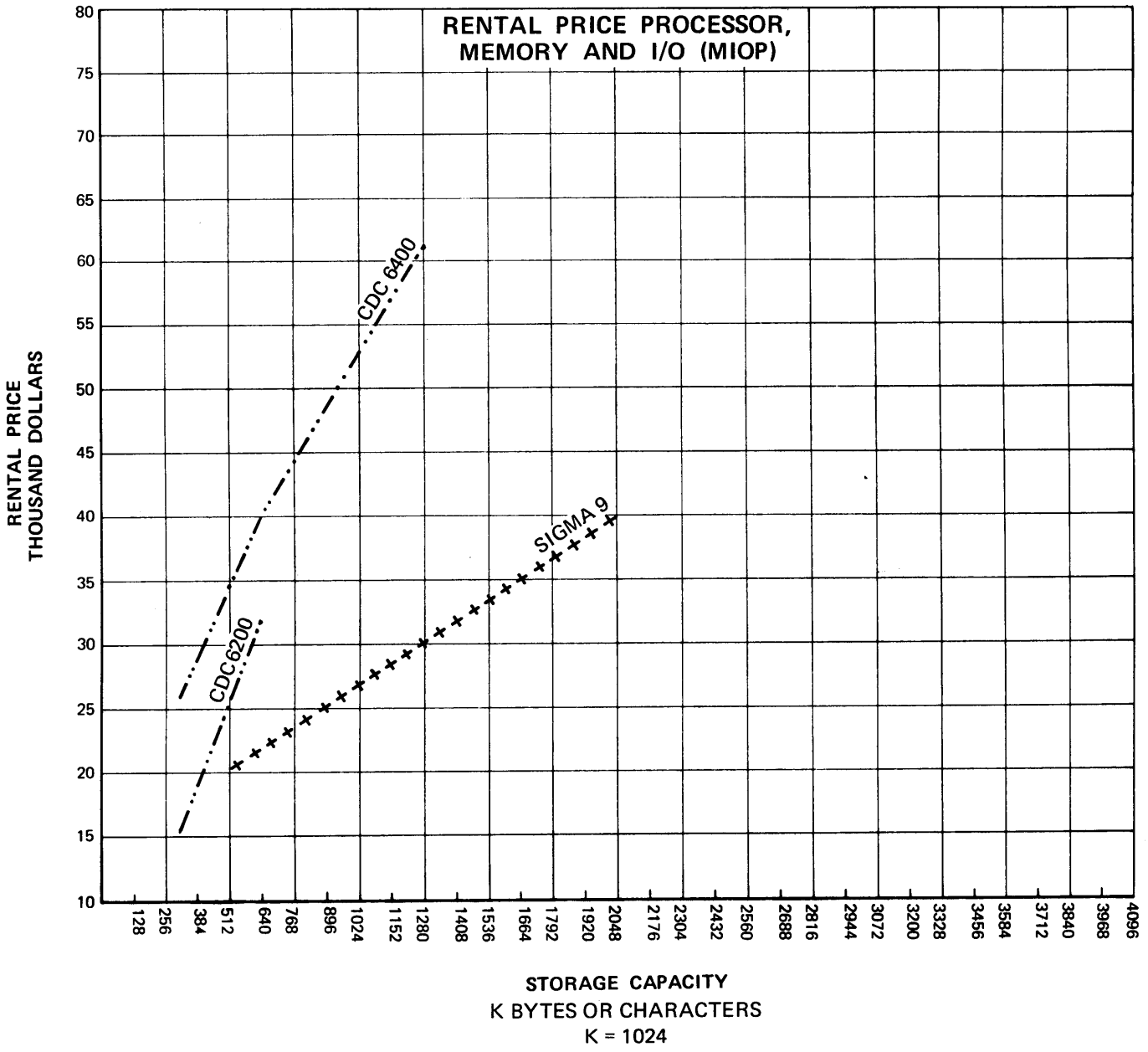
- **POLICIES**

  - CDC's basic lease contract provides for unlimited use of equipment.

  - CDC excludes maintenance from basic monthly rental charges to get total monthly rental, maintenance charges must be added to basic rental price. Throughout this study, all quoted prices include maintenance.

  - Long-Term Lease Plans: Quoted discounts apply only to basic rental price, exclusive of maintenance.

Term	Discount
3-year	5%
4-year	7%
5-year	10%

  - Commercial Credit Long Term Lease (5 - 8 Years): Through Commercial Credit, a subsidiary of CDC, more favorable lease rates are available. Details of these options are not known.

  - Educational Allowance: CDC does not have a standard educational discount. This is a negotiated rather than standard policy. Twenty percent is a good estimate of probable discount. They will also offer grants.

  - Purchase Options: CDC allows a purchase option credit of 68% of 1st year's rental and 45% of second year's rental.

  - Separate Pricing: CDC has gone to separate pricing of training, systems analysis support, and standard software systems and products released after January 1, 1970.

PURCHASE PRICE PROCESSOR, MEMORY AND I/O (MIOP)

PURCHASE PRICE
THOUSAND DOLLARS

5400
5200
5000
4800
4600
4400
4200
4000
3800
3600
3400
3200
3000
2800
2600
2400
2200
2000
1800
1600
1400
1200
1000
800
600
400
200

CDC 6400

SIGMA 9

CDC 6200

128 256 384 512 640 768 896 1024 1152 1280 1408 1536 1664 1792 1920 2048 2176 2304 2432 2560 2688 2816 2944 3072 3200 3328 3456 3584 3712 3840 3968 4096

STORAGE CAPACITY
K BYTES OR CHARACTERS
K = 1024

RENTAL PRICE PROCESSOR, MEMORY AND I/O (MIOP)

STORAGE CAPACITY
K BYTES OR CHARACTERS
K = 1024

Competitive Sales Guide

OCTOBER 1970   COMPETITIVE ANALYSIS   File: BUR
                                                         6500 Series

Contained in this Sales Guide is an assessment of the Burroughs 6503, 6504, and 6506 systems versus Sigma 9.

CONTENTS

Please refer to the Sigma 9 Overview (File: XDS) Sales Guide to compare the configuration diagrams of the Burroughs 6500 to Sigma 9, and also refer to the other Sales Guides on Sigma 9 Competitors for additional information.

Distribution: Sigma 9 Announcement Distribution and Competitive Information Reference Libraries

Kent L. Cootes, Manager
Competitive Analysis
XDS   AFC

Xerox Data Systems

SYSTEM SUMMARY

● INTRODUCTION

The B6500, a family of three computers with differing processor clock times and memory cycle time, was a logical upgrade and enhancement of the earlier and highly successful B5500. It retains the unconventional design of the B5500, Polish notation operand stacks or segments rather than instruction sequences. These machines were primarily designed with compilers in mind (that is, from a software requirement viewpoint). The key design objective of the B5500 was to eliminate the necessity for assembly language programming by providing a hardware structure designed for fast and efficient ALGOL and COBOL compilations (including FORTRAN on the B6500). It was also designed for efficient operating system control and as a result the MCP operating system probably has the lowest overhead factor of any medium to large scale multiprogrammed batch operating system in existence.

The B6500 is not machine language compatible with the earlier B5500, but is source language compatible. The other members of the so called B500 family – the B2500/3500/4500 – are entirely different machines from a design standpoint. Upgrading to a B6500 from one of these will require extensive conversion effort.

The B6500 was first announced in May, 1966. In July, 1967 Burroughs announced the B7500 series and revised the specifications for the B6500 series. The B7500 was dropped sometime in 1969 and the B6500 was revised again (thin film memories originally announced were replaced by core memory with the same cycle time). First delivery of a B6500 was finally made in June, 1969.

Essential differences between the three B6500 systems are:

Model	Memory Size	I/O Channels	CPU Clock Rate (Megacycles)	Memory Cycle Time
6503	96 – 768K Bytes	1 MPXer	2.5	1.2
6504	96K – 3.1M Bytes	2 MPXer	5.0	1.2
6506	96K – 3.1M Bytes	2 MPXer	5.0	0.6

● B6500 STRENGTHS

- Proven multiprogramming
- Compatible software (10 years B5500 experience)
- Floating channels greater flexibility (general I/O capacity/data rate flexibility)
- Inexpensive peripherals
- Fixed head/track (large data base capacity)

- Channel controller device exchanges
  Flexibility
  Redundancy
  Availability multi-path
     4 x 16 & 2 x 10 tape controller
     4 x 20 Disk controller
- Hi-level machine language, ALGOL - ESPOL
- Mixed mode (fixed and floating) operands
- Reentrancy/stack concept
- Context switching/stack concept
- Wider range of peripherals (e.g., 9-track, 1600 bpi magnetic tapes)

● B6500 WEAKNESSES

- No hi-speed I/O capability (limited aggregate bandwidth of fastest device - 395KB)
- Expensive CPU and memory
- Expensive IOP (10 x Sigma 9 - $200K vs only $20K)
- Limited time-sharing (currently only remote batch entry)
- No removable disk packs!
- Segment mapping requires much programmer awareness of memory stress on small segments (maximum of 32 segments and small segments philosophy forces writing of small programs)
- Maximum of 40 controllers/6504 and 6506, 20/6503. ·Every card reader, card punch, printer and console requires a separate controller
- Scientific computation is shown on the 6503 and 6504 (poor price/performer)
- Only has one set of two registers
- No real time interrupt capability (1-level part of I/O structure)
- No arm/disarm of interrupts
- No Power Fail-Safe
- No remote diagnostic capability

HARDWARE

Some of the highlights of the B6500 hardware features include:

● MEMORY

- 1.2 μsec cycle time for the B6503 and B6504
  .6 μsec time for the B6506

- Memory Range: 96 - 768K bytes for the B6503
                 96K - 3.1M bytes for the B6504 and B6506

- Word Size: 48 bits plus 3 control bits plus parity

- 4, 6, or 8 bit character and word addressing

- Two ports to memory for the B6503
  Two to four ports to memory for the B6504/6506

- Two way interleaving

- Memory map consists of 32 registers which address variable size segments. Limits of a segment are defined by a Mark Stack control word.

- Additional memory protection is provided via a write protect bit in each word.

- CENTRAL PROCESSOR

  - Polish notation string operator stacks rather than conventional instruction sequences

  - One 48-bit instruction word contains 6 (8-bit) syllables. An instruction consists of from 1 to 12 syllables depending on the type of operation. Average is 2 to 3 bytes. Syllables are executed sequentially from left to right.

  - Word-oriented operators but decimal and byte string operators are available.

  - Floating point can be decimal or binary (in any mix).

  - Multi-level indirect addressing via data descriptors.

  - Dual processor-oriented system is fully software supported. Either processor can handle any I/O interrupt occurring on either multiplexor regardless of which processor initiated the I/O.

  - Many operating system functions are implemented in hardware or assisted by unique hardware features.

  - 2.5 Megacycle clock rate for the B6503
    5.0 Megacycle clock rate for the B6504 and B6506

- INPUT/OUTPUT

  - 1 Multiplexor channel is available on the B6503
    1 or 2 Multiplexor channels on the B6504 and B6506

  - Each multiplexor is capable of processing up to ten simultaneous I/O operations

  - Each multiplexor can have 4 to 10 data switching channels. The channels float, assigned by the multiplexor to specific peripherals as required. Up to 20 peripheral controllers are serviced by these channels.

  - Additionally, each multiplexor can have from 1 to 4 data communication processors with their own channels (exclusive of the data switching channels).

  - Also, one real time adapter can be attached to each multiplexor.

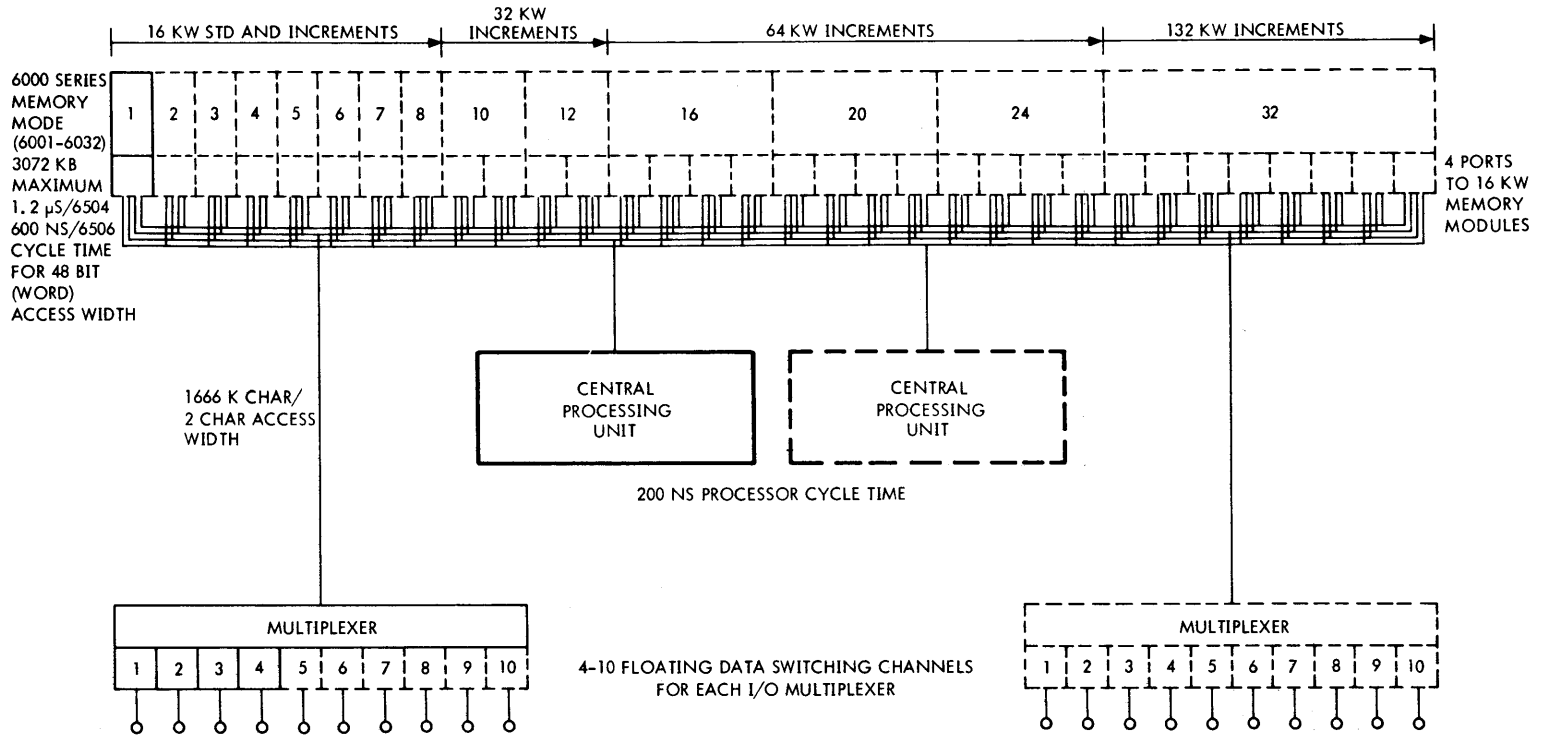 - Maximum aggregate transfer rate per multiplexor is 1.667MB.

16 KW STD AND INCREMENTS →   32 KW INCREMENTS →

6000 SERIES
MEMORY
MODES
(6001-6012)
768 KB MAX
1.2 μS CYCLE
TIME FOR
48-BIT (WORD)
ACCESS WIDTH

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 12 |

1666 K CHAR/
2 CHAR ACCESS
WIDTH

CENTRAL
PROCESSING
UNIT

400 NS PROCESSOR
CYCLE TIME

MULTIPLEXER

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

4-10 FLOATING DATA
SWITCHING CHANNELS
FOR EACH I/O
MULTIPLEXER

NOTES :
    (1) MULTIPLEXER – NO COMMAND CHAINING
    (2) MULTIPLEXER – NO DATA CHAINING

BUR 6503

6

16 KW STD AND INCREMENTS | 32 KW INCREMENTS | 64 KW INCREMENTS | 132 KW INCREMENTS

6000 SERIES
MEMORY
MODE
(6001-6032)
3072 KB
MAXIMUM
1.2 μS/6504
600 NS/6506
CYCLE TIME
FOR 48 BIT
(WORD)
ACCESS WIDTH

1 2 3 4 5 6 7 8 10 12 16 20 24 32

4 PORTS
TO 16 KW
MEMORY
MODULES

1666 K CHAR/
2 CHAR ACCESS
WIDTH

CENTRAL
PROCESSING
UNIT

CENTRAL
PROCESSING
UNIT

200 NS PROCESSOR CYCLE TIME

MULTIPLEXER

1 2 3 4 5 6 7 8 9 10

4-10 FLOATING DATA SWITCHING CHANNELS
FOR EACH I/O MULTIPLEXER

MULTIPLEXER

1 2 3 4 5 6 7 8 9 10

NOTES :
   (1) MULTIPLEXER NO COMMAND CHAINING
   (2) MULTIPLEXER NO DATA CHAINING

BUR 6504/6506 DUAL PROCESSOR

MANUFACTURER: BURROUGHS

MACHINE SERIES: B6500

SYSTEM / CHARACTERISTIC	B6503	B6504	XDS SIGMA 9
● <u>MAIN MEMORY</u>			
Size Range	96-768K Bytes	96K-3.1M Bytes	512K - 2M Bytes
Cycle Time	1.2 us	→	850 NS
Access Width	1-48* Bit Word	→	4 Bytes
Interleave	2-Way CE:CNTRLD.	→	4 Way
Access Ports	2	4	2 - 12
Map	Var. Length Seg.	→	Std.
Protection	Yes	→	1st 512KB Only
Min. Increase	64KB	→	128KB
● <u>BUFFER MEMORY</u>	NA	NA	NA
Size Range			
Cycle Time			
Access Width	↓	↓	↓
Interleave			
Access Ports			
Protection			
● <u>LARGE CAPACITY STORAGE</u>	NA	NA	NA
Size Range			
Cycle Time			
Access Width			
Interleave			
Access Ports			
Xfer Rate			
Map			
Protection	↓	↓	↓
Exec. Instr. From			
● <u>CENTRAL PROCESSING UNIT</u>			
CPU Clock Cycle	400NS	200NS	Instr. & Seq. Dep.
<u>Look Ahead</u>			
Instr. Prefetch	1-6 Polish Str. Oprs.	→	12 Bytes Less Current Instr.
Levels of Decode	1-Current Level	→	One Beyond Current

DETAILED HARDWARE SPECIFICATION SHEET

MANUFACTURER: BURROUGHS

MACHINE SERIES: B6500

SYSTEM CHARACTERISTIC	B6503	B6504	XDS SIGMA 9
● INSTRUCTIONS			
Commercial KIPS	105.9	212.0	64.7
Scientific KIPS	379.8	567.2	763.4
Word Size	48*Bits - 6 Bytes	⟶	32 Bits or 4 Bytes
Instruction Size	1-12 Bytes, 2-3 Byte Ave.	⟶	2 or 4 Bytes
Mult. Addr. Ops.	1	⟶	
Decimal	Yes	⟶	Std.
Fixed Point	Yes	⟶	Std.
Floating Point	Floating/Dec & Bin	⟶	Sing. & Dbl. Std
Stor.-to-Stor.	Stacks Only	⟶	NA
Byte/Char. Manip.	Yes	⟶	Yes
Stack	Automatic	⟶	Yes
● ADDRESSING			
Indexing	Yes	⟶	Yes
Indirect	Yes	⟶	Yes
Virtual	Var. Size Seg.	⟶	Map - 512KB/Prog.
Page Size	Variable	⟶	512KB
No. of Pages	32/Seg. Regs.	⟶	
Scheme	Displacement/Seg.	⟶	Direct Reference
Size	Bit, Byte+	⟶	1, 2, 4 & 8 Byte
Max. Addr.	1MW or 6MB	⟶	4MW or 16MB/ Extended Mode
● REGISTERS			
Index	Gen. Purpose Reg.	⟶	General Purpose
Gen. Purpose	2	⟶	2 Blks. of 16 Std.
Mult. Gen. Reg. Blks.	No	⟶	2 Std. & Opt.
Floating Point	Sing. or Dbl. Std /Gen. Purp. Regs	⟶	Single & Double Std.
Decimal	Gen. Purp. Regs.	⟶	
● INTERVAL TIMER	512us-Max 1 Sec	⟶	500 HZ & one of (50/60, 500, 2K, or 8K HZ) Switchable

MANUFACTURER: BURROUGHS

MACHINE SERIES: B6500

SYSTEM CHARACTERISTIC	B6503	B6504	XDS SIGMA 9
⊙ INTEGRAL MULTIPLEXER	NA	NA	NA
Channels			
Subchans./Controllers			
Aggregate Xfer			
MPX – Max Chan Xfer			
Burst – Max Chan Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Multiplexing			
Burst			
Command Chain			
Data Chain			
Xfer in Channel	↓	↓	↓
⊙ EXTERNAL MULTIPLEXER	1 - I/O MPX	2 - I/O MPX	1 MIOP Std 11 Max
Channels	4 - 10 Floating+	⟶	1 Std. 2 Max.
Subchans./Controllers	20	2-Sets of 20 Sub-Chan	8-32
Aggregate Xfer	1.667MB	⟶	1.9MB W/4 Byte & 2 Chan.
MPX – Max Chan Xfer	?	?	470KB
Burst – Max Chan Xfer	?	?	
Command Chaining	No	⟶	Yes
Data Chaining	No	⟶	Yes
Data Path Width	Byte	⟶	1 or 4 Byte/Dev & 4 Byte to Mem.
⊙ INTEGRAL SELECTOR	NA	NA	NA
Channels			
Subchan./Controllers			
Max. Xfer			
Command Chaining			
Data Chaining			
Data Path Width			
CPU Loading			
Xfer.			
Command Chain.			
Data Chain.	↓	↓	↓

DETAILED HARDWARE SPECIFICATION SHEET

MANUFACTURER: BURROUGHS

MACHINE SERIES: B6500

SYSTEM CHARACTERISTIC	B6503	B6504	XDS SIGMA 9
● EXTERNAL SELECTOR	NA	NA	1 - 4 RIOP's
Channels			1
Subchans./Controllers			1 Integral
Max. Xfer.			3MB
Command Chaining			Yes
Data Chaining	↓	↓	Yes
Data Path Width			4 Byte to Mem.
● SYSTEM BANDWIDTH	?	?	25.3MB
● AVAILABILITY			
Watchdog Timer	2 Sec Intrpt. Timer	⟶	Yes
Power Fail Safe	NA	⟶	Saves Volatile Regs. & Brings to Orderly Halt
Recoverability	Diag. Test -No Auto	⟶	Yes
Relocatible Homespace	NA	⟶	Yes
Instruction Failure	NA	NA	
Pre. Exec. Test			Yes
Isolation			Yes
Retry			
CPU			Yes
Channel	↓	↓	Yes

● PERIPHERALS

Burroughs peripherals are expensive compared to Sigma. For example, their 1400 cpm card reader costs $25,800 versus $24,000 for our 1500 cpm reader. This is unusual since peripherals for the B2500/3500/4500 are priced very competitively and they are the same devices (only the controllers differ). This may reflect a Burrough's policy that, where big systems are concerned, peripheral prices are not as critical. Note also that Burroughs has no low speed card readers or punches for the B6500.

PERIPHERALS - W/PRICE FOR COMPARISON

MANUFACTURER   Burroughs

MACHINE SERIES   B6500

DEVICE		MODEL NO.	SPEED	PURCHASE	MAINT.	1 YR. LEASE
Card Reader		B9111	800 cpm	$ 20,450	$ 99	$ 425
		B9112	1400 cpm	25,800	142	550
Card Punch		B9213	300 cpm	29,640	150	630
Printer	Drum	B9242	860 lpm	62,000	216	1,150
	Drum	B9243	1100 lpm	67,500	236	1,255
	Chain	B9245	315 lpm	33,120	161	690
Magnetic Tape	7 track	B9380	36 KC	79,200	275	1,700
		B9391	72 KC	131,400	675	2,800
		B9394-1	96 KC	147,900	695	3,150
	9 track 800 BPI	B9381	36 KB	79,200	275	1,700
		B9392	72 KB	131,400	675	2,800
		B9394-2	96 KB	147,900	695	3,150
	9 track 1600 BPI	B9382	72 KB	100,500	305	2,150
		B9393-1	144 KB	124,800	600	2,600
		B9393-2	192 KB	148,800	640	3,100
		B9393-3	240 KB	182,400	680	3,800

NOTE: All prices include controller price where required.
            Magnetic tape prices are prices of controller and 4 drives.
            Printer prices include price of buffer where available.

● MASS RANDOM ACCESS STORAGE

Burroughs offers only fixed (non-removable) head per track devices and their chief sales strategy is to stress heavily the inherent reliability and higher throughput potential of these devices. They offer three types with differing capacities, access times and transfer rates. Lease prices below include control and electronics.

Burrough's prices are highly price competitive with our entire Sigma random access device product line. Burrough's prices are consistently lower for a given device capacity. Burroughs does not offer removable pack devices. The best strategy to use with regard to Random Access Device capability is to emphasize the advantages of removable packs. Also, our 7232 does have a higher transfer rate (384KB versus 218KB for their 10 megabyte unit). Note well that Burroughs does not have a random access device comparable to the XDS 7212!

RANDOM ACCESS DEVICES

MANUFACTURER Burroughs

MACHINE SERIES B6500

MODEL NO.	CAPACITY PER MODULE	KB RATE	AVE. ACCESS	CAPACITY	PURCHASE	MAINT.	1 YR. LEASE
SYSTEM STORAGE AND SWAPPING RADS							
B9372-11	10MB	218KB	20 Msec	10MB	$ 92,000	$ 215	$ 1,900
				20MB	133,800	330	2,750
				40MB	209,200	560	4,450
				50MB	244,300	675	5,300
FILE STORAGE (NON REMOVABLE)							
B9375-10	100MB	377KB	23 Msec	100MB	254,400	615	5,350
				200MB	505,440	1227	10,580
				400MB	984,960	2423	20,570
				800MB	1,939,680	4809	40,460
				1000MB	2,419,200 ·	6005	50,450
B9375-12	100MB	216KB	40 Msec	100MB	241,000	515	4,350
				200MB	480,080	1029	8,610
				400MB	931,360	2023	16,570
				800MB	1,833,920	4011	31,302
				1000MB	2,285,200	5005	40,450
B9375-13	100MB	395KB	60 Msec	100MB	146,400	415	3,100
				200MB	290,880	829	8,810
				400MB	552,960	1623	11,570
				800MB	1,077,120	3211	22,490
				1000MB	1,339,200	4005	27,950

● COMMUNICATIONS

Burroughs has only one communications processor available for the B6500. They have no multiplexors and no single line control. The communications processor is extremely flexible, programmable, able to handle almost any combination of terminals – similar to our systems CIOP. However, it is extremely expensive for simple networks consisting of one or two types of terminals or for small networks. For example, for a teletype network the DCP rents for $900 plus $200 per 16 lines plus $15 per line. Our 7611 character oriented multiplexor can handle 32 lines for approximately $900.

MANUFACTURER Burroughs

MACHINE SERIES B6500

SYSTEM / CHARACTERISTIC	DATA COMM. PROCESSOR (DCP)		
Function	Multi Line Contr.		
Programmable Contr.	Yes		
Memory	Main or local (OPT)		
Data Path Interface	MIOP		
Number of lines	16 – 256		
Max. bit rate/line	45 to 9600 bps		
Aggregate bit rate	?		
Rate sensing	No		
Mode(s)			
HDX,FDX,ECHPLX	SIMPLX, HDX		
ASYNC, SYNC	Both		
BISYNC	Yes		
Data Format	4, 5, 7, 8 level		
Code translation	DCP programmable		
SYNC pattern	Fixed per line		
Control Code Recog.	DCP programmable		
Buffer size	1 bit or 1 byte		
Interrupts	DCP programmable		
Auto dial	Yes		
Auto answer	Yes		
Parallel data input	Yes		
Voice answer back	Yes		
Polling	Yes		
Error check	DCP programmable		
NOTES:			

MANUFACTURER: BURROUGHS

MACHINE SERIES: B6500

SYSTEM CHARACTERISTIC	B6503	B6504	XDS SIGMA 9
● TIME CRITICAL REAL-TIME			
Priority Interrupts			
Levels	1	1 or 2	224
Branch Address	?	?	PSD SWAP
Arm/Disarm	No	→	Yes
Enable/Disable	Yes	→	Yes
Triggerable	NA	→	Yes
Clocks	2.4μs/R-T Clock	→	2-STD, 2-OPT (500HZ and 50/60, 500, 2K, or 8K HZ, or Other - Switchable)
Context Switch			
Exec. Time to Change Control	?	?	XPSD 5.43 μs
Mult. Reg. Blks. or Load/Store Mult.	Auto Push-Down/ Gen. Purp. Regs.	→	Mult. Reg. Blks.
Instrumentation	NA	→	Extensive List of Standard Products in Addition to Separate Systems Division Product Line.

- REAL TIME

    The B6500 can have one real time adapter per multiplexor. Thus, the B6503 can handle only one line. The B6504/6506 can handle two lines maximum. Needless to say, the B6500 is not designed for real time operation and software support of the real time adapter is doubtful.

SOFTWARE

- INTRODUCTION

    Burroughs has proven multiprogrammed batch capability in the Master Control Program (MCP). MCP was developed originally for the B5500, it has been revised and enhanced some for the B6500 but is, in general, the same operating system. It is an extremely efficient operating system with probably the lowest overhead factor of any multiprogramming system in the industry. The efficiency is the result of two factors: The hardware was designed around the operating system, implementing many functions in hardware that, in other systems, are exclusively software. Second, it is simple, does not have some of the more complex features of IBM's OS/MVT for example.

- MASTER CONTROL PROGRAM (MCP)

    Among the key elements of MCP are:

    - An executive routine which: (1) coordinates MCP operations by determining the type of control routine required and transferring control to the appropriate routine; (2) supervises I/O operations and the execution of all program segments in main memory; and (3) maintains a log of system operations.

    - A scheduling routine which evaluates the priorities and equipment requirements of a batch of jobs and maintains a schedule that ensures effective system utilization.

    - An environment control routine which dynamically allocates main memory and assigns I/O devices in accordance with each job's requirements.

    - An exception condition routine which provides standard procedures for dealing with errors.

    All programs including language processors are totally reentrant. The system features dynamic memory allocation and peripheral allocation, priority scheduling of a variable number of multiprogrammed jobs and multiprocessing (dual processor systems are totally supported).

    A Message Control System is available under MCP which provides remote batch and interactive text and file edit capability. Interactive compilers are not available. The system was designed primarily for transaction and message handling capability. Because of the ability to do text and file editing some time-sharing capability is present, however, it is extremely limited.

- PROGRAM PRODUCTS

  Extremely efficient ALGOL, COBOL, and FORTRAN compilers are available for
  the B6500. The COBOL is an ANSI 68 COBOL. FORTRAN is compatible with
  IBM 7094 FORTRAN IV, Version 13, with ANSI FORTRAN as a compatible subset.
  All compilers are extremely fast (FORTRAN compilation speed upwards of 10,000
  card images per minute, COBOL – 5000 per minute), and produce object code that
  compares favorably with assembly language programming. Burroughs claims that be-
  cause of the fast efficient compilers, a symbolic assembler is not required. All
  compilers produce reentrant code and are, themselves, reentrant.

  A full set of scientific and commercial application packages are available including
  a math package, simulation languages, inventory control systems, PERT, and an on-line
  data base management system.

- **B6500 MARKETING STRATEGIES — B6500 MCP**

  Expect the following items to be stressed by Burroughs:

  - Batch/Remote Batch Multiprogramming: MCP is aimed at the general purpose — both commercial and scientific — computer market.

  - MCP is a Proven Product: The history of MCP on the B5500 is emphasized and strongly related to MCP for B6500.

  - Software/Hardware Combination: The unique design of the B6500 closely ties software and hardware together. The B6500 represents a sharp departure from other computers in the industry and as such it has a certain attraction to the sophisticated user.

- **STRONGPOINTS — B6500 MCP**

  - Multiprogramming/Communications: Based on the history of the B5500 and limited reports on the B6500, this is a very strong feature. Compiling is fast, the code created is good and overhead is low. Programs are written in problem-oriented languages — very little assembly language is used. Communications software, for low and high speed transaction devices, is integrated into the monitor.

- **WEAKNESSES — B6500 MCP**

  MCP appears to create I/O for the sake of I/O much like a demand paging time-sharing system does. Program segments are dynamically brought into core as needed, overlaying segments that may have to be written out of disk.

  - MCP has no real time support
  - MCP has no time-sharing

- XDS STRATEGIES — GENERAL — MCP

  - <u>Time-Sharing/Real Time:</u>   Whenever possible emphasize our unique supporting T/S and R.T.

  - <u>Avoid Multiprogramming Benchmarks:</u>   There is good multiprogramming on the Sigma 9. This is only meant to advise caution because of the fast compilation reentrant compilers, reentrant compiled code – particularly in the area of many very small jobs.

- SIGMA 9 STRENGTHS — UTS

  Burroughs MCP has no capability to compare to the UTS time-sharing capability. All of the many outstanding features of UTS should be used when facing MCP competition.

- SIGMA 9 STRENGTHS — XOS

  - Stress the time-sharing and real time capability of XOS. MCP provides neither of these capabilities

  - When combined with time-sharing and/or real time, the multiprogramming capability adds to the total XOS ability to meet the customer's needs.

Manufacturer: Burroughs
Operating System: MCP
Computer System: B6500
Operating System Classification: General Purpose

	XOS	MCP
Operational Modes —		
. Batch/Remote Batch	Yes	Yes
. Dedicated Time-Sharing (max no. of terminals)	Yes (variable)	No
. Batch/Remote Batch/Time-Sharing	Yes	No
. Real Time	Yes	No
Batch/Remote Batch Multiprogramming —		
. Minimum System Requirements	48KW	?
. Number of Concurrent Programs	7 (R.T. included)	Variable
. Remote Batch is Part of Monitor	Optional	Yes
. Symbionts Are Part of Monitor	Yes	Yes
. Scheduling Criteria	Priority, Avail Resources	Priority, Core
. Checkpoint/Restart	Yes	Yes
. Disk Pack Support	Yes	No
. Memory Management		
- System	Mapped	Dedicated
- User	Mapped	Dynamic (much like MAP with variable page sizes)
. Processor Support	FLAG, FORTRAN, COBOL, SORT/ MERGE	COBOL, FORTRAN, PL/I, ALGOL (All reentrant)
. Methods of Job Entry	CR, Tape, Disk, RBT	CR, Tape, Disk, RBT
. CPU Time Allocation	Priority	Time Slice, Priority
Communications Support —	Character and Msg oriented	Transaction
Time-Sharing —		
. Languages	All Standard	N/A
. Performance Management	Yes	Unknown
Real Time Support —	Full	N/A
Applications —	DMS, GPDS, SL-1, FMPS	Data mgt., Inventory, Banking

SYSTEM PRICES AND POLICIES

- **PRICING**

    Burrough's system prices are higher than Sigma 9 for processor, memory, peripherals, and data communications. Random Access devices are cheaper particularly for file storage devices. However, a given B6500 configuration should cost more than a comparable Sigma 9. Expect Burroughs to reduce prices in the near future, particularly on processor and memory.

- **POLICIES**

    - Burrough's basic lease contract provides for unlimited use of equipment. This is an assumption on our part based on the GSA Schedule and partially verified by Burroughs.

    - Burroughs includes maintenance in basic rental charges.

    - Burroughs has two long-term lease plans:

3-year	5% discount
5-year	10% discount

    - Purchase Option: Burroughs allows a purchase option to the Federal Government (GSA Schedule) of 75% of the first 6 months rental plus 50% of the 7th through the 24th month's rental. We have no information on purchase option policy to non-government customers; however, we can assume that it will be lower than that given to the Federal Government (probably 60% of the first year's rental).

    - Burroughs has not gone to separate pricing of software products or analyst support on standard equipment.

PURCHASE PRICE PROCESSOR, MEMORY AND I/O (MIOP)

STORAGE CAPACITY
K BYTES
K = 1024

RENTAL PRICE PROCESSOR, MEMORY AND I/O (MIOP)

STORAGE CAPACITY
K BYTES OR CHARACTERS
K = 1024
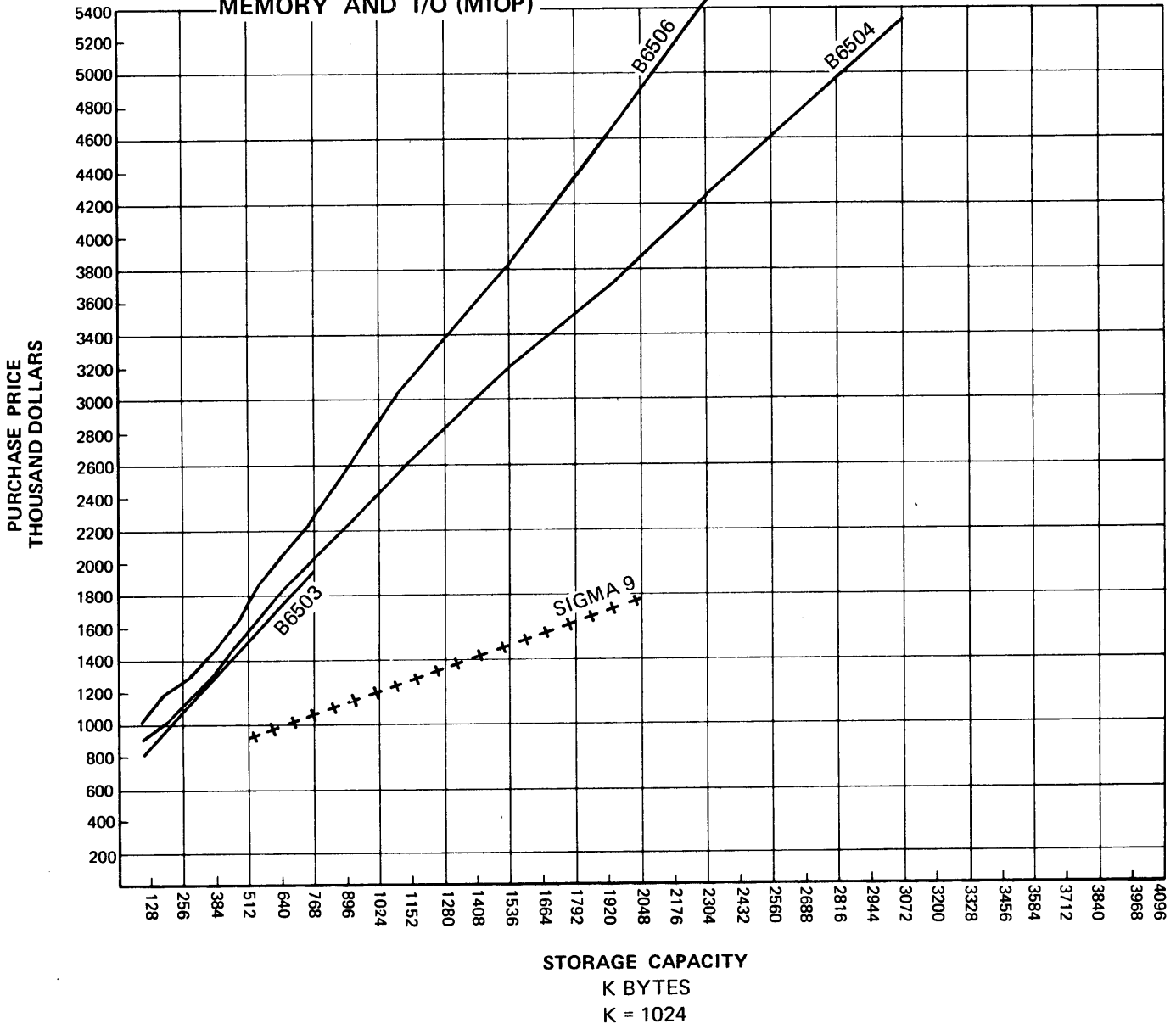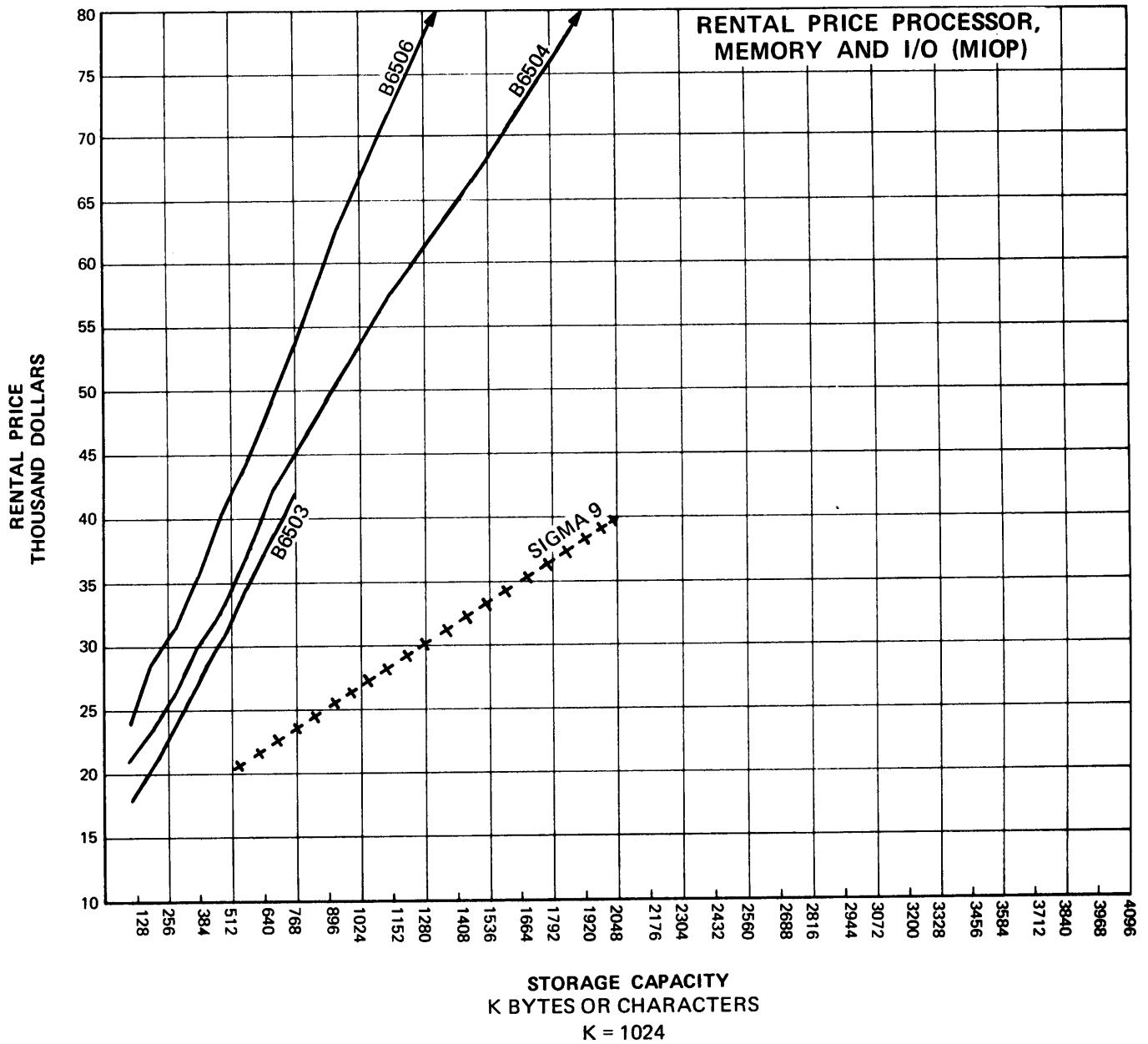
Competitive Sales Guide

OCTOBER 1970   COMPETITIVE ANALYSIS   File: XDS

                                                                               DMS

This Sales Guide presents an overview of XDS DMS versus the competition — primarily IBM's IMS/DL-1 system. We think you will find it to be especially valuable.

CONTENTS

Competitive Analysis extends its appreciation to Wylye Robertson, Applications Support, for preparing this report.

Xerox Data Systems

INTRODUCTION

- There are few data base management systems available from other manufacturers which contain capabilities close to those of DMS. One exception is IBM's Data Language/I (DL/I) which operates as an integral subsystem under the IBM information Management System, Version 2 (IMS, Version 2). Since DL/I's capabilities begin to approach DMS's, and since IBM is expected to provide the most potent competition to the Sigma 9, this report will be devoted almost exclusively to a comparison of DL/I vs DMS.

- DMS and IMS should not be directly compared. DMS is a very powerful data management system operating in the batch mode under BPM. IMS is a transaction oriented teleprocessing system serving remote key driven terminals, queueing messages, scheduling programs, etc., with its own control program running under OS/360 MVT or MFT-II. Incorporated within IMS is a data management facility called Data Language/I (DL/I), and it is this package that is compared here to DMS. DL/I is stored using (currently) three access methods: .
    - SAM (Sequential Access Method)
    - ISAM (Index Sequential Access Method)
    - OSAM (Overflow Sequential Access Method)

  A new direct access method is forthcoming in Version 2 of IMS, scheduled for availability March 1, 1971.

- Other competitive systems are treated very briefly on the following page. Thereafter all discussion and comparison will be centered on DMS vs IBM's DL/I.

- The reader is assumed to have some familiarity with data base management systems. It is important to note that IBM terminology differs significantly from XDS terminology in reference to data base structure names and access techniques. The reader should refer to the OVERVIEW section of this report for resolution of key terminology differences.

SUMMARY - COMPETITIVE SYSTEMS

- **IMS Version 2** -- OS/360 MVT, MFT-II -- Version 2 available March 1, 1971
  - Transaction oriented, message scheduling and processing
  - Customer written message processing and application programs tailored to each installation.
  - Uses DL/I for file management. Version 2 provides list structures, file inversion, etc.

- **UNIMS** -- Univac 1106/1108 under Exec 8 and Exec II
  - Fixed length records only
  - No data chaining
  - Package to assist conventional programmer in arranging files and retrieving reports
  - Not in DMS class at all!

- **IDS** -- GE-600 series (now Honeywell)
  - Similar to DMS in structure
  - No secondary indices(for file inversion)
  - No separate FDP for DDL -- Data Definition embedded in COBOL program
  - Runs under COBOL only (Advertised under FORTRAN, but no known users)

- **Disk FORTE** -- Burroughs 3500/4500
  - Handles chain file structure
  - Not in Sigma 9 class

- **MRP Manufacturing Records Processor** -- IBM OS/360
  - No information available to Competitive Analysis

- **Bill of Material Processor** -- IBM DOS
  - Not in Sigma 9 class
  - Will be covered in subsequent comparison

SUMMARY - DMS vs IBM DL/I

DL/I ADVANTAGES
- Can have multiple data bases. This is an apparent advantage due to IBM terminology.
- Allows a shared data base between two or more users
- Provides teleprocessing capabilities -- as part of full IMS facilities (Transaction oriented system)

DMS ADVANTAGES
- Superior system design to DL/I. IMS improvements are tacked-on additions to old hierarchic structure.
- Superior file re-organization methods -- dynamic and automatic -- deletions physically removed.
- Secondary indices (for file inversion) maintained automatically for any item if specified by parameter in DDL.

- Faster access methods -- Random access requires fewer seeks than ISAM, OSAM, etc.
- Sort keys are optional under DMS providing greater user flexibility. Every DL/I group requires a sort key.
- DMS has more commands for traversing chains, and other features which allow faster, easier access to data than DL/I.
- Better recovery procedures are available under DMS than under current DL/I.
- Password protection is provided all the way down to Item level under DMS -- DL/I provides protection only to segment (group) level.
- DMS handles floating point data. DL/I does not!
- DMS has more efficient space allocation methods.
- Under DMS, file maintenance is easier -- a single INSERT places a group on many chains.
- COBOL copy file is automatic under DMS whereas this requires manual action in DL/I.
- An optional check sum for each data page is provided in DMS but not in DL/I.
- Inventory pages for automatic space management are provided under DMS.
- DMS runs under FORTRAN (DL/I under COBOL, assembly language, and PL/I).

COMPARISON OF DMS vs DL/I

ADVANTAGES OF DL/I

- Multiple Data Base Processing: IBM's definition of a data base is a group of data base records (cf. Fig. 2 and 3). The same identical data base that would be defined in DMS as a single data base, could be defined in DL/I as many data bases. They are, in fact, separate OS/360 data sets each with its own set of Data Definition (DD) cards in the Job Control Language (JCL). These data bases (or data sets) may be a product of other, non-DL/I applications and may interface with other programs.

  In such a case the interfacing data base will be either a Sequential or Index Sequential data set, and will be processed in the batch mode in most cases. DMS is designed to eliminate the necessity for batch processing, while DL/I has compromised processing efficiency as a result of its adherence to the old batch processing concepts.

  DMS cannot process two different DMS-defined data bases concurrently by the same processing program. However, there is no restriction on permitting the application program to open other non-DMS files, such as Keyed or Sequential files, should such an interface be required.

- Shared Files: Several programs may concurrently process against the same DL/I data base under multiprogramming and/or time sharing. The only requirement is that no two programs may process the same Segment type at the same time. DMS does not, at the present, have this capability.

- Teleprocessing Capabilities: DL/I is designed as part of the IBM Information Management System (IMS), which is a complicated transaction oriented system. XDS currently has no comparable program product.

  DMS data base access can be accomplished from terminals, but until shared file protection exists, this will not be practical.

ADVANTAGES OF DMS
- General Remarks: On the surface, it is possible to paint a picture of DL/I that closely resembles DMS. Data relationships can be shown, using Version 2, that closely resemble many of the chains and nested chains characteristic of DMS data bases.

  The big difference is that DMS was designed originally with these data relationships in mind. DMS was designed with full knowledge of the slow access and expensive overhead that can result as a consequence of this approach to data management. For that reason optimizing techniques, and a wide variety of methods for accessing data were incorporated into DMS from the beginning as a part of its original design.

  DL/I, on the other hand, was originally designed as a hierarchical data structure on a sequential or index sequential data file. Hierarchical file structures are not new to data processing. Sequentially organized hierarchical files were utilized over twenty years ago in large card files, long before even magnetic tapes came into prominence. DL/I was merely an attempt to process these files utilizing the only tools available through IBM contributed software (namely, sequential and index sequential access methods - SAM and ISAM). ISAM was not capable of handling the data presented without first being modified to the Overflow Sequential Access Method (OSAM). Further extensions were compounded on this to come up with the advertised file structure of the forthcoming Version 2. The fact remains that efficient access methods to a DMS type data base were compromised in the original design. DL/I application programs will still view data with the same old hierarchical data relationships that they did in Version 1.

- File Re-organization: When a record is deleted from an ISAM file it is not physically removed from the storage device, but its key is written over with a series of binary ones. It continues to occupy the same space. A new record to be inserted is written at the place on a track where its key logically falls. Any data physically residing there is shifted toward the end of the track together with all subsequent data on that track. The last record may be shifted off the end of the track, and if so, it will be re-written on an overflow track. As a result of a number of such insertions and deletions the ISAM file becomes bulky, and disorganized. Records are retrieved more slowly due to additional accesses, and a good percentage of the space is occupied by deleted records. Performance gradually deteriorates, and eventually the file must be re-organized. This is accomplished by copying the

file to another device. It is sometimes time consuming, and it causes the file to be unavailable to regular users during the process.

The Hierarchical Direct Access Method (HDAM) organization under Version 2 is said to be able to re-use space occupied by deleted segments through the maintenance of free space addresses. This is a step in the right direction, but insertions can still cause the file to become disorganized. To re-organize a file based on physical addresses for pointers can be a very difficult task.

DMS has two basic methods for deleting records. Any group deleted using the REMOVE option is immediately deleted from the data base physically. The space it occupied on a page is made available at this time. Any data following it on the page is packed up toward the beginning, thus leaving a large contiguous area at the end of the page. Chains are re-linked around the deleted group, and as a result additional accesses may be required to the data base. It is for this reason that the DELETE verb was created.

Using the DELETE option the group is flagged as a deleted group the same as in DL/I but with one big difference. If the group can be deleted and chains re-linked without additional accesses, it is physically removed as in the REMOVE option. Otherwise chains will be re-linked during subsequent processing which requires these pages to be accessed anyway. This greatly reduces the number of accesses to the data base, and increases the overall efficiency. Two points should be remembered here:
- Any deleted data is removed physically from the data base, either immediately, or on a subsequent (free) access to the involved pages.
- File re-organization is maintained dynamically and automatically by DMS.

This feature causes DMS to be ideally suited for such data bases as may be found in a Hotel reservation system, say, where the data base is continuously "breathing" or "pulsating" -- growing larger during the hours when guests are checking-in and smaller during that time when checking-out is at a peak.

- Access Methods: DL/I uses HSAM (Hierarchical Sequential), HISAM (Hierarchical Indexed Sequential), HDAM (Hierarchical Direct), or HIDAM (Hierarchical Indexed Direct) for accessing the data base. HSAM being sequential is trivial in this discussion. HISAM and HIDAM both start out every unique access via the ISAM route. ISAM requires first a search of the Master Index to locate the cylinder, then it must access that cylinder and search the track index at the top to find the proper track, then must access the track to find the data segment desired, which may be on some other cylinder in the OSAM area. At this point additional accesses may be required chasing around following the pointers to the desired segment. It is probable that more users will resort to the GET NEXT call more often than the GET UNIQUE, simply to reduce access time, which is tantamount to sequential, batch processing. This is exactly what DMS was designed to avoid.

How does DMS access groups? First, it uses the RANDOM file organization under BPM File Management. Using the RANDOM organization, a block of contiguous pages is allocated to the data base. Data is accessed by supplying File Management a relative granule (i.e. page) number, and the data page is fetched immediately with only one access to the disk. This is probably the fastest method known for accessing data from direct access devices. No Master Index or multi-level index search is made for the key value. The price for this is the requirement that the user (in this case DMS) do his own bookkeeping of data addresses. DMS will optionally compute random addresses directly based on data content in specified fields, or the user may specifically address a page number himself, if he should so desire.

Between these two extremes are a number of options which provide the DMS user a wide variety of selections in determining the manner in which his data is stored and accessed over the available disk space. He may confine certain groups to a specific range of pages, or he may ask for them to be distributed at specified intervals across the entire data base. He may specify that certain data groups be stored on the same page as their master group (if space permits) and reduce the number of physical accesses in stepping around the chain. He may directly access the master group through the use of header pointers (which are not available in DL/I).

Under DL/I every segment must have a sort key before it may be inserted into the data base. Groups may be inserted on DMS chains chronologically without sort keys, if desired. Later accessing can be in either direction around the chain.

Under DMS, data may be stored and accessed according to the relationships it has in real life. Under DL/I it must first be organized in a Hierarchical structure which can later be viewed and accessed according to this logically sequential hierarchical fashion.

- Data Independence: Regardless of which of the data structures or access methods are used, a DL/I application program is typically insensitive to the particular organization or access method. Declaration of the logical data base record structure is made separate from the application program. This allows upward compatibility from HISAM, say, to HIDAM without the necessity of recompilation. It follows that verbs used to access data will be rigidly fixed and not be varied to take advantage of the more sophisticated file structure offered as a result of Version 2. There are five verbs used for accessing data:
    - GET UNIQUE: Retrieves a unique segment
    - GET NEXT: Retrieves the next sequential segment
    - REPLACE: Replaces data in an existing segment
    - DELETE: Deletes an existing segment
    - INSERT: Inserts a new segment

Actually the GET NEXT possesses the ability to GET NEXT WITHIN PARENT, which allows accessing subordinate segments once the root segment has been found, so that a program may skip along at the same level or search down the hierarchy, but in a sequential manner.

In contrast to this DMS is provided with a large vocabulary of verbs to handle the data to be processed. For example:
- STORE: Stores a group according to specifications in the DDL statement
- DELETE: Deletes a group along with all its dependent detail groups
- REMOVE: Same as DELETE, but physically removes data immediately
- MODIFY: Modifies items within a group as specified in working storage
- FIND: Retrieves requested group occurrence from data base to buffer

There are a number of FIND verbs which can be used, not just to locate a unique group, but for stepping around chains either forward or backward, for finding the master group of a specified chain, searching the data base for the first group occurrence that falls within a range of reference codes, searching serially from an initial reference to some limit, finding group occurrences that match a specific value (file inversion using secondary indices), plus a number of others.
- GET: Moves data from the buffer to the user's working storage
- HEAD: Retrieves the master group of a specified chain and moves it to the user's working storage

It is this rich repertoire of verbs that give the DMS user a tremendous advantage over DL/I in allowing him to take advantage of the special data relationships defined under DMS. Furthermore DMS programs are likewise insensitive to data file re-organization provided the original relationships are not destroyed when the data base is re-defined. This is probably the most important point to be considered in comparing DMS to DL/I. Although the logical relationship of the data base under Version 2 may be linked by chains, very little is to be gained if processing programs can not utilize access methods which take advantage of this set of data relationships.

- Data Base Recovery: In Version 1 (the present IMS) data base recovery is a feature of IMS, not DL/I. The IMS system maintains a journal file on tape of all transactions against the file. In the event of a data base destruction for some cause, the latest good checkpoint may be restored, and the journal of transactions may be re-processed. When using DL/I alone in a batch processing environment, the journal file is not available unless specially coded into the application program.

DMS provides its own journal file regardless of the source of the transaction, be it from terminal or batch background. The journal file is program independent; it stores before and after images of each page modified. There is no need to re-process transactions to restore the data base. Furthermore, the utility programs are capable of restoring only selected ranges of the data base if desired, thus saving a considerable amount of time.

According to the literature, Version 2 will carry a journal file for batch processing as well as IMS transactions. Furthermore, they will carry physical images rather than transactions.

● Data Sensitivity: Sensitivity is the term used to describe privileged access to certain Segments of a DL/I data base. An application program may be declared insensitive to certain segment types, or may be placed in a read-only status with certain segment types.

Under DMS, individual items within a group may be specified insensitive to reading or writing by means of a set of Query/Access codes which are associated with passwords rather than programs.

● Data Base Definition: DL/I uses a DBD (Data Base Description) processor which is similar to the File Definition Processor of DMS. Some eight or ten statements are available which are roughly similar to the DDL of DMS, except that DMS includes a CHAIN statement, permitting greater flexibility in the specification of data relationships.

DL/I data types are X, P, or C (Hexidecimal, Packed, or Combined). DMS has five types: Numeric, Alphanumeric, Packed Decimal, Binary, Floating Point short and long.

DMS produces a COBOL copy file automatically when the DDL is processed by the File Definition Processor. DL/I requires data definition to be defined manually. They can still use a COBOL copy file but it must be created manually, and modified manually each time the data base definition is changed.

● Space Allocation: Space must be allocated for DL/I data bases by the user, inserting DD cards in his JCL. Due to the number of different types of space handled by DL/I it is possible to waste space and still not hold the entire data base. Choice of sizes also affects the performance drastically. Take the case of an HISAM data base being defined for a 2314 disk storage system.

This would probably require three DD cards, one for the index space, one for the prime data space, and one for the OSAM overflow areas.

The prime area usually holds the majority of data base records and all the index records. For any given data base a record might vary from only the root segment to the root segment plus all occurrences of all its dependent segment types. To define the ISAM logical record size (LRECL) large enough to hold the largest record could cause an enormous amount of wasted space, particularly if only 10% of the data records were this large. To define the ISAM logical record length as small as the root segment only would cause the majority of records to spill over into the OSAM overflow area, and would be costly in terms of additional seeks jumping from track to track.

The user, in fact, must perform a careful analysis of his data to determine what percentage of his records require 100 bytes, 200 bytes, etc. After forming his distribution curve, he must perform further analysis to determine the proper trade off of wasted space versus performance for his application. After establishing the size of the ISAM logical record for his prime data area, he must continue his analysis to determine the size of his OSAM overflow area.

This is not an exaggeration, but actually an over-simplification of the number of considerations involved in determining space allocation for the DL/I data base.

DMS does not have all these different kinds of spaces, since it was not designed on top of a kludged up system. All pages in DMS are fixed in length to 512 words. The number of lines per page varies according to the size of the data base (i.e., total number of pages), but is always optimized to provide the maximum for any given data base size. This is not meant to imply that the user does not have options for storing his data. As stated earlier, he may request that certain groups be automatically placed near others, or spaced uniformly throughout a range of pages, etc. But DMS performance is less sensitive to becoming degraded as a result of poor space allocation, and, due to its design, will make much more efficient use of space than DL/I.

This factor is extremely important in applications involving a data base that is constantly growing larger and smaller periodically, requiring automatic re-organization and efficient use of all available disk storage space.

● File Maintenance: In discussing HIDAM, later, it is mentioned that HIDAM consists of an ISAM file of indices and physical addresses to the HDAM data base. For this reason, a second, or third, etc., ISAM file can be maintained for the same data base but sorted according to different keys. No mention is made of how these indices are created and maintained. It is strongly suspected that much of the maintenance is placed under the responsibility of the user. This takes a lot of the magic out of the capability, especially when problems of multiple index updating at the user level are considered.

This is one of the powerful features of DMS. Defining chains which order a data base according to different keys is extremely simple with the Data Definition Language provided and the powerful File Definition Processor. A group may be defined to have any number of chains linking it to other groups by different sort keys. Updating is quite simple. A new group may be inserted, for example, with a single call to the INSERT verb. DMS will automatically establish linkages for all chains passing through that group. It is not known whether DL/I can perform this automatically, but if it is advertised as capable of more than one logical sort sequence, this capability of file maintenance should certainly be questioned.

- File Inversion: DMS automatically maintains a secondary index on any item defined in the DDL with the INDEX parameter (INVERTED is specified on the group definition). Secondary indices are automatically updated during normal file maintenance activities involving that item. Special verbs have been provided to permit secondary indices to be searched when retrievals involving partially inverted files are performed. Sorting of these values is done automatically by DMS. With DL/I this is all the responsibility of the user (even though file inversion is claimed).

- Other Features: There are a number of less significant features which can be compared, but in most cases DMS is superior or they are a stand off. For example, both are written in assembler language, both occupy approximately the same amount of core storage, and both maintain directories or schema separate from the data base. DMS has a number of small features such as automatic check sum maintenance on each page (optional), and inventory pages for automatic space management. Both operate under several host languages. (DL/I under PL/I, which is not available on Sigma systems, but DMS operates under FORTRAN and handles floating point information.)

  Finally comes the question of availability. DL/I Version 1 is now in the field under IMS, but has none of the chaining capabilities of Version 2. DL/I under IMS Version 2 is announced for March 1, 1971.

  DMS is announced for January 1971 availability, but is running now and can be demonstrated.

OVERVIEW OF DL/I

DL/I DATA STRUCTURE & DEFINITIONS

DL/I views all data as hierarchical. A logical record under DL/I can be a rather large piece of data. It consists of smaller pieces called "segments" which more closely resemble the old fashioned records that most people are familiar with. The following definitions may prove helpful:

- Segment: Fixed Length. The basic data element that interfaces between the application program and DL/I.

  The DL/I Segment corresponds to the DMS Group. A segment may be broken down further into fields just as the DMS Group is broken down into Items. A root segment in DL/I would correspond to a Master Group in DMS, and a subordinate, or "Child" segment in DL/I would correspond to a Detail Group in DMS.

- Logical Data Base Record: A hierarchically related set of fixed length segments. Always viewed by the application program as a tree structure of segments, it may contain up to 15 hierarchical levels of segments. Any, or all, of these segments may have multiple occurrences called "twins". DL/I uses the terms "Parent", "Child", and "Twin" to describe the logical relationship between segments of a record. Figures 1 and 2 illustrate examples of logical data base records.
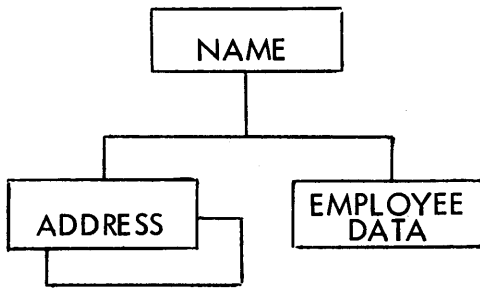
Logical data base record
Figure 1
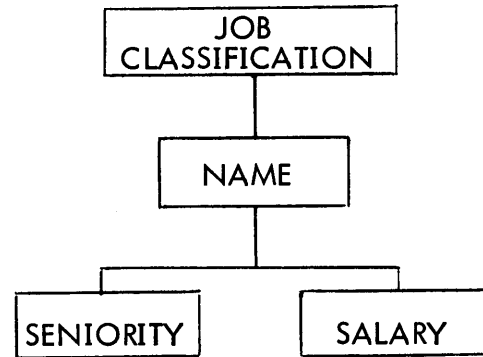
Logical data base record
Figure 2

Each block in the two figures represents a segment. In Figure 1 the NAME segment is the root segment of the data base record. The ADDRESS and EMPLOYEE DATA segments are subordinate, or child, segments and their parent is the NAME segment. Note that there are two occurrences of the ADDRESS segment. These are referred to as twins.

- Logical Data Base: A set of logical data base records stored in the DL/I organizations (i.e., SAM, ISAM, or OSAM). Typically one or more OS/360 data sets.

DL/I PHYSICAL STORAGE ORGANIZATIONS AND ACCESS METHODS
There are two physical storage organizations, each with its own access method. These are summarized below:

- Hierarchical Sequential: Access Methods are HSAM and HISAM
  - HSAM (Hierarchical Sequential Access Method)
        Tape or Disk -- Based on OS/360 Sequential Access Method(SAM)
  - HISAM (Hierarchical Index Sequential Access Method)
        Based on ISAM and OSAM

- Hierarchical Direct (New with Version 2): Access methods are HDAM and HIDAM
  - HDAM
    Direct address to the OSAM data area. Uses algorithmic addressing (User provided algorithm optional)
  - HIDAM
    HDAM organization with indexes added. Indices are held separate from data. They are kept in an ISAM file and consist only of Keys and Direct Addresses which point to the data records.

OVERFLOW SEQUENTIAL ACCESS METHOD (OSAM)

ISAM has an overflow capability built in, but this operates on entire records and does not allow splitting a record between two tracks. It is possible that a logical data base record as defined above will not fit in the physical space available in the ISAM logical record. In this case it is allowed to overflow into an OSAM overflow area.

Each physical data base record starts within an ISAM logical record. As many segments of the physical data base record as can be accommodated are placed in the ISAM logical record. If additional space is required to store segments of this data base record, one or more Overflow Sequential Access Method (OSAM) blocks are used. Direct addresses relate the ISAM logical record and all OSAM physical blocks for one physical data base record.

The overflow blocks accessed by OSAM are defined in the user JCL via data definition (DD) cards. They are standard OS/360 physical sequential data sets.

Under Version 1 (Now in field) of IMS, these blocks were treated by OSAM as sequential data sets. OSAM blocks were linked together through pointers if additional space were needed. A special count field with zero length for its associated key and data fields was used to mark the end of data stored in a block. With Version 2, OSAM will allow direct physical addresses to these blocks as before, but under HDAM segments will be linked together in this overflow area.

Figure 3 represents the manner in which the data structure of Figure 2 would be stored under existing IMS. Multiple occurrences of a data type are stored physically sequentially (this is true for both sequential and index sequential organization). In Figure 4 it can be seen that, under Version 2, multiple occurrences, or Twins, need not be placed in juxtaposition with one another, but may be scattered over the data base and connected by pointers.

HIERARCHICAL DIRECT ACCESS METHOD (HDAM)

HDAM provides the ability to access the OSAM file organization described above using direct physical addresses. Algorithmic addressing is provided, and the user may provide his own algorithm optionally.

HIERARCHICAL INDEXED SEQUENTIAL ACCESS METHOD (HIDAM)

HIDAM is used to provide an indexed access support of the hierarchical direct organization. An ISAM file is created consisting only of Keys and Physical Addresses to the segments in the OSAM overflow area. No segments are carried in the ISAM data set; only pointers to the data. For this reason, it is possible to have a second set of keys in another ISAM file pointing to the same data in a different sequence (i.e., based on another key). HDAM and HIDAM provide Version 2 with data relationships which more closely resemble that of DMS.
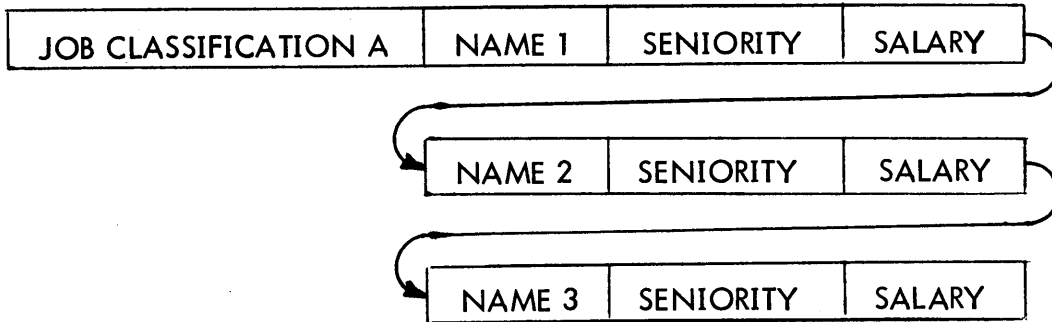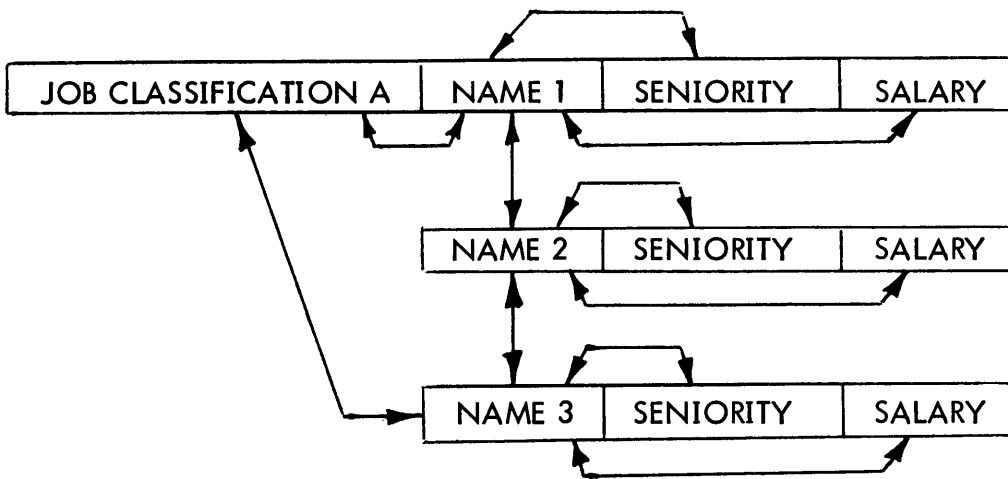
Figure 3

Figure 4

CHAINS

When the HDAM or HIDAM organization is used, segments are linked together by pointers. The first and last segment are linked to the root segment (cf. Fig. 4), and thus become a closed chain. The pointers may be physical addresses or symbolic keys.

POINTERS

Redundant segments may be replaced by pointers (Figure 5). This is true even if they belong to different data bases. (In the figure, if the data base on the left were an HSAM organization and the one on the right an HDAM organization, then the pointer must be a symbolic key since direct addresses may not be used to point to members of sequential files.)
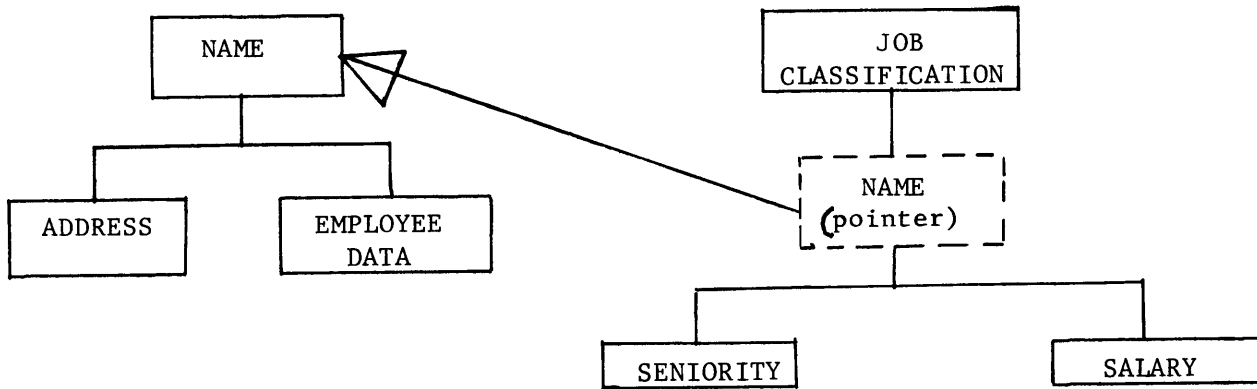
Figure 5

INVERSION

If the relations shown in Figure 5 carried a pointer in the reverse direction also, it would be possible to enter the data base on the left, and determine all job classifications associated with a given name occurrence. This can be viewed logically as the structure shown in Figure 6, and is termed File Inversion by IBM.
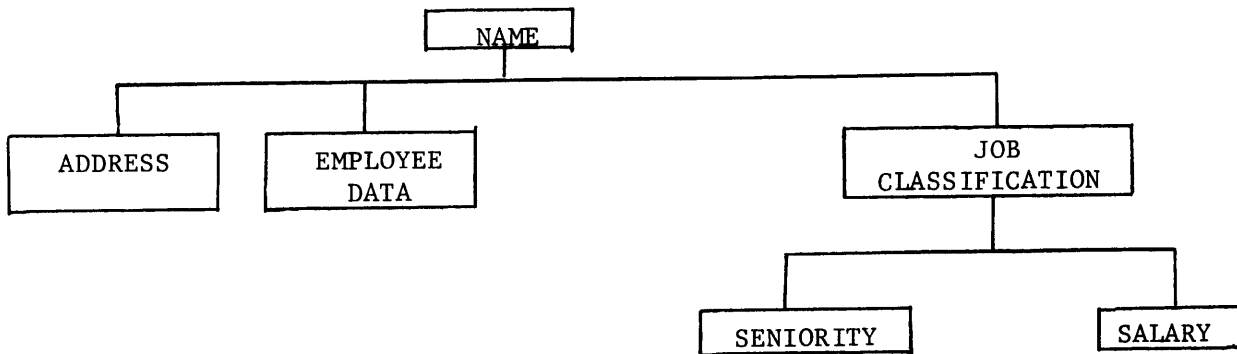
Figure 6

SECTION VI
MARKETS/APPLICATIONS

6.0 <u>INTRODUCTION</u>

The Sigma 7 introduction in 1966 marked a new concept in computer architecture. Sigma's memory orientated design and versatile I/O structure gained wide acceptance in general scientific, time-sharing, and educational markets. The Sigma 9 — with up to three times the power of Sigma 7, expanded memory, and new reliability features — continues this philosophy of offering maximum performance at economical cost.
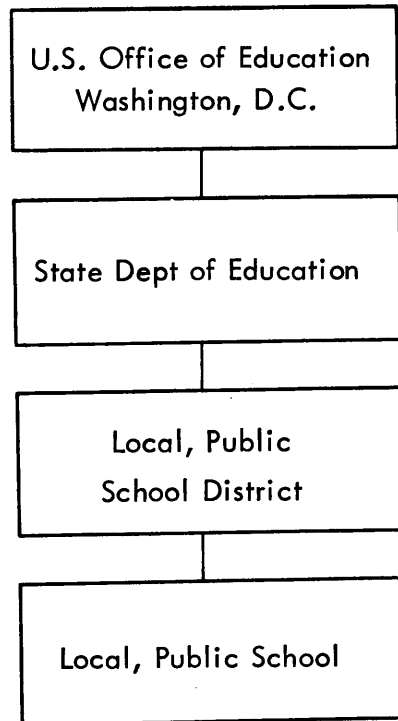
Some of the most significant features of Sigma 9 are evidenced through its software. UTS, designed for large time-sharing applications, will allow XDS to expand its base in this major marketplace. Industry forecasts indicate in-house time-sharing will grow dramatically during the next 3 to 5 years. Established service bureaus will continue to be excellent prospects and will, in turn, prove to be credible references for increased penetration of the growing in-house time-sharing market.

For the commercial and educational user, XOS offers proven teleprocessing capabilities, together with data base manipulation and concurrent multi-programming of batch jobs. This combination of XOS and Sigma 9 provides an economical solution to most large scale commercial applications which, in turn, will assure XDS of broader penetration of these markets.

The following discussion of education, general scientific, time-sharing, and data processing markets is intended to provide an insight into those market segments and applications where you can expect Sigma 9 to be most competitive. An understanding of industry requirements and the characteristics of Sigma 9 is necessary to guarantee the success of this major new XDS product.
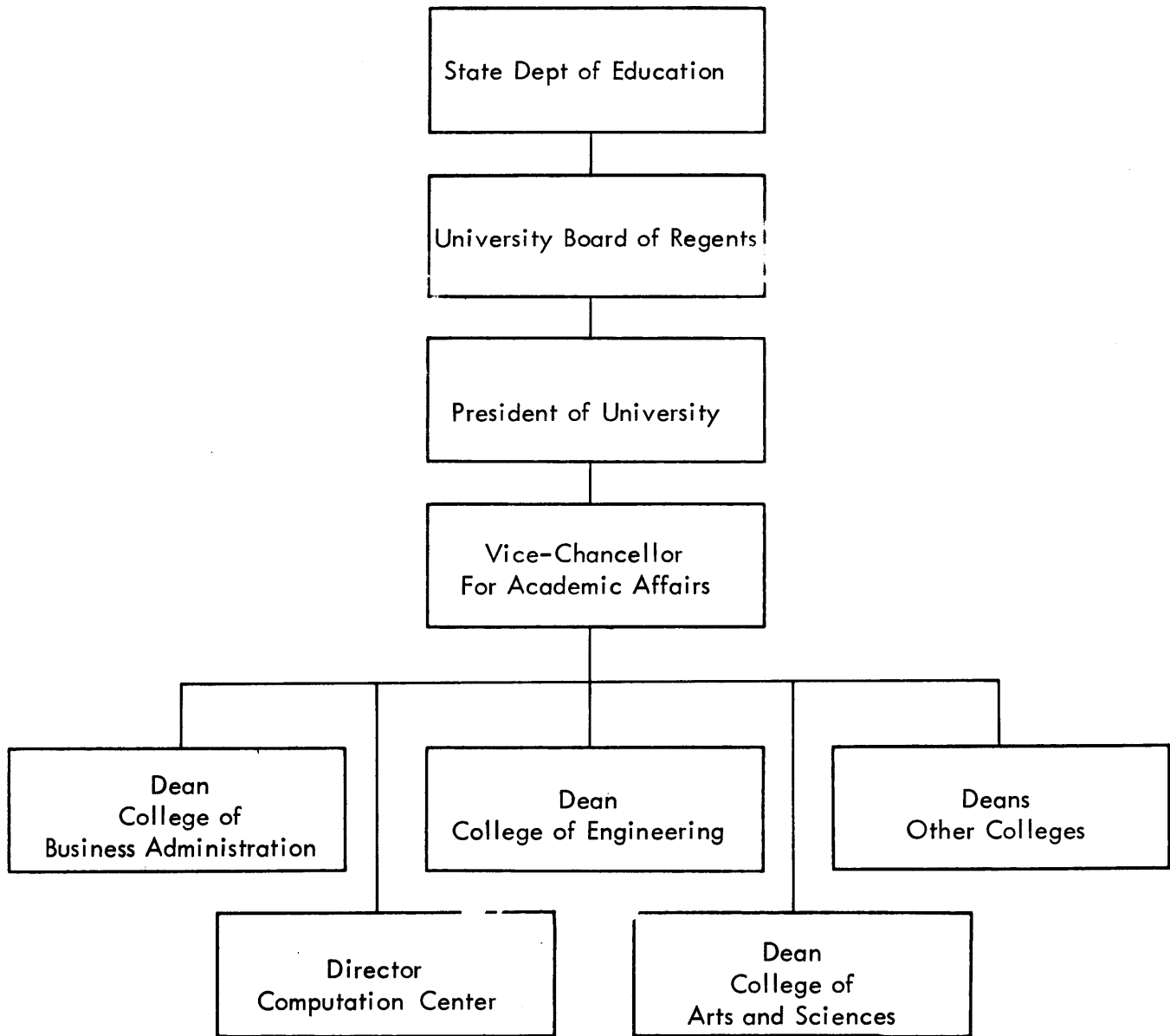
6.1 THE EDUCATION MARKETPLACE

To approach the Education marketplace properly, an organizational perspective is required, since several groups are usually involved in a data processing procurement. The following organizational charts provide such a perspective.

```
┌─────────────────────────┐
│  U.S. Office of Education│
│     Washington, D.C.     │
└─────────────────────────┘
             │
┌─────────────────────────┐
│  State Dept of Education  │
└─────────────────────────┘
             │
┌─────────────────────────┐
│     Local, Public        │
│    School District       │
└─────────────────────────┘
             │
┌─────────────────────────┐
│   Local, Public School   │
└─────────────────────────┘
```

While the U.S. Office of Education has little direct control over the local elementary/secondary school system, its influence is more than minimal due to the methods whereby school funds are made available under various federal programs.

The State Department of Education exerts strong influence on the public school system in the areas of curriculum development, average daily attendance requirements, etc. The Department may or may not be a factor in the procurement of computing equipment, leaving the decision to the School District Headquarters personnel.

```
                  ┌─────────────────────────┐
                  │  State Dept of Education │
                  └───────────┬─────────────┘
                              │
                  ┌───────────┴─────────────┐
                  │ University Board of Regents│
                  └───────────┬─────────────┘
                              │
                  ┌───────────┴─────────────┐
                  │  President of University │
                  └───────────┬─────────────┘
                              │
                  ┌───────────┴─────────────┐
                  │     Vice-Chancellor     │
                  │   For Academic Affairs  │
                  └───────────┬─────────────┘
```

| Dean
College of
Business Administration | Dean
College of Engineering | Deans
Other Colleges |

| Director
Computation Center | Dean
College of
Arts and Sciences |

In some cases, the State Department of Education has little or no influence upon the State University when the question of computer selection is involved; however, the department's influence should be thoroughly evaluated. In a typical case, recommendation is made to the Vice-Chancellor by an evaluation committee composed of the Director of the Computation Center and various college Deans, with ultimate authority being vested in the President who must, of course, secure concurrence from the Board of Regents.

Education is one of the fastest growing segments of our economy and affords outstanding opportunity for XDS. Basically, it consists of two distinct elements: (1) elementary/secondary schools,

and (2) schools of higher education. In total, it represents one of the United States' largest businesses, with an expected expenditure of approximately $70 billion in 1970. This is more than double the 1960 rate of spending and accounts for more than 7% of the Gross National Product. The growth of the American college and university in this century has been spectacular and has now reached a point of near-crisis for those charged with administering this dynamic enterprise of higher education. Enrollment of students has sky-rocketed from the 1900 level of 250,000, to the present level of 7,000,000, and is on its way to an anticipated 10,000,000 by 1975. Following World War II, facilities and staff for performing basic and applied research have greatly expanded, particularly with increasing grants and research contracts by the Federal government. There are approximately 60,000,000 children presently in the elementary/secondary age group today. It is expected that this number will increase during the next several years; then, there will be a decline in the elementary group with increases in high school and college enrollments.

The Education market is generally characterized as one requiring a multi-use, concurrent computing capability. With the availability of the Sigma 9, XDS is in an excellent position to broaden its share of this market. XDS has already achieved credibility in this environment, particularly at the higher education level.

Historically, universities have been susceptible to the "reference sell" technique. In this connection, basic information on specific XDS university accounts may be located in the University Directory, recently placed in each of our sales locations.

6.1.1   AREAS OF INVOLVEMENT

Automated data processing is presently being utilized in three broad areas of the Education market:

- Administration
- Instruction
- Research

6.1.1.1 Administration

Traditionally, this area is one of the first to be considered in a program of mechanization for an education account. A point made by Caffrey (1967)[1] should be understood by the XDS salesman in order to place himself in an advantageous position with his educational opportunities. Caffrey indicates that any time an administration converts to an automated system, the resulting system is always an improvement over existing methods, simply because those involved in the system had to stop and decide "what is wanted and why", and that "this decision process alone is worth the price of the machine".

Among the numerous applications in the Administrative area are:

- Payroll and Personnel Management
- Accounts Payable
- Income Accounting
- Investment Management
- Purchasing and Materials Control
- Plant and Equipment Management
- General Ledger
- Long Range Planning
- Resource Scheduling
- Admissions
- Registration
- Student Accounting
- Class Lists
- Testing and Counseling
- Grade Reporting

---

[1] Caffrey, J.
 1967 "Computers in Higher Education", in Bushnell, D. D. and Dwight W. Allen (eds.). The Computer in American Education. New York: John Wiley and Sons, Inc.

- Permanent Records
- Alumni Records
- Gift Accounting
- Library Control

Registration and Grade Reporting are typical examples of applications requiring the maintenance and manipulation of large, highly dynamic data bases. The matching of students, faculty, and facilities several times yearly must be done to assist department heads in determining what classes to schedule. Grade Reporting and the associated postings also generate large workloads. Sigma 9, with associated inquiry capability, affords the Administrator and faculty an opportunity to improve these areas with significant reductions in manual effort. A typical data flow in an academic environment is offered on the following page.

6.1.1.2   Instruction

Various computer approaches are utilized to impart knowledge to students. In some locations, even at the elementary level, children are using on-line terminals to learn. In other cases, particularly at the secondary and university levels, students have access to computing facilities for performing class assignments and research projects.

Although there is considerable activity in the instructional area, vast territory remains unexploited. The Pierce Report, prepared by the Presidents Science Committee, states "We believe that under-graduate college education without adequate computing facilities is deficient education, just as under-graduate education without adequate library facilities would be deficient education. At present, deficiency in computing is wide-spread".

In this report, it is claimed that the majority of college students are receiving a second class education because most do not receive adequate computer-assisted training, if any. A study of data processing expenditures for computers in the education community indicates that only 5% of students received adequate computer service in 1965. There is a growing recognition that the computer capacity at a college or university is part of its total power to attract both students and faculty.

REGISTRATION

OTHER PREREGIS-
TRATION MATERIAL

COURSE AND
FACULTY DATA

APPROVED
CLASS REQUEST

CLASSROOM
ALLOTMENTS

STUDENT
DATA
MASTER

STUDENT
ACADEMIC
MASTER

COURSE
TIMETABLE
MASTER

SIGMA 9

INQUIRY
—
UPDATE
TERMINAL

MAILING LABELS

I.D. CARD

DIRECTORY

SCHEDULES

ENROLLMENT
REPORTS

CLASS ROSTERS

GRADE REPORTING

GRADES POSTED
TO CLASS ROSTER

INQUIRY
—
UPDATE
TERMINAL

SIGMA 9

STUDENT
DATA
MASTER

STUDENT
ACADEMIC
MASTER

STATISTICAL REPORTS

PERMANENT RECORD

FINAL GRADE REPORT

MISSING GRADES
REPORT

There is continuing activity in the Computer Assisted Instruction (CAI) area, but this market is virtually untapped at present. Educators are experimenting with various methods and languages to implement CAI; however, no clear trend has developed. Systems Development Corporation, Santa Monica, has developed PLANIT (Programming LANguage for Interactive Teaching), a general purpose teaching system which allows a lesson designer to enter course content into the computer for use as a teaching device. The user (lesson designer or student) communicates with the system via a keyboard device linked by either TWX, telex, or telephone to the computer. Interacting with PLANIT, the user can build and edit lessons, present lessons, and perform computations. Typically, an author using the facilities of PLANIT, designs a lesson (i.e., mathematics, history, psychology, etc.), edits and stores it for subsequent use by the student. The student, interacting with the computer via a terminal, progresses through the lesson while computer keeps track of his progress. PLANIT permits the designer to specify conditions for branching for response latency, number of errors, etc., based upon the student's performance. Sigma 9's interactive time-sharing capability ideally suits it for this environment. PLANIT is available to XDS customers.

6.1.1.3   Research

The research environment affected the design of early-day computers and continues to employ computers heavily. Applications in research are unlimited in variety and new applications are found with great frequency.

Time-sharing or remote-access computer capability plays an increasing role in the research area. Closely related to problem-solving aids in instruction, the use of time-sharing terminals is spreading into every discipline on the university campus, with uses ranging from mathematical and statistical calculations to complex simulation problems.

Examples of XDS education customers employing computers in research are:

    Air Force Academy - Hybrid Computing
    American Council on Education - Social Sciences
    Michigan State University - Nuclear and Astro Physics

Montana State University - Chemical and Electrical Engineering

Queens College - Real Time Experimentation

Saskatchewan, University of - Simulation of Power Systems

Stanford University - Radio Science

Washington University - Computer Graphics

Research work utilizing a computer frequently employs programs with a large number of re-iterative steps. The cache memory of Sigma 9 will permit this work to be done in less time.

Additional information on these or other scientific applications are available upon request.

6.1.2   SIGMA 9 — UTS AND XOS

One of the major requirements in the educational marketplace is the maintenance and manipulation of large, highly dynamic data bases. Sigma 9 offers on-line maintenance, high reliability, and sophisticated data base manipulation. By taking advantage of the capability to place data on line for direct inquiry and updating, XDS can satisfy the Administrator's desire to operate with current information from financial, student records, alumni, and library control areas.

6.1.2.1   UTS

Student Instruction and Research will continue to exert pressures for increases on line interactive time-sharing, real time, and overall computational capability. UTS meets this requirement at present, as well as providing a base from which to grow vis-a-vis modular memory expansion, IOP's, peripheral compatibility, etc.

Most education computing facilities experience wide fluctuations in workload conditions. Through the facilities available with UTS, installation management can dynamically "tune" the system during operation, to adjust for varying conditions of usage. Operating personnel can quickly and easily alternate operations between full batch usage and concurrent batch/on-line usage, through the use of UTS terminal startup/shutdown facility. By analyzing system utilization, installation management may allocate resources to the areas where they are most needed

and in the desired proportions. For example, it may become necessary to restrict the on-line users and allocate the total system to process the workload during the registration and grade reporting periods.

By virtue of the dynamic nature of education computing, it becomes increasingly important to keep the system operating to its maximum. UTS's provision for diagnostic and recoverability capabilities, as well as its ability to re-allocate resources for a degraded mode of operation, are outstanding attributes of the system and should be fully understood by the education user.

The Education market has unique requirements involving the concurrent handling of on-line interactive time-sharing, batch, and real-time operations. The capabilities available in the combination of Sigma 9 and UTS provide a high throughput, high price/performance system for the user.

Among the advantages of using Sigma 9 and UTS in an academic environment are the following:

- Availability of more detailed data and high speed processing permit a more thorough analysis for planning and fund justification
- Administrative officers and faculty are released from numerous time-consuming clerical tasks to perform their primary jobs
- The ability to use the system as a basis for building a simulator of future actions could well provide the greatest return of any application
- Peak-load deadlines can be met sooner, less expensively, and more conveniently
- The logical ability of the system may be used for automatic editing of data to save on clerical time and costs
- The ability to centralize and process the numerous files reduce over-lapping assembly of information with corresponding reduction in clerical effort
- High speed processing and printing of student schedules, student directories, grade reports, budgets, annual reports, etc., permits reduction of time and printing costs in preparing these documents
- An enhanced data processing capability to assist much-needed research work could help a research team acquire government funding

6.1.2.2 XOS

The education marketplace requires the concurrent handling of batch, remote batch, data base manipulation, and real time processing. The powerful facilities available with XOS can be used to enhance the capability of Sigma 9 to fulfill the multi-use requirements of this market.

XOS has outstanding multi-programming performance through its job management/resource allocation capabilities. Multi-programming provides for maximum CPU utilization — an important consideration in the education environment which has a particularly heavy workload. In addition, through intervention at the operator's console, a job with a lower priority may be scheduled for immediate execution. This is a major advantage in an academic environment since the user is thereby able to alleviate many of the "crises" situations encountered in education accounts.

Under XOS, an unlimited number of parallel jobs, five operation jobs, and one serial job can be in core at any given time. This makes the use of Sigma 9 and XOS an outstanding price/performer and permits the education user truly to fulfill the TOTAL requirements of student instruction, administrative, and real time processing.

6.1.3 SUMMARY

The Education market is one of the largest businesses in America, with a current rate of spending equal to 7% of the Gross National Product.

Opportunities for XDS in education involve the academic, instruction, and research areas, requiring hardware/software capable of supporting concurrent batch, remote batch, inquiry/update of dynamic data bases, and real time operations.

There is an increasing demand for more sophisticated hardware/software from the vendors to education. XDS unquestionably has proven credibility in this environment. The price/performance ratio of Sigma 9, coupled with the outstanding facilities available under UTS and XOS, enhance our opportunity to occupy a larger segment of an influential part of the nation's economy.

6.2 GENERAL SCIENTIFIC

6.2.1 INDUSTRY DESCRIPTION

General Scientific areas which are potential Sigma 9 customers consist of a multitude of varying industries. These areas are characterized by industries requiring a high degree of engineering skills for their products and large amounts of bulk storage for product information.

A typical Sigma 9 user requires considerable hardware and software capability. For instance, the ability to provide a very large number of terminals to the user, with good response time, is necessary. This is provided by UTS and the high speed RAD. The user who requires a mixture of capabilities including multiprogramming is serviced with XOS. The ability to expand partition sizes to accommodate engineering programs (with unusually large blocks of core), while continuing normal operations, is another typical requirement. With two million bytes of core available, the Sigma 9 meets this requirement. Many of the typical Sigma 9 installations will consolidate their operational areas into one centralized organization. The high degree of compatibility of the Sigma 9 with the Sigma 5, Sigma 6, and Sigma 7, plus the increased instruction speed (1.5 to 3 times faster than a Sigma 7) will allow users to achieve increased cost effectiveness as well as raw compute power.

6.2.2 TYPICAL INDUSTRY AREAS AND APPLICATIONS

Examples of typical scientific organizations who require the extensive compute power provided by the Sigma 9 include:

- State and Federal Government Laboratories
- Weather Bureau
- Educational Institutions
- Steel Industry
- Automotive
- Manufacturing
- Airlines
- Medical Industries

- Aerospace
- Utilities
- Transportation

6.2.3   UTS

Large computer facilities in the areas mentioned above are concerned with compute power, job accounting capability, restart capabilities, file security, priority scheduling, remote batch processing, core availability, and general distribution of computer time for their normal and abnormal workload. The UTS monitor provides all of these facilities plus the capability to dynamically tune the system to optimally allocate available resources.

The services and functions provided by UTS to accommodate the large scientific user's demanding requirements are:

- Job Accounting — The centralized data processing department can accurately charge users for their processing
- Comprehensive File Management Service — Permits configuration changes without modifying production programs.
- Dynamic Core Memory Allocation — Assures maximum computer usage under the most varied environment
- Checkpoint Restart — Prevents costly delays for job reruns
- Job-Stack Priority Scheduling — Operation management can dynamically adjust priorities of jobs
- Remote Batch Processing — Permits centralized data processing computer control, with remote input/output capability
- Operational Activity Monitoring — Provides operations management with facilities to monitor and provide efficient service
- System-Integrity Facilities — Provides maximum security for user's files, even in the event of total system failure

6.2.4 XOS

XOS ideally serves that segment of the General Scientific marketplace which requires the concurrent handling of batch, remote batch, data base manipulation, and real time processing. The powerful facilities available with XOS may be used to enhance the capability of Sigma 9 to fulfill the multi-use requirements of this market.

XOS has outstanding multiprogramming performance through its job management/resource allocation capabilities. Multiprogramming provides for maximum CPU utilization — an important consideration in the batch scientific environment which has a particularly heavy workload. In addition, through intervention at the operator's console, a job with a lower priority may be scheduled for immediate execution. This is a major advantage in this environment since the user is thereby able to alleviate many of the "crises" situations encountered by the users of the system.

Under XOS, an unlimited number of parallel jobs, plus five operation jobs and one serial job, may be in core at any given time. This makes the Sigma 9 and XOS an outstanding price/performer and permits the user truly to meet his total requirements.

6.2.5   APPLICATION PROGRAMS

XDS also provides a varied library of processors and applications for most engineering type processing. Program products consisting of extended FORTRAN IV-H and FORTRAN IV, BASIC, SL-1, DMS, and FMPS provide processing capability required by a scientifically-oriented company.

Most engineering areas have special application programs which require both high throughput and large blocks of core. Examples of these applications are:

- Astronomy systems
- Telemetry data analysis
- Satellite positioning

- Large matrix computation
- New product development
- Varied linear programming applications
- PERT

For these types of applications, operations personnel can use the "fine" tuning capability of UTS to adjust to unusual requirements, while continuing to provide other areas use of the remaining resources. Where applications require real time and batch processing under an umbrella of multiprogramming, XOS provides the ideal solution.

6.3   TIME-SHARING SERVICES

6.3.1   INDUSTRY DESCRIPTION

The computer service area is currently one of the fastest growing segments of the computer industry, and service bureaus are now faced with the necessity of expanding their capability in order to meet the needs of the changing service market. The service area in the past has been dominated by the batch service bureau; but, this method of operation has approached its limit of expansion. Scientific time-sharing has also enjoyed an extremely rapid growth rate, but it, too, has reached the point where further expansion depends upon improved services and additional software packages made available to customers.

Independent service bureaus have been extensively used on an interim basis by companies planning to install their own equipment, and for overflow, back-up, and special one-time requirements. However, the economics of such limited activities have become less and less favorable for the independent operator as fixed costs have increased substantially and the professional content of his service has risen markedly. The independent service bureau cannot afford to have its customer base turning over as frequently as customer functions inherently demand. It has become clear to many service bureau managers that they must convince a large segment of the business community that the service bureau can provide all of the data processing needs of a customer on a long-term basis.

Users of data processing equipment have endured the reality of installing three generations of hardware and software in the short span of 10-12 years, and a considerable amount of the glamour has faded. Pride in owning and operating the tools of his trade has diminished considerably as the realities of implementation and day-to-day administration have over-shadowed his business. Even large organizations have considered using external assistance because in-house administration of the complex skills required to meet their data processing requirements have become a built-in disadvantage.

Technological developments in hardware and software — particularly in telecommunications, data management, and mass memory — have significantly increased the advantages of off-premise processing to the user. The location of the computer has become far less important simply because the customer can be in instant contact with it by means of communication lines. Moreover, if necessary, his data files can be continuously available to him; and finally, the work of many users can be going on simultaneously.

Another factor which favors the service bureaus' future viability is the increasing development of the so-called pre-programmed applications or "packages". More and more data processing users, large and small, realize that complete customizing of their systems and programs is expensive and wasteful, that it is not necessary or desirable to "reinvent the wheel" in the areas of basic business record keeping. Technology has provided a big assist in this regard, because the flexibilities and capacities of present hardware and software pre-programmed systems are designed with considerable user options. Service bureaus are in a unique position to capitalize on this trend because of their close interface with the marketplace and their knowledge of customer needs.

6.3.2   SERVICE BUREAU NEEDS

Of the 2500 service bureaus in this country, approximately 80% provide only batch service capability in a mono-programming environment. Despite the continued growth rate of 20 to 30% in revenue, the center's "bread and butter" is being siphoned off by the in-house computer with surplus time available for sale.

As predicted earlier this year by the ADAPSO's (Association of Data Processing Service Organizations, Inc.) newest economic study, the computer service industry for 1970 will continue its rapid growth and exceed $2.4 billion in revenue by the end of the year. Regardless of the reduction in the growth rate of the general economy, ADAPSO believes the computer industry will continue to expand in 1970 and 1971.

Since the time-sharing and remote batch segment of the industry will continue to grow, a large portion of the mono-programming local batch only service bureaus will eventually have to provide multi-use capability in order to survive. All of these opportunities are open to XDS, but it requires an effort by Sales to inform these people of the Sigma 9 multi-use capability, especially for those applications requiring remote job entry, large data bases, and large partitions of core dynamically made available to a user.

6.3.3   SIGMA 9/UTS/XOS TIME-SHARING CAPABILITIES

The greatest potential customer of the service bureaus are the commercial users and the engineering applications which require bulk storage and large partitions of core in which to manipulate their data.

Remote batch services capability of the Sigma 9 provide potential users with the advantages of an in-house computer installation as well as decreased overhead expense with expanding interactive capabilities. The Sigma 9 system offers vastly improved system reliability, assures the service bureau customer of minimum down-time, and the service bureau of maximum computer availability, maximum profits, and minimum embarrassment.

The local batch oriented service bureau will now be able to increase their capability and cost effectiveness from XDS's high performance multiprogramming operating system. Up to five problem programs, such as sequential batch processing, random processing, communications, disk or tape sorting, and engineering type applications may run concurrently. The operating system, XOS, which controls the problem programs, employs the technique of the distribution of processing time to independent programs based upon priorities, time slice allocation, and input/output device utilization. XOS assures all users an even distribution of process time.

Time-sharing service bureau customers have reported increased use of time-sharing as well as the fact that a time-sharing capability has had a significant positive impact upon their business. They also predict that future applications will be even more expansive as the capability of time-sharing systems is improved and they become more skillful in applying these new techniques to their operations. The Sigma 9 system provides a base from which users will grow in time-sharing capability.

6.3.4   PRIMARY APPLICATIONS OF TIME-SHARING

Listed below are some of the most currently used applications in the various industry groups:

6.3.4.1   Banking

- Savings and demand deposit forecasting model
- Financial lease analysis
- Credit scoring (ranking) analysis
- Cash flow projections
- Technical analysis (access to special on-line stock market data files)
- Installation loan monitor program
- Financial statement preparation (individual clients)
- Financial analysis (ratio analysis using 10-year historical data)
- Effective yield-to-maturity calculations
- Pension fund performance model
- Portfolio analysis
- Scheduling loan payments
- Inventory control (client)
- Financial lease rate calculations
- General purpose financial statement analysis

6.3.4.2 Manufacturing — Industrial Goods

- Production scheduling (special linear programming model)
- Cost analysis (direct labor)
- Bulk distribution planning model
- Project control monitor
- Project budgeting evaluation
- Inventory control analysis
- Market evaluation system
- Direct labor planning model
- Monthly sales analysis
- New product cost analysis
- New product pricing strategy studies
- Cash budgeting
- Project evaluation analysis
- Preparation of N/C tapes and machining
- Automatic mechanical design and layout
- Automatic instrumentation service
- Scheduling men, machines, and operations

6.3.4.3 Manufacturing — Consumer Goods

- Sales forecasting model (algebraic)
- Financial statement preparation
- Rate of return analysis (project evaluation)
- Alternative project evaluation (decision tree analysis)
- Sales forecasting (market simulation)
- New product pricing analysis
- Inventory system analysis
- Credit screening
- Corporate-wide, long-range planning model

- Machine utilization analysis
- Overhead distribution program
- Accounts receivable analysis

6.3.4.4   Investment

- Legal capital requirements evaluation
- Corporate earnings projection studies
- "Fast mover" stock detection analysis
- Municipal bond package (yield analysis)
- Bond redemption calculations
- Technical evaluation of utilities
- Bond bidding and pricing models
- Portfolio evaluation (current market valuation) and selection
- Economic forecasting
- Real estate financing model
- Merger analysis (balance sheet and per-share earnings impact)
- Rate of return analysis (capital budgeting)
- Various technical analysis programs
- Pro forma projection program

6.3.4.5   Transportation — Public Utility

- Inventory system modeling
- Comprehensive tax projection studies
- Financial ratios and operating statistics determination
- Credit scoring
- Job shop scheduling (simulation model)
- Flight crew manning and pay model
- Distribution system network analysis
- Time of day aircraft scheduling
- Profit budgeting

- Rate of return analysis (project evaluation)
- Personnel requirement projections
- Pro forma financial statement program
- Market forecasting
- Facilities location (simulation model)

6.3.4.6 Management Consulting

- Facilities location model
- Pension plan funding projections
- Cash budget model
- Urban management studies
- Rate of return analysis (capital investment)
- Acquisition and merger analysis
- Personnel scheduling
- Inventory system analysis (simulation)
- Lease or buy analysis
- Distribution system analysis
- Ship scheduling
- Population projection programs
- Enrollment projection models (schools)
- Advertising expense allocation
- Fleet scheduling
- Container movement (simulation)

6.3.5 SUMMARY

In summary, XDS now brings a full complement of software to cover the range of programming, operating, and computational needs of the service industry market. Overall, an increasing number of companies currently using time-sharing, indicate that computer systems with bulk storage capability, large core availability, and faster response are becoming more useful to management. With the additional benefits of increased reliability, improved terminals, and more economical

execution costs, the use of time-sharing for business and engineering problems is firmly established. Finally, smaller companies that have never used computers are realizing some of the business management benefits that their larger competitors have been enjoying for some years.

Now that XDS's software has been further developed to maintain excellent operating efficiency for tape and disk processing, and full compatibility with most used methods of file management, we can offer the following benefits to service bureaus:

- Multiprogramming for maximum system utilization
- Data management input/output control system
- Job stream control for flexible program operation
- Tape and disk utilities
- Sigma APL
- COBOL
- Extended FORTRAN
- A full package of testing and debugging aids
- A batch oriented monitor for local batch services (XOS)
- A time-sharing operating system, utilizing the total hardware capabilities (UTS)

Regardless of the operating system selected, the Sigma 9 can process more jobs concurrently. More users can share the computer's resources which add up to increased productivity from your computer, expanded scope of capability for the customer, and a greater return on investment for the service bureau.

6.4   DATA PROCESSING

Every segment of the American economy today is in some manner affected by data processing, and every type of business is represented as a user. The installed base of all vendors at the beginning of 1970 was $20.0 billion. Shipments during 1971 are forecast to be $7.3 billion, increasing to $13.4 billion in 1976.

The sale and installation of computers in the commercial environment is a challenge to XDS. Unlike XDS's traditional market where computers have served only a few functions, use of computers in the commercial market involves many operations of the establishment. Characteristics of the commercial marketplace for XDS are generally:

- Multi-Use — With user requirements for any combination of batch, time-sharing, on-line processing, and batch scientific processing
- Packaged systems — Utilizing application packages such as the newly acquired brokerage system, DAI-Secure
- Replacement and/or additional processors for expanding 360/30, 360/40, and 360/50 installations, primarily where higher level languages of COBOL and FORTRAN are involved

The market that best fits the capabilities of the Sigma 9 also happens to be the leading area for commercial data processing in the 70's. The Diebold Research Program reports that: "The most characteristic feature of the computer in the next five-year period, beginning in 1970, will be its spread to organizations and into functions that up until the present have not felt its impact". The investment of the 1960's has given many leading companies a head start in using computers in a variety of new, high-payoff applications. The following paragraphs describe some of these high payoff applications and include Diebold projected estimates for the 70's.

6.4.1 FINANCIAL CONTROL AND PLANNING

The first major corporate use of computers was in the financial area, including payroll, billing, and accounting, all largely clerical replacement tasks. Over the next five years, the computer will play an increasing role in the actual forecasting and planning function, providing senior management with alternatives and choices in planning capital investment, money management, allocation of resources, budgeting and business planning, as well as levels of expenditures and alternative acquisition evaluation.

A small number of leading firms have already developed techniques for financial control and planning that will spread widely by 1975. Specific uses include:

- Capital investment

- Money management
- Allocation of resources
- Budget and business planning
- Acquisition evaluation
- Modeling and simulation

The most significant developments making the growth of these applications possible are computer systems which provide on-line access through installed terminals and the computer languages which permit direct use by financial staff members. Display terminals will total 700,000 by 1975, growing from 75,000 in 1970; of these, approximately 11% will be employed by management in retrieving company and market data and in the display of financial results. Special software, in the form of new languages (e.g., BASIC), now exist which permit financial managers to use computers directly without benefit of training as programmers. Other programs, which have created time-sharing systems and data file management and retrieval (e.g., DMS), all combine to facilitate use of computers in these financial applications.

The great power of these techniques is that financial managers are able to compare and select from many alternatives developed and displayed by the computer. For example, in preparing a five-year budget and plan, the return on investment to be achieved under different assumptions of market penetration, cost of money, installed production facilities, engineering expense, etc., are all computed. The manager is then able to bring his own judgement to bear in choosing which are most likely to be the actual situations encountered, thereby selecting the most probable future result.

One of the important stimuli to the increasing use of computers in financial management is the large number of students graduating from universities and business schools who have been trained to use computers in the solution of management problems. As they enter the ranks of industry management, they will rely on the power of the computer to prepare and display alternatives for decision-making.

6.4.2 PRODUCTION AND MANUFACTURING CONTROL

The manufacturing function today is a dynamic corporate activity. To cope with new materials and increased demand for a variety of products, an efficient shop floor requires new machine tools, handling methods, and production controls. What is needed is a real measure of effectiveness — every company and plant has ways of determining how much items cost to produce, but virtually none can determine what they should cost. Rapidly expanding use of computers in this area can be expected by 1975.

Leading industrial advocates of computer-assisted manufacturing now use these systems to operate production machinery. These innovations start with the installation of new computer-controlled production equipment, and include miniature, stand-alone computers that run them. Approximately 5000 small computers will be in use for this function in 1970, growing to 22,000 by 1975. Output from these small computers supports the function of manufacturing control wherein scheduling of facilities and personnel are accomplished. Larger, centralized computers combine and coordinate this data with information concerning required factory output. Data collection is facilitated by the use of terminals located throughout the factory. In 1965, 3100 such terminals were shipped, growing to a projected 38,000 shipped in 1975, for a total estimated installed quantity of 190,000 by that time.

The significance of this use of computers is in the fact that management is investing in new production facilities and techniques in order to gain greater effective control over the function of manufacturing. An integrated system that relates orders received with facilities, personnel, control of machines, raw material, and finished inventories, all leading to shipment, is the goal. As evidence of the trend in this direction, it is estimated that by 1975, 20% of computer time in industry will be allocated to manufacturing management.

6.4.3 PROCUREMENT

Procurement, as differentiated from purchasing, is the overall management of materials, encompassing purchasing, receiving, stores, materials, and inventory control, and (at times) material handling and traffic. Along with marketing, manufacturing, engineering, and finance,

procurement is developing into a major corporate function. Leaders in the aerospace industry now use computer systems for:

- Vendor selection
- Bid evaluation
- Purchase orders
- Material flow control

To do so requires (1) very large data files that describe vendors, specifications, standards, and the results of previous requirements; (2) very large computers; (3) remote terminals; and (4) programming for information retrieval.

These requirements benefit from standardized nomenclature and material descriptions now appearing in many industries. (The steel industry is now actively preparing standard descriptions of iron and steel products, in recognition of the importance of these computer uses.)

6.4.4   NON-INDUSTRIAL USES OF THE COMPUTER

Over the next five years, the computer will become the nerve center of remote terminal systems, being used to solve major problems outside of its more conventional corporate role. Data processing will help to keep managers in close, perhaps hourly, contact with everyday financial and retail transactions on the floors of stock exchanges, banks, and department stores. Similar systems will have an important impact on such service industries as transportation, medicine, and education, as well as government operations.

6.4.4.1   Finance and Banking

The Securities Industry is ready for more extensive application of computers. The stock exchanges have computerized their quotations which are distributed internationally to three service firms who maintain over 16,000 display terminals in brokerage offices. During the next five years, automated machine-readable stock certificates will be introduced, as will on-line processing of trades through the back office. Further, the actual movement of certificates will be largely replaced by automated clearing-house operations supported by computers and terminals.

Banks are moving quickly to forms of automated credit. Nationwide credit approval is now technically feasible through the advent of large scale memories and low cost terminals. Once the cost of checking credit is as low as the cost of a local telephone call, automatic validation of checks and credit will become a reality.

6.4.4.2   Retail

Point-of-sale terminals are a reality in 1970. These bring to the retail establishment many of the management controls developed for industry. The more sophisticated of these systems provide cash control, sales analysis, inventory management, and charge account approval — all at point of sale.

A future extension planned for these systems includes displays of merchandise available for delivery but not on view at the store. This will also facilitate the placing of orders made by telephone, as order clerks will have ready knowledge of merchandise available for delivery.

6.4.5   SIGMA 9 OPERATING SYSTEMS

The commercial market use of computers is growing rapidly in the areas where XDS and Sigma 9, and its operating systems, have specific strengths and can offer significant advantages over competition. Importantly, one of the requirements of this market is the maintenance and manipulation of large, highly dynamic data bases. Sigma 9 permits on-line maintenance and sophisticated data base manipulation. By taking advantage of the capability to place data on line for direct inquiry and updating, XDS can satisfy business management's desire to have <u>current</u> operating information available upon which decisions can be based.

6.4.5.1   UTS

Most commercial computing facilities experience wide fluctuations in workload conditions. Through the facilities available with UTS, installation management can dynamically "tune" the system during operation, to adjust for varying conditions of usage. Operating personnel may quickly and easily alternate operations between full batch usage and concurrent batch/on-line

usage, through the use of the UTS terminal startup/shutdown facility. By analyzing system utilization, installation management may allocate resources to the areas where they are most needed and in the desired proportions. For example, in a manufacturing environment, it may become necessary to restrict the on-line users and allocate the total system to process the workload during an accounting cycle.

Businesses increasingly rely upon the output from computers in daily operations; therefore, it is very important to keep the system up and operating at its maximum. UTS's diagnostic and automatic recoverability provisions, as well as the ability to re-allocate resources for a degraded mode of operation, are outstanding attributes of the system and should be fully understood by the commercial user.

The commercial environment, generally, is experienced in the use of computers. There is a strong undercurrent, however, which demands more sophisticated hardware and software. Sigma 9 with UTS, a sophisticated time-sharing, batch oriented system, provides the commercial marketplace with:

- Multi-use operations
- Dynamic allocation of computing resources
- Continued operation in degraded mode when portion of system is malfunctioning
- Diagnostic and recovery capability covering hardware and software contingencies
- Simultaneous memory access by multiple I/O devices and CPU
- Availability of system to larger number of concurrent on-line time-sharing users

6.4.5.2   XOS

The commercial marketplace is characterized as one requiring concurrent handling of batch, remote batch, and data base manipulation. Occasionally, a real time requirement will be encountered. The outstanding facilities available in XOS can be used to enhance the capability of Sigma 9 to fulfill the multi-use requirements of this market.

XOS has powerful multiprogramming performance through its job management/resource allocation capabilities. Through its use, maximum CPU utilization is assured. This is particularly advantageous to the commercial user because of the heavy workload and diverse applications. In addition, in commercial data processing, the requirement exists for the immediate rescheduling of jobs. Through intervention at the operator's console, a job with a lower priority can begin immediate execution. This is an advantage since, with this facility, the user alleviates the "crises" situations which occur almost daily in this environment.

With XOS, an unlimited number of parallel jobs, five operation jobs and one serial job, can be in core at any one time. This makes the use of Sigma 9 and XOS an outstanding price/performer and permits the commercial user to fulfill his TOTAL multi-use requirements.

6.4.6   XDS COMMERCIAL SOFTWARE

The commercial user requires a variety of vendor-supplied software as the entry fee for getting into data processing. XDS is in an excellent competitive position with several higher level languages (COBOL, FORTRAN, BASIC, and MANAGE) to fill this need, as well as FMS and DMS, described below.

6.4.6.1   File Management System (FMS)

The characteristics required for commercial processing include:

- Space management
- Error recovery
- Loading and unloading
- Index maintenance
- Physical reads and writes
- File organization capabilities including:
    - Sequential — all references to a file start at the beginning and each logical reference provides the next item
    - Keyed Access — all references to the file are through a user supplied key, with maintenance of keys performed by FMS

6-29

– Random Access — all references to the file are made by the user submitting a direct

address that he has calculated or a relative key that FMS translates to an address

These file management capabilities are currently available and will be enhanced even further.

6.4.6.2 Data Management System (DMS)

A vendor who can provide a well designed data management system becomes a serious contender in the commercial market. XDS is in that position with DMS-1. DMS-1 permits the user to define records and their relationship to other records. Access is accomplished through pointers called chains. DMS-1 utilizes FMS to place and retrieve records. The primary features of DMS-1 are to:

- Eliminate redundancy of records by chaining together related files
- Provide file organization facilities, eliminating them from the user program
- Provide the user with data in any required sequence through the use of logical chains without physically repeating or re-ordering the data

6.4.7 TRENDS IN COMMERCIAL BUSINESS DATA PROCESSING

Initially, computers were utilized in commercial businesses solely in a batch environment. As the user gained experience and hardware/software improved, it became apparent that there was a need for a multi-use capability. Today, there is visible evidence that the trend is in the direction of multi-use and that it involves significant percentages in various industries, as shown by the chart on the following page.

6.4.8 COMMERCIAL ACCOUNT PENETRATION

The commercial prospect/customer views the computer salesman as a professional problem solver, consultant, and educator. Many current users in this market state that vendor representatives oftentimes know more about the overall details of the user's business than do the employees of his business. This is because the justification for using computers in this environment usually dictates that more than one application area be mechanized; therefore, the knowledge required for successful installation is wider in scope.

MULTI-USE

Percent of Business-Oriented Installations that Perform Business and Scientific or Real Time Processing

Category	Value
Agriculture	
Mining	29
Construction	22
Fab. Production	26
Fab. Process	7
Fab. Assembly	14
Railroad	10
Transportation	4
Air Line	30
Comm & Utilities	17
Wholesale	2
Retail	2
Banking	
Credit	
Stock Market	
Insurance	3
Other Finance	2
Serv-Consumer	20
Serv-Medical	15
Serv-Education	36
Fed Government	25
Other Government	11
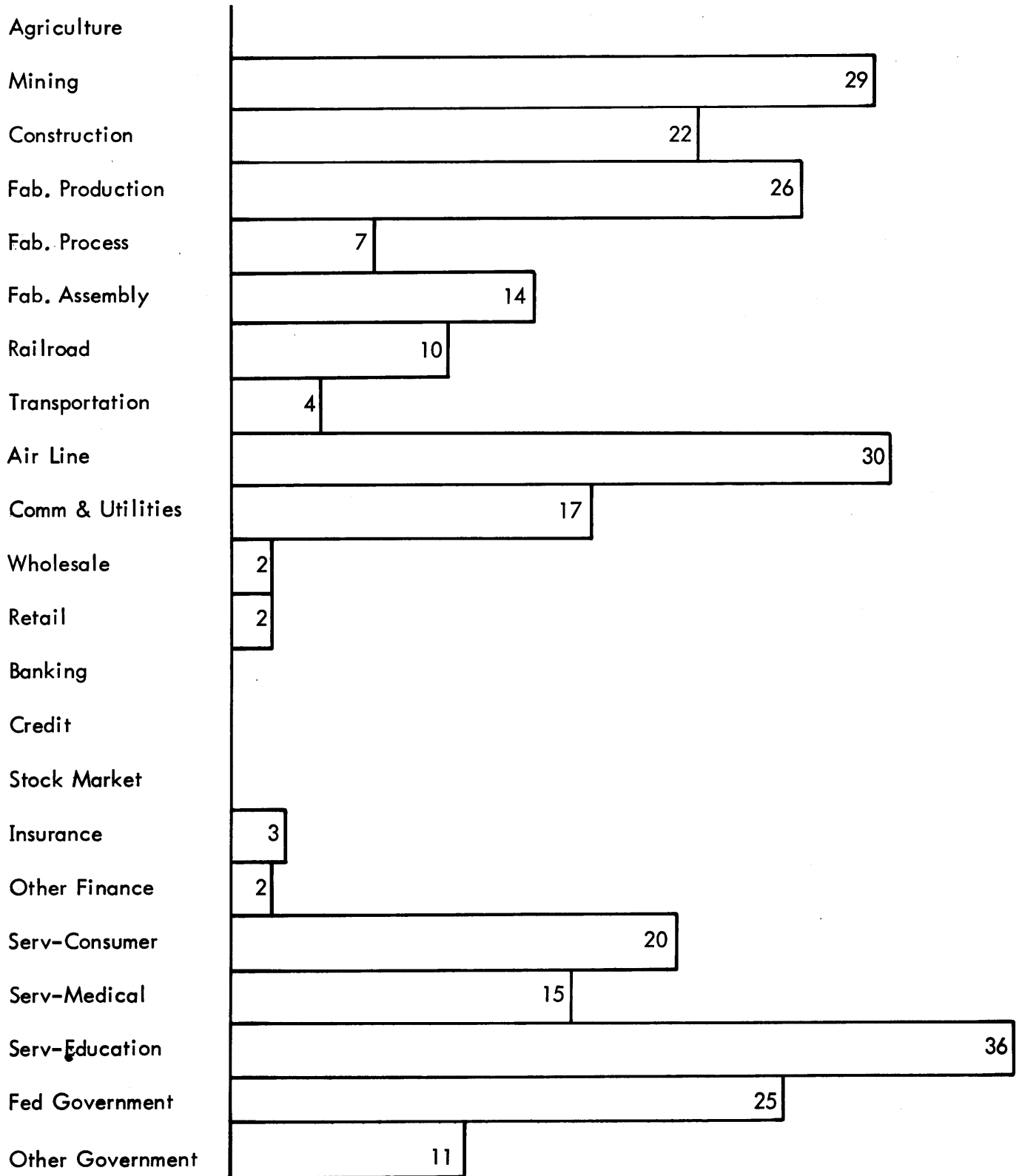
The purpose of the information which follows is to outline the path to be followed before a commercial account can be successfully penetrated. Other details should become apparent as a particular opportunity is investigated.

- SELECT THE PROSPECT
- OBTAIN EXECUTIVE AND DATA PROCESSING MANAGEMENT NAMES
- OBTAIN COMPANY PRODUCT AND FINANCIAL INFORMATION
- OUTLINE KEY QUESTIONS TO ASK EXECUTIVES
- PRE-QUALIFY
- REQUEST STAFF INTRODUCTIONS
- OPEN THE DOOR FOR FUTURE EXECUTIVE LEVEL CALLS
- OBTAIN THE RFP (IF ANY)
- COMPLETE THE ACCOUNT PROFILE
- DETERMINE CUSTOMER PRIORITIES AND REQUIREMENTS
- DEVELOP POSSIBLE SOLUTIONS AND MANAGEMENT ALTERNATIVES
- DETERMINE CONVERSION REQUIREMENTS
- UTILIZE XDS "EXPERT" SUPPORT
- SELECT HARDWARE AND SOFTWARE
- SCHEDULE ON-GOING EXECUTIVE CALLS
- PREPARE A PROPOSAL
- CONDUCT PRE-ORDER SEMINARS
- COORDINATE BENCHMARKS
- PREPARE A PRELIMINARY SITE PLAN
- DETERMINE EDUCATION REQUIREMENTS
- PRE-SELL THE PROPOSAL AT ALL LEVELS
- CONDUCT EXECUTIVE PRESENTATIONS
- PERFORM CONTINUOUS FOLLOW-UP
- PROTECT YOUR ORDER
- OBTAIN THE ORDER
- CONTROL THE ACCOUNT

As can be seen from the preceeding, many details must be known and attended to if one is to be successful in the commercial market. Because of the wide-ranging knowledge required, it is critical that XDS representatives familiarize themselves with the details necessary for true account penetration.

No longer is it necessary for XDS salesmen to "reinvent the wheel" in designing the forms, etc., to document an account profile. AMMO (Account Management and Marketing Organizer — document number 67-11-06) is available for this purpose. It is an invaluable aid in identifying the information necessary to understanding, closing, and retaining control of an account.

6.4.9 SUMMARY

Forecasts indicate that during the next five years computer usage will spread rapidly to companies and into functions of organizations that have previously not benefitted from their use. Further, there will be a more pronounced transition from the typical batch mode of operation to one of greater sophistication.

As the market evolves, there will be increasing demands for hardware/software combinations capable of supporting concurrent batch, remote batch, real time operations, and remote inquiry/ update of dynamic data bases. XDS is in a strong position to capture a large segment of this business — we have credibility in the multi-use environment and the price/performance ratio of Sigma 9, coupled with the outstanding facilities available in UTS and XOS, enhance our capability to capture a significant portion of the commercial market.

SECTION VII
PRESENTATION MATERIAL

7.0 INTRODUCTION

The presentation aids that have been produced to assist the Sales Force in selling Sigma 9 are standard 2-color flip-charts (8-1/2 x 11 and 17 x 22 sizes). These charts have been shipped to the field along with the additional announcement materials and should meet the needs of most sales situations. However, for large audiences, where the 17 x 22 charts are too small, 35mm slides of these charts are available on a loan basis from Presentation Services, C6-30 (Bill Driedger, X4379, or Carl Boehm, X4380).

7.1 SIGMA 9 MATERIAL

Four presentations have been prepared for Sigma 9 and presentation guides for them are included in this notebook to aid in the familiarization of Sales personnel with Sigma 9 in training at the home office and, subsequently, in the field. Each presentation guide is a discussion outline with black-and-white representations of the related flip charts and are contained in the following pages:

- Sigma 9 Management Presentation – pages MGMT-1 through MGMT-10
- Sigma 9 Hardware – pages HDW-1 through HDW-21
- XOS – pages XOS-1 through XOS-21
- UTS – pages UTS-1 through UTS-33

7.2   EXISTING APPLICABLE MATERIAL

Other presentations, previously prepared and distributed to the field – and applicable to Sigma 9, are:

- FMPS, 67-04-XX
- SL-1, 67-04-XX
- 1400 Simulator, 67-04-XX
- SORT/MERGE, 67-04-XX
- MANAGE, 67-04-01
- RBM 5/7, 67-04-24
- Graphics, 67-04-25
- BTM, 67-04-26
- HEARTS, 67-04-35
- TS-400, 67-04-39
- XDS Motion Picture (check your District Office)

INTRODUCTION

This Sigma 9 Management presentation is an overview of the hardware, XOS, UTS, and supporting software. It can be used as:

- A stand-alone pitch to management

- Part of a total capability presentation

- An introduction to a more detailed Sigma 9 presentation

- Support of a Sigma 7 presentation

You can customize your presentation by selectively integrating charts from Sigma 9 Management, Hardware, XOS, and UTS presentations.

Charts 1, 2, and 9 of this presentation are not exclusively Sigma 9 oriented and can be used to good effect in any presentation where you want to show product relationships.

CHART #1

The title chart can be used to introduce your talk in a number of ways:

- A history of our company and products
- Highlight key product features
- Discuss our leadership in real time, time-sharing, etc.

SIGMA 9		
SIGMA 6		SIGMA 7
SIGMA 3		SIGMA 5
940		SIGMA 2
930	9300	92
910	920	925
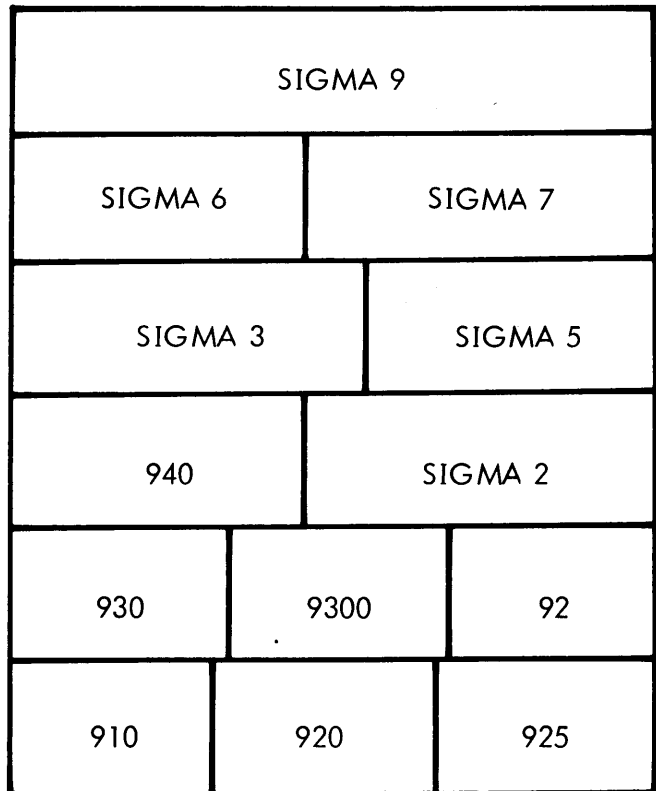
NOTES

CHART #2

(The inverted triangle happens to be the inter-
national traffic sign for a major street, with
free-flowing traffic)

The top two are universal to the Sigma family.
The last one is an additional capability pro-
vided by Sigma 9 with the new Xerox Operat-
ing System.

- Multi-Access (to memory)

  - Removes bottleneck
  - Increases throughput
  - Permits effective multi-use

- Multi-Use

  (Discuss in terms of the customer's
  applications)

- Multiprogramming

  - Multiple jobs are in memory simul-
    taneously
  - Effective use of resources
  - Faster job turnaround
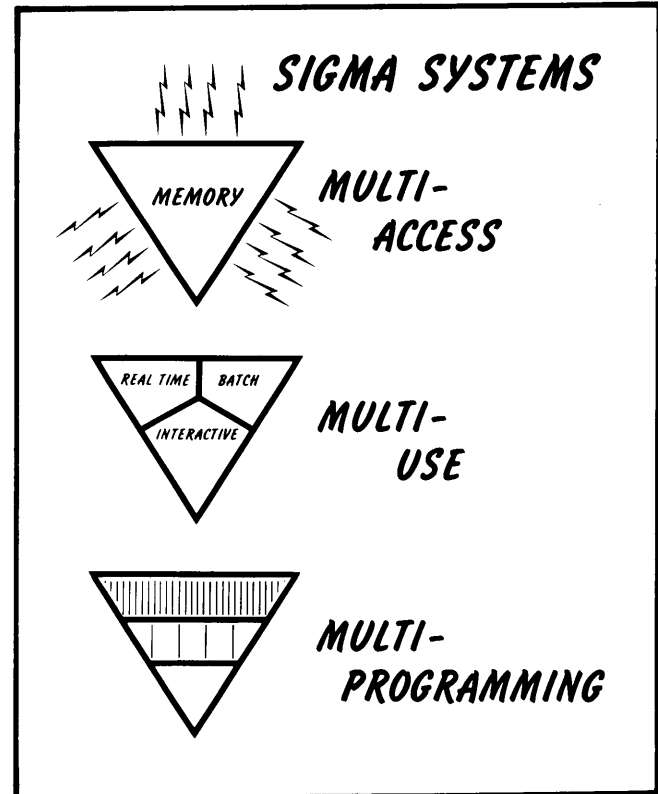
NOTES

CHART #3

In addition to throughput, multi-access provides flexibility for continued expansion and addition of new applications:

- Simultaneous, independent computing and input/output

- Independent communications. Can handle new, faster communications gear as it is made available, and high speed data acquisition

- Special devices — A/D converters, custom-designed IOP's, other manufacturer's equipment, etc.

- Multiprocessor systems (as technology advances), computer-to-computer communication and memory sharing

SIGMA 9 MULTI-ACCESS

OBJECTIVE: CAPACITY FOR THROUGHPUT AND GROWTH

12-WAY ACCESS TO MULTI-MILLION BYTE MEMORY

- Computing
- Input-Output
- Communications
- Special Devices
- Multiprocessor
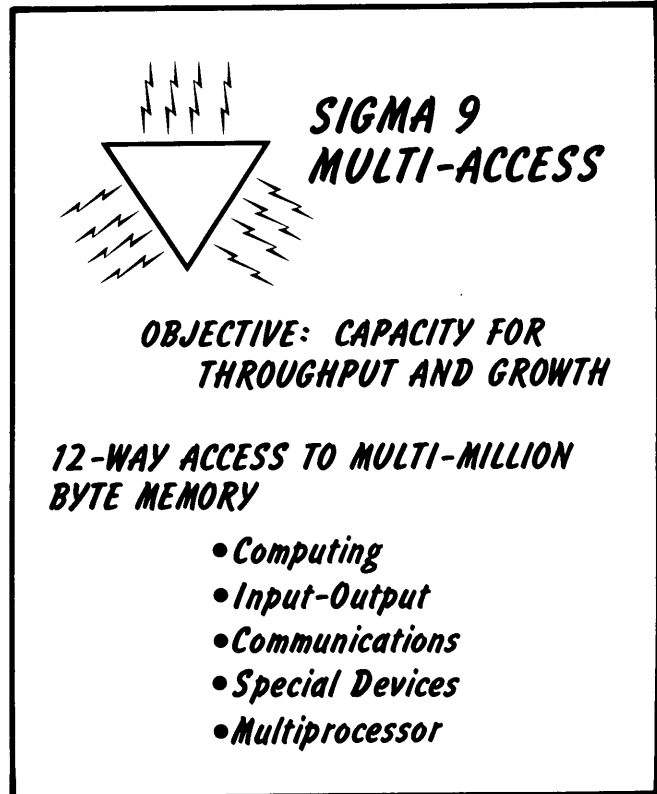
NOTES

CHART #4

Sigma 9 is adaptable to production needs and priorities.

Multi-use is built into architecture.

Selective optimization of services with five powerful operating systems, compatible with Sigma family.

Here is where you explain why we have two big operating systems and how they differ:

- XOS — A high throughput multiprogram-ming system with real time and time-sharing

- UTS — A highly interactive time-sharing system with batch and real time

SIGMA 9
MULTI-USE

OBJECTIVE: *ADAPT TO OPERATIONAL NEEDS*

Multi-Use with

Optimized Throughput of Volume Jobs

OR

Optimized Response to Interactive Users
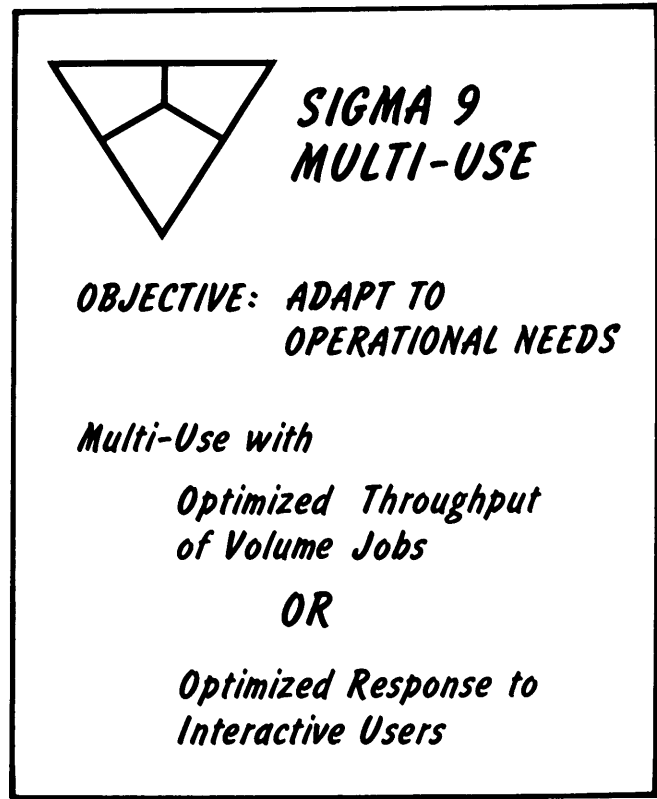
NOTES

CHART #5

XOS permits a very large number of jobs to be active in memory simultaneously. This enables the system to use its resources efficiently. Any parts of the system that are not being productively used by one job are matched with other jobs that, at the moment, can be processed with the available resources.

- Multiple Parallel Jobs (highest priority)

  The number of these active jobs are limited only by memory size. These are <u>typically</u> one-step operations such as the moving of information within the system, changing of priorities, or starting a real time program

- Five classes of production jobs. One job of each class is in memory and others are queued

- One lower priority (serial) stack

  Jobs that can be done whenever resources become available. Can be moved into higher priority if they don't get done. Priority of jobs can be changed by the operator from his console

- Jobs can be processed in local or remote batch modes. Their priority assignment is independent of mode of entry

- (Check on availability of time-sharing at time of presentation)

XOS MULTIPROGRAMMING

OBJECTIVE: MULTI-USE WITH HIGH VOLUME THROUGHPUT

Multiple Parallel Jobs

+5 *Prioritized Jobs with Related Queues*

+1 *Low Priority Job with Related Queue*

LOCAL + REMOTE WITH INTERACTIVE CAPABILITY
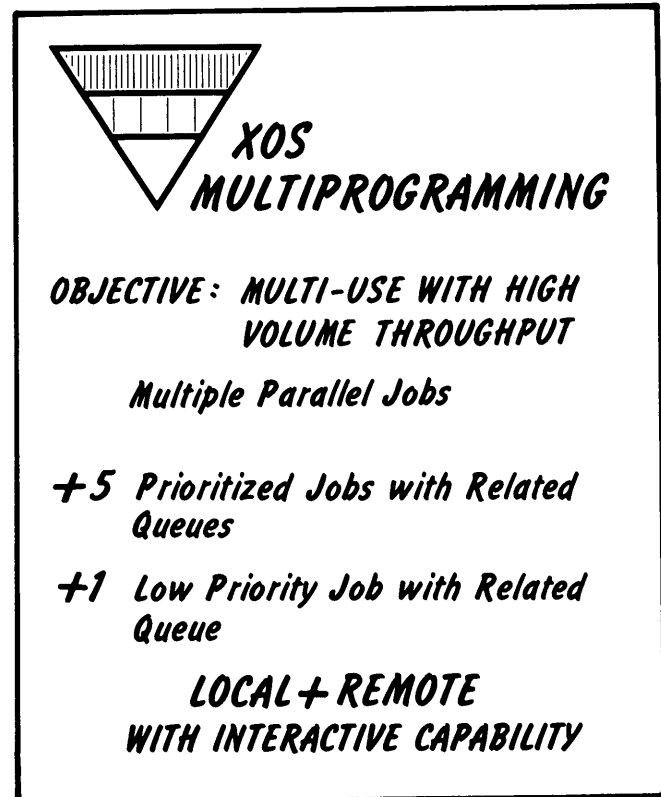
NOTES

CHART #6

UTS can perform in all the modes simultane-
ously, but it is optimized to be highly responsive
in a time-sharing environment.

- Scheduled jobs - batch jobs

- Unscheduled jobs - high priority batch
  jobs pushed ahead of normally scheduled
  jobs

- Interactive processing - conversational
  time-sharing

- Real time

(The above terms were used to get away from
common buzz-words)

**UNIVERSAL
TIME SHARING
SYSTEM**

**OBJECTIVE: MULTI-USE WITH
      OPTIMIZED RESPONSE TO
      INTERACTIVE USERS**

- **Scheduled Jobs**

  - **Unscheduled Jobs**

    - **Interactive Processing**

      - **Real- Time Control**
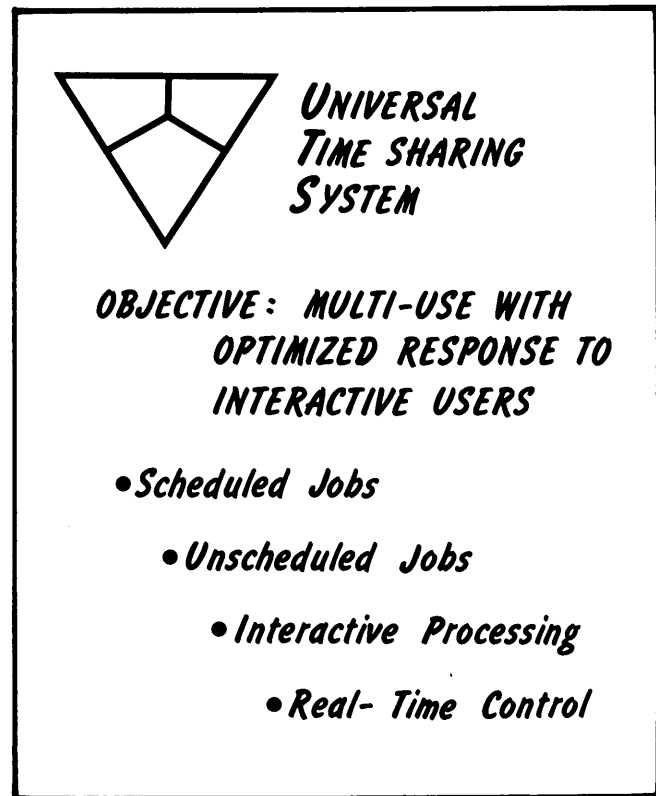
NOTES

CHART #7

The productivity of a system is greatly depend-
ent on the way the software operates.

- Multi-mode: Real time, batch, time-
sharing language compatibility, UTS
(batch/conversational)

- Reentrant processors (UTS): time and stor-
age conservation

- Diagnostics - to speed up programming

- Resource allocation - to assure throughput
and responsiveness

- Accounting - to allocate and charge
computer operations

- Memory management - to keep down con-
figuration costs and maintain throughput

- System tuning - to dynamically meet
changing priorities

<blockquote>

SIGMA SYSTEM RESOURCES

**OBJECTIVE: PRODUCTIVE
                       UTILIZATION**

SOFTWARE AND PROCESSORS
> *Multi-Mode*
> *Re-Entrant*
> *Diagnostic*

SYSTEM MANAGEMENT
> *Resource Allocation*
> *Accounting*
> *Memory Management*
> *System Tuning*

</blockquote>

NOTES

CHART #8

In addition to providing throughput, the system must keep going, if the jobs are to be completed:

- Degradation or failure can be diagnosed while the system is operational

- If memory fails, that module can be bypassed and serviced

- If a problem is suspected, it can often be isolated from <u>remote locations</u> in several ways:

    - A diagnostician on a time-sharing terminal can perform a diagnosis while the system continues operation
    - The total system can be exercised and diagnosed by specialists from remote locations

- Most of the Sigma 9 hardware and software have been proven under field conditions in Sigma 5/6/7

RELIABILITY
+
MAINTAINABILITY

OBJECTIVE: UPTIME

Dynamic Diagnostics

Memory Reconfiguration

On-Line Diagnostics

+

Field-Proven Components

NOTES

CHART #9

This chart shows how Sigma systems are designed for both vertical and horizontal growth, through the entire family. Discuss this around the prospect's specific future needs.

The top diagram relates to:

- RBM - Sigma 3 - 9, except Sigma 6

- BPM - Sigma 5 - 9, except Sigma 6 (no real time)

- XOS - Sigma 7 and 9

The second diagram relates to:

- BTM - Sigma 5 - 9

- UTS - Sigma 6 - 9

- XOS - Sigma 6, 7, 9

The lower diagram relates to:

- BTM - Sigma 5, 7, 9

- UTS - Sigma 7 and 9

- XOS - Sigma 7 and 9
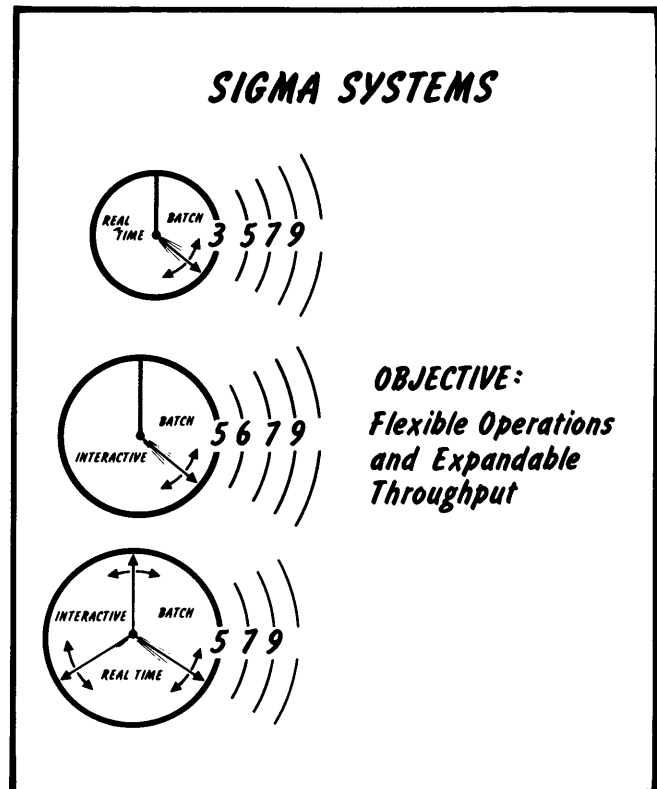
NOTES

CHART #1   Originated by: Marketing Technology

INITIAL BENEFITS STATEMENT — "What you want, we've got."

General Benefit — "What You Want"

Rapidly increasing demands on data processing requires that today's computing systems...

- Provide high throughput, short response
- Adapt to multiple-use environments
- Expand easily to meet future needs

At the same time, economic considerations demand that your system...

- Make maximum use of existing software
- Provide the highest possible availability
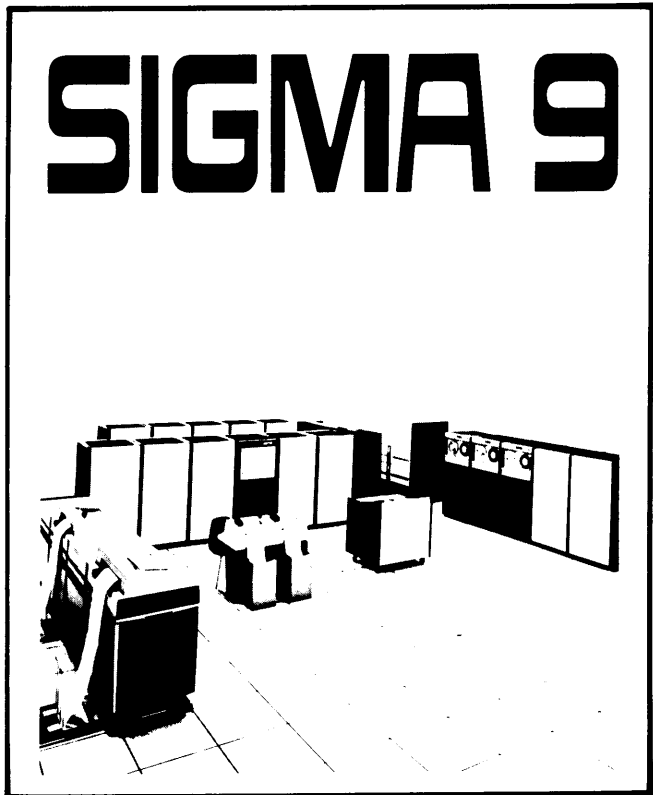
SIGMA 9

NOTES

CHART #2   Originated by: Marketing Technology

Product Features — "We've Got"

To meet the data processing demands of today, the Sigma 9 incorporates the advanced features of...

- Proven Sigma memory-centered architecture permitting simultaneous program execution and I/O processing
- Operating characteristics that permit efficient functioning in general purpose, multi-processing, time-sharing, and real time environments
- Expandability in all dimensions — memory, input/output processors, I/O devices — to meet individual user requirements

Superiority of the Sigma 9 as a price-performer is assured by...

- Total compatability with other Sigma computers; all programs written for Sigma 7 will run on equivalent configurations of Sigma 9
- Unique reliability and maintainability features integrated into the basic design of the system

- **Advanced Design**
- **Field Proven**
- **Application Flexibility**
- **Multi Dimensional Growth**
- **Sigma Compatibility**
- **Maximum Availability**

NOTES

CHART #3   Originated by: Marketing Technology

NOTES

INDEPENDENT ORGANIZATION
Simultaneous Processing

MEMORY ORIENTED
High I/O Throughput
Maximum CPU Utilization
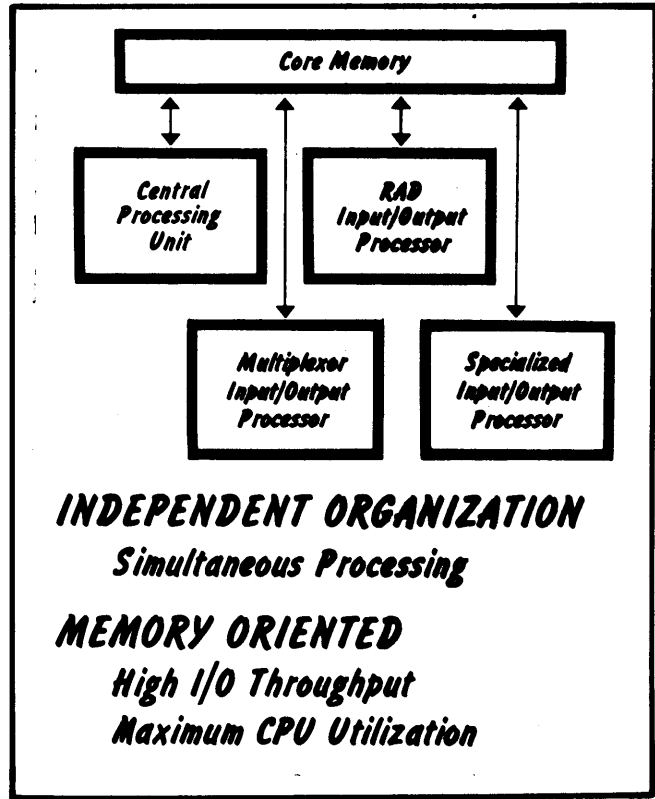
INTRODUCTION — System Organization

Features —

- Extension of Sigma family memory-centered architecture

- Independent central and I/O processors

- I/O processors have independent access to memory

- I/O processing proceeds on non cycle-stealing basis

- Memory expandable from 512K bytes to 2048K bytes. Up to 12 ports. Up to 11 I/O processors

Benefits —

No risk, field-proven, successful

Simultaneous program execution and input/output processing

High I/O volume with minimum CPU interference

Maximum CPU utilization

Permits multi-dimensional growth capability to meet unique requirements

CHART #4

NOTES

MEMORY

2048K BYTES
512K WORDS

ASYCHRONOUS TIMING

FIELD EXPANDABLE

MEMORY INTERLEAVING

12-WAY ACCESS

32-BIT WORDS
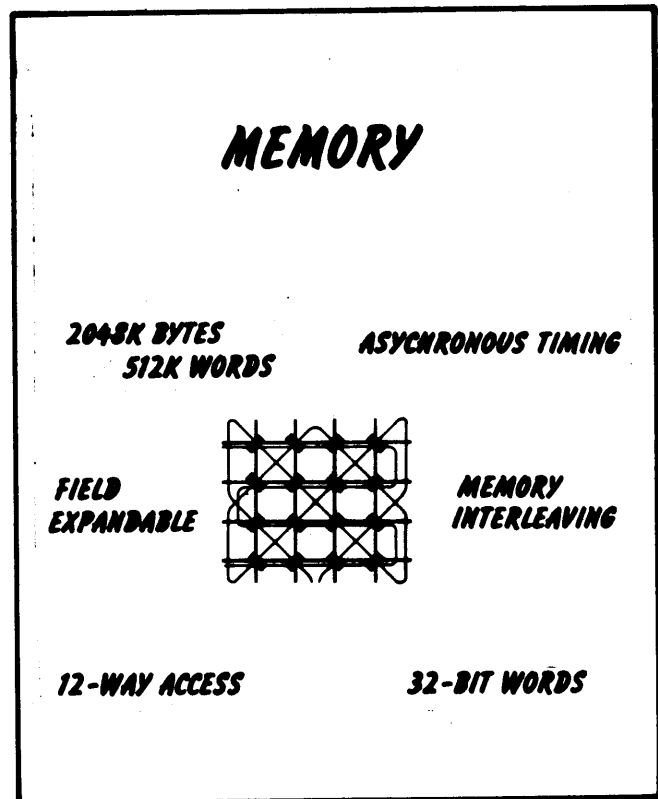
MEMORY — Core

Sigma 9 memory is organized into BANKS and UNITS. A bank may be 64K bytes (16K words). A unit (cabinet) is always 128K bytes (32K words).

Features —

- Word (32-bit plus sign) oriented; address-able on byte (8-bit), half-word (2-byte), word (4-byte), and doubleword (8-byte) boundaries

- Field expandable from 512K bytes (128K words) to 2048K bytes (512K words): 128K byte increment from 512K byte to 1024K-byte; 256K-byte increment from 1024K byte to 2048K byte

- Ports can be added singly up to limit of 12 ports.

Benefits —

Eliminates programming overhead in CPU op-erations involving multiple data boundaries

Meets any future growth requirements
Convenient size increments easy to add

Permits up to 11 I/O processors

CHART #4 (Cont'd)   Originated by: Marketing Technology

Features —   Benefits —

- Direct addressing of entire memory (2048K   No base register required. Fast, direct access
  bytes)

- Two-way interleaving between two   Reduces effective memory cycle time
  64K byte banks and four-way interleaving
  between two 128K byte units

CHART #5   Originated by: Marketing Technology

NOTES

Rapid Access Data Storage Device

- *Minimum Latency*
- *High Transfer Rate*
- *Logical Extension of Core*
- *Selection of Capacities and Transfer Rates*
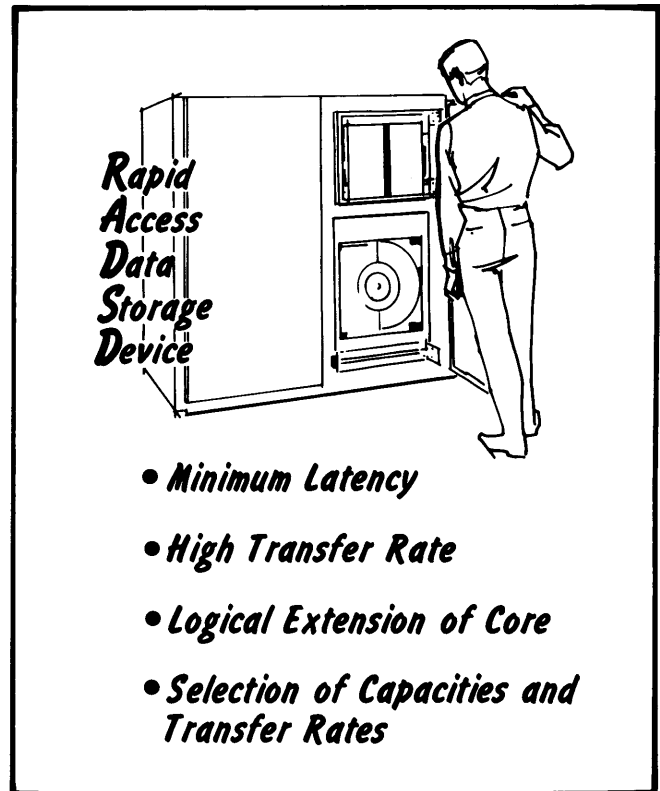
MEMORY — RAD

The Rapid Access Data disk is a logical exten-
sion to core.

Features —

- Fixed head per track disk

- Capacities to 6.2M bytes per unit

- Transfer rates to 3M bytes per second

Benefits —

Minimum latency (average hardware access
time of only 17 milliseconds)

Large storage, low cost mass memory

With overlaying techniques core can be kept to
a minimum

CHART #6   Originated by: Marketing Technology

CPU GENERAL FEATURES

The data processing world is becoming one which requires multiprogrammed multi-application capability.

The Sigma 9 fits this multi-use requirement.

General Purpose (Batch) Features —

- Floating point hardware for scientific computation speed and significance
- Decimal hardware for business computation speed and simplicity
- Displacement indexing for more efficient use of index registers

Time-Sharing Features —

- Multiple register blocks and hardware stack manipulation for rapid context switching
- Multiple levels of system protection for user and operating system security
- Hardware storage management for maximum utility of memory

Real-Time Features —

- Most advanced interrupt system available
- Rapid context switching
- Three methods of I/O, including direct to a general purpose register
- Variable precision arithmetic – adaptable to the precision of the data

Multiprogramming/Multi-Usage Features —

- Multiple access to memory for more efficient use of system resources
- Independent and simultaneous I/O operations allow processor to process
- Hardware memory mapping allows many users to be resident in core and protected
- Interrupt system permits quick response
- Rapid context switching increases throughput

MULTI - USE

On-Line Time Sharing

Batch
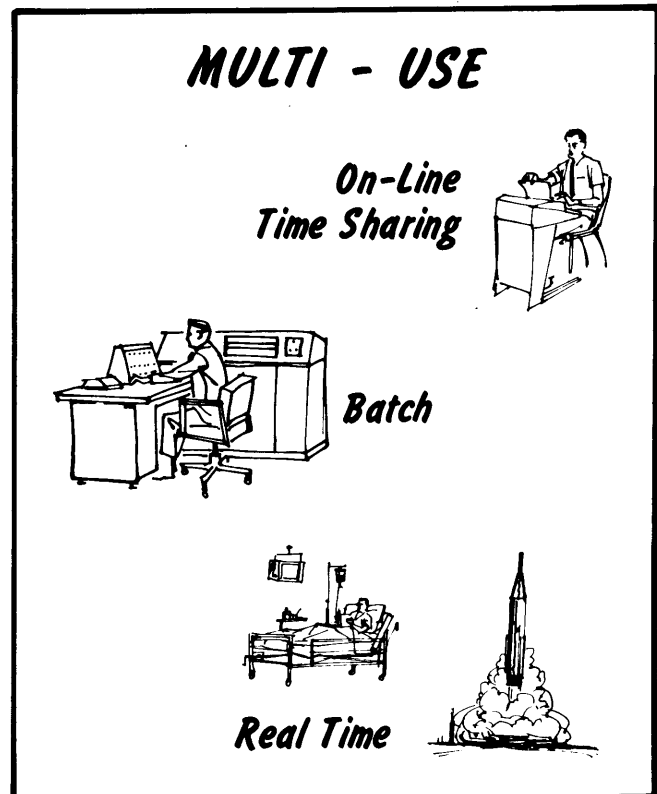
Real Time

NOTES

CHART #7   Originated by: Marketing Technology

MEMORY WRITE PROTECTION

Operating system and resident real time programs must be guaranteed security.

Memory write protect is a set of 256 two-bit lock registers which guard the first 512K bytes (128K words) from accidental destruction by other programs.

Each 2-bit register protects a 2048-byte (512-word) page of this first part of core. A user must have the proper write key in his PSD to be permitted to write on that page.

The key can only be changed by a privileged instruction.

Privileged instructions (XPSD, I/O, control register alterations) can only be executed by master mode programs — operating system and fully debugged resident real time programs (user option).
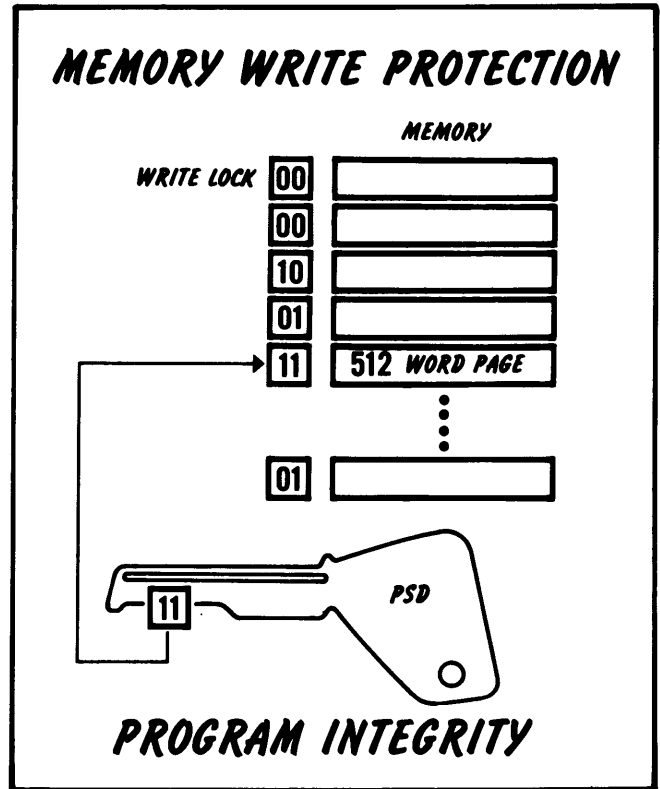
NOTES

CHART #8　　　　　　　　Originated by:　Marketing Technology

MEMORY MAP

Multiprogramming and multi-use applications
can require large amounts of core memory.

The Sigma 9 can have as much as 2 million
bytes (512K words) of core memory. To main-
tain compatability with Sigma 5 and Sigma 7,
memory is divided such that each user sees up
to 512K bytes (128K words). This is accom-
plished using hardware mapping registers iden-
tical in function with those in Sigma 7 but ex-
tended in size.

The 512K bytes per user is not necessarily con-
tiguous. The memory map distributes the user's
program among the available core pages. The
storage looks contiguous to the user.

Mapping is done in hardware which minimizes
overhead time. Increased throughput due to
better use of core more than makes up for the
little amount of overhead.

MEMORY MAP

Efficient Core Utilization
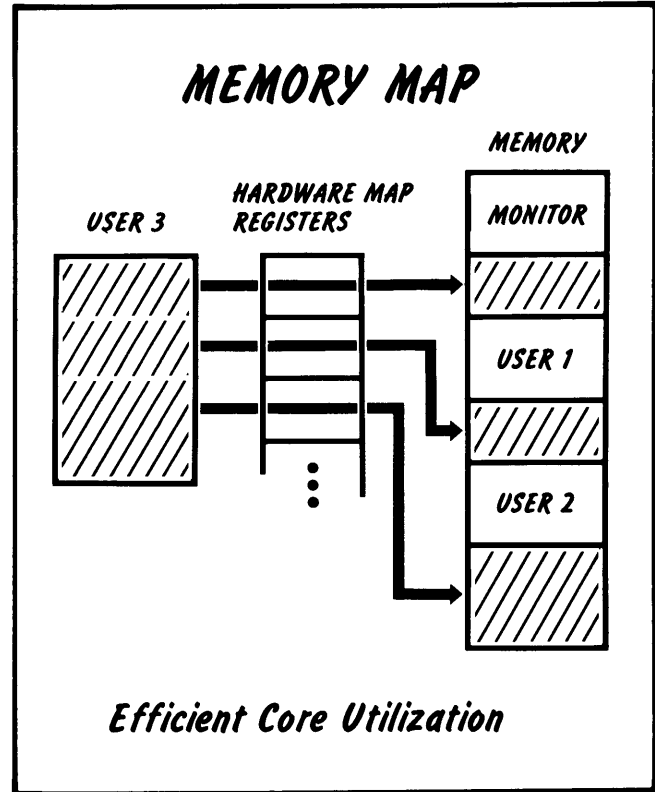
NOTES

CHART #9   Originated by: Marketing Technology

ACCESS PROTECTION

As part of the memory map, a further level of
system protection is included. A separate set
of registers protects the access to each 2048-
byte (512-word) page of memory.

On Sigma 7, this concept only applied to slave
mode programs. For Sigma 9 the concept is ex-
tended to master mode programs. Now, all of
memory is under the control of access protection
while operating in the mapping mode.

Master mode programs perform I/O operations.
Every program is now protected from every other
program even during I/O operations.

On Sigma 9, a new mode called the "Master
Protected Mode" has been added which allows
privileged instructions such as I/O instructions
to be executed while, at the same time, access
protection is enforced.

MEMORY MAP ACCESS PROTECTION

PRIVATE   DO NOT DESTROY

Access Protection For:

    Master Protected Mode Programs

    Slave Mode Programs

Access Codes:

    00 = Read, Write, Execute

    01 = Read or Execute

    10 = Read Only

    11 = No Access

NOTES

CHART #10   Originated by: Marketing Technology

CPU LOOK-AHEAD

Faster execution time means increased through-put, but faster core memory is prohibitively expensive.

"Look Ahead" fetches and prepares instructions for execution while other instructions are being executed. The result is a significant decrease in job execution time.

With reference to the chart, i represents the current instruction; i+1, the next instruction; and so on. Those instruction phases shown in each horizontal row of boxes are occurring at the same time.

NOTE: Fetch = Instruction Fetch
        Decode = Set OP code and effective
                   operand address
        Execute = Fetch operand and execute

CPU 'LOOK AHEAD'

INSTRUCTIONS

TIME	i	i + 1	i + 2
t_0	Fetch		
t_1	Decode	Fetch	
t_2	Execute	Decode	Fetch
t_3		Execute	Decode
			Execute

Overlapped Operations
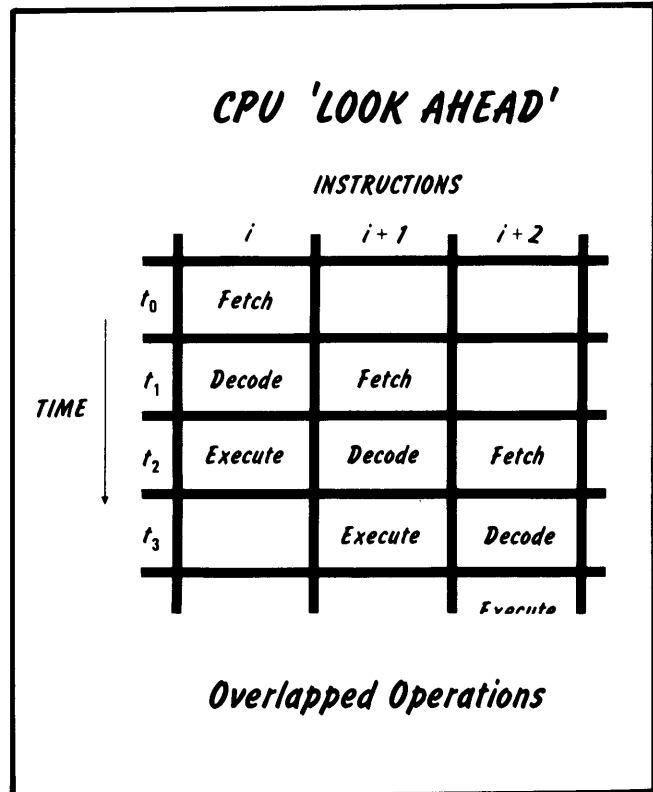
NOTES

CHART #11   Originated by: Marketing Technology

CPU SUMMARY

Instruction Set —

- Extension of Sigma 7 instruction set
- Sigma 7 instructions expanded for flexibility (e.g., RD and WD instructions)
- New instructions enhance maintainability
- New instructions expand memory addressing

Expanded Addressing Capability —

- More programs in core simultaneously
- Larger programs without overlays
- Larger I/O buffers

Rapid Context Switching —

- Program Status Doubleword
- Four register blocks
- Push-Pull instructions

Security —

- Power fail-safe
- Memory protection
- Access protection

Adaptable —

- Double precision floating point
- Decimal arithmetic
- Interrupts

CENTRAL PROCESSING UNIT

16 Million Byte Addressing Capability

4 General Register Blocks

Power Fail - Safe

Double Precision Floating Point Hardware

Decimal Instruction Set

224 External Interrupts

NOTES

CHART #12   Originated by: Marketing Technology

NOTES

SIGMA ARCHITECTURE

Memory   Memory   Memory   Memory

Central Processing Unit (CPU)

Specialized Input/Output Processor

Multiplexor Input/Output Processor (MIOP)

RAD Input/Output Processor

I/O   I/O

Up to 11 Independent IOP's Per System
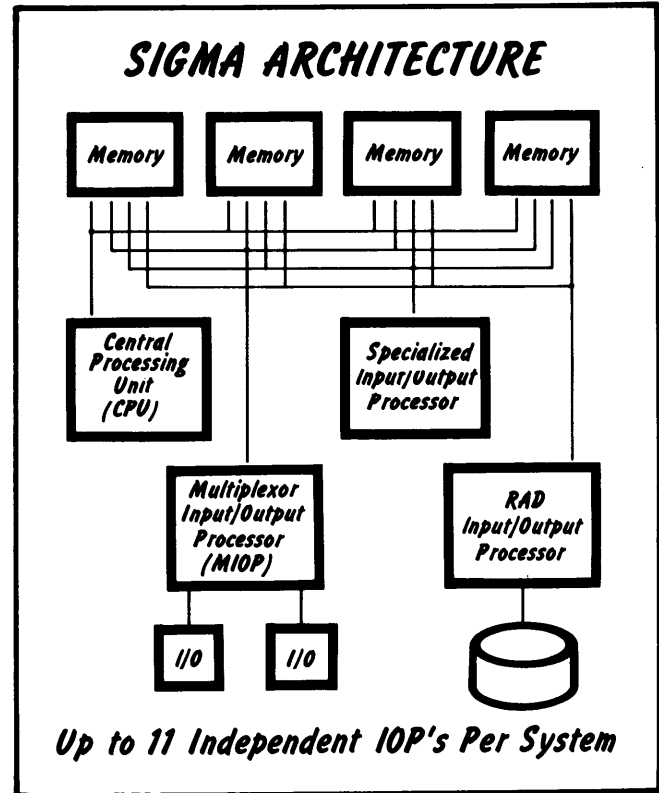
INPUT/OUTPUT — Introduction (Review Architecture)

Features —

Benefits —

- Many I/O processors (up to eleven) limited only by the number of ports, operate asynchronously from and simultaneously with the CPU

  Simultaneous program execution and I/O processing
  Maximum CPU utilization

- Direct I/O of a full word without use of a channel

  Provides direct I/O to CPU general registers

- Direct memory access

  Adapts system to user's unique device interface

- Both data and command chaining

  Improved gather-read, scatter-write operations

CHART #13   Originated by: Marketing Technology

NOTES

MULTIPLEXOR IOP

| CPU |
| MEMORY ...... MEMORY |
| MIOP — Channel A | Channel B |
| I/O I/O ···· I/O I/O |
| 8-24 Subchannels   8 Subchannels |

Memory-to-Memory Move Feature
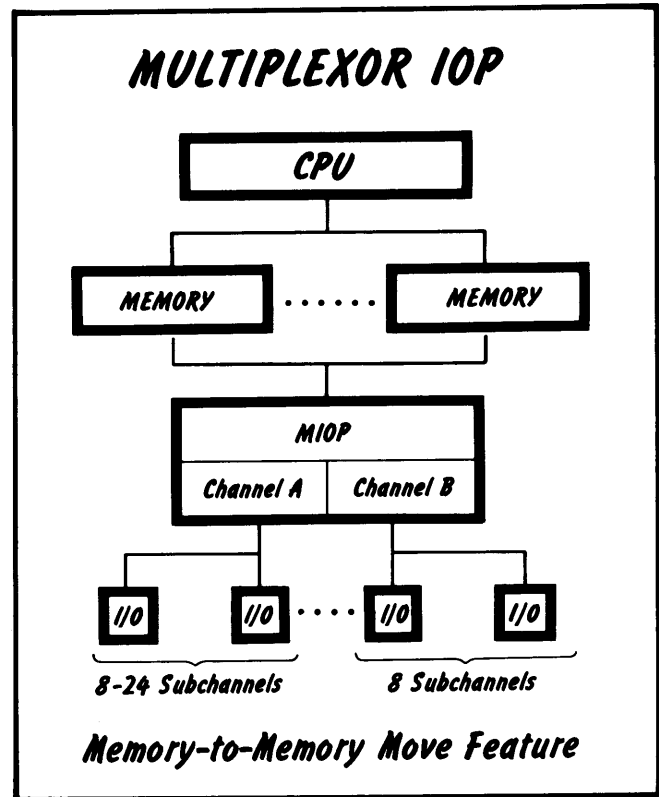
INPUT/OUTPUT — MIOP

Features —

- Dual channel capability with up to 24 subchannels (8 standard, expanded in increments of 8) on Channel A and 8 subchannels on an optional Channel B

- Channel transfer rate of 470,000 bytes per second

- Optional 4-byte interface doubles channel transfer rate (940,000 bps)

- Memory-to-memory move option (requires dedication of subchannels 8 and 9 on Channel A)

- Memory-to-memory move is a symbiont operation

Benefits —

Provides simultaneous operation of up to 24 devices on Channel A, concurrent with simultaneous operation of up to eight devices on channel B

Aggregate MIOP bandwidth of 940K bytes per second

Aggregate MIOP bandwidth of 1.88M bytes per second

Allows user to move data from one area of core to another at a rate of 0.5M words per second

Once initiated, no CPU intervention is required for execution. If requested, the CPU is interrupted at the end of the move

CHART #14   Originated by: Marketing Technology

NOTES
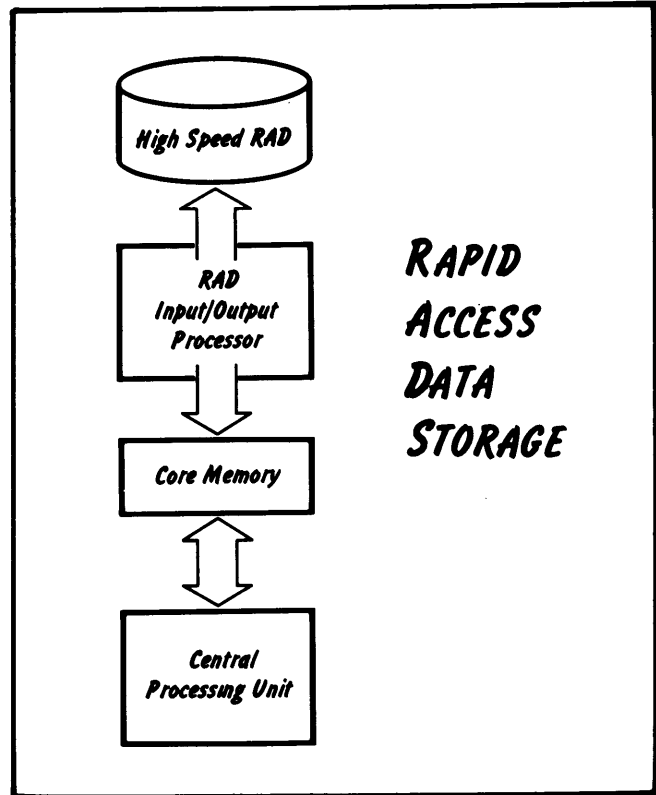
RIOP

This new I/O processor combines functions of the earlier SIOP* and the high speed RAD (7212) device controller into a single package.

Features —	Benefits —
• Controls up to four Model 7212 High Speed RAD's	Over 5.3M bytes of storage each RAD
• Average transfer rate of 2.47M bytes per second	Extremely high speed swapper – minimum system response for time-sharing users
• Fully asynchronous from central processing	Maximum system efficiency

---

* Note to salesman: Sigma 9 cannot receive an SIOP.

CHART #15   Originated by: Marketing Technology

PERIPHERALS

Computer systems are required to interface with
the world in a wide variety of ways. XDS offers
a complete line of peripherals for this purpose.
The range is from analog to digital converters
(SIU's) to the fixed head per track disk (RAD).
All of these peripherals are in use on the Sig-
ma 5 and Sigma 7 and are fully compatible
with the Sigma 9. Conversion causes minimum
difficulty.

SIGMA PERIPHERALS

RADS
Magnetic Tape Units
Removable Disk Storage
Card Readers
Card Punches
Line Printers
High Speed Paper Tape
Communications Controllers
Systems Interface Units
Plotters
Graphic Display
Remote Batch Terminal
Magnetic Card Memory

NOTES

CHART #16   Originated by: Marketing Technology

SOFTWARE — GENERAL

A computer system is only as good as the software that is available to it. XDS has a comprehensive set of proven software for the Sigma 9. All current operating systems will run on Sigma 9. The typical operating system for Sigma 9 will be UTS or XOS. Both of these operating systems interface to an extensive library of processors.

- COBOL
- MANAGE
- Data Management System (DMS)
- META-SYMBOL
- FORTRAN
- BASIC
- FMPS
- SL-1
- Plus a full range of utility programs. Both operating systems are designed to serve the full spectrum of user needs — batch processing, time-sharing, and real time

UTS

UTS is a field proven multi-use operating system.

- Simple for on-line users to learn and use
- Many users in core simultaneously – highly responsive
- Complex scheduling algorithm – makes use of system resources and maximizes throughput
- Full security for on-line, local, or remote batch users
- High batch throughput
- On-line diagnostics
- Full accounting services
- Automatic recovery capability

XOS

XOS – for multiprogramming and multi-use:

- Provides multiprogramming services for up to three job classes – parallel, operations, and serial

COMPREHENSIVE SOFTWARE SERVICES

OPERATING SYSTEMS

LANGUAGE PROCESSORS

UTILITY PROGRAMS

NOTES

CHART #16 (Cont'd)   Originated by: Marketing Technology

- Permits console initiation of jobs typically found in the parallel job class – no limit to the number of parallel jobs allowed in memory at one time
- Supports 5 sub-classes by an operations job class, each with a job in core and an open file of waiting jobs on the RAD
    - Jobs of the operations class are the production jobs requiring relatively h high I/O
    - Jobs of the serial class are the heavy computing, low I/O type jobs
- Successfully manages several jobs simultaneously in memory, while at the same time improving an installation's efficiency through resource management

CHART #17   Originated by: Marketing Technology

NOTES

RELIABILITY

✓ *Complete Parity Checking*

✓ *Memory Fault Detection*

✓ *Programmable Clock Margins*

✓ *Partitioning Controls*
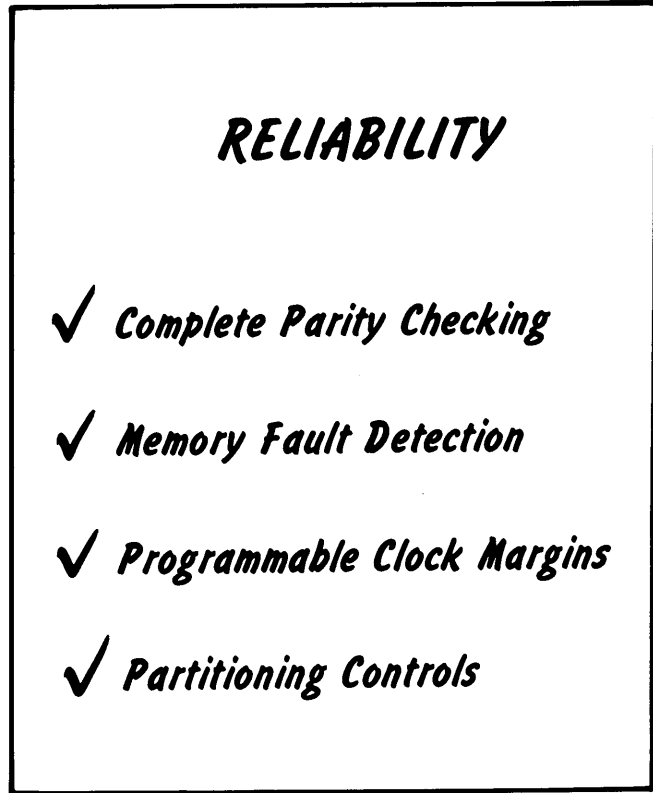
AVAILABILITY: RELIABILITY

Maximum system availability is assured by unique reliability features incorporated into the
Sigma 9.

Features —

- Parity checked on all information communicated memory and processors

- Partitioning controls permit isolation of failed unit from system

- Program testing of clock and voltage margins

Benefits —

Complete fault detection/location permits partitioning to isolate fault

Assures maximum system availability even during maintenance operations

Simplifies task of locating intermittent fault

CHART #18   Originated by: Marketing Technology

NOTES

```
┌──────────────────────────────────────────┐
│                                          │
│        MAINTAINABILITY                   │
│                                          │
│                                          │
│    ✓ Snapshot Feature                    │
│                                          │
│    √ Error Logging                       │
│                                          │
│    √ Memory Scan Feature                 │
│                                          │
│    √ CPU-MIOP Maintenance Interface      │
│                                          │
│    √ MIOP Maintenance Subcontrollers     │
│                                          │
│                                          │
└──────────────────────────────────────────┘
```

AVAILABILITY: MAINTAINABILITY

Maximum system availability is assured by unique availability features incorporated into Sigma 9.

Features —

- Snapshot features in memory, CPU, and IOP'

- Extensive error logging

- Manual memory clear and scan

- Maintenance interface bus

- Maintenance subcontroller on each I/O channel

Benefits —

Enables diagnostics to determine internal system status at time of failure for quick, easy fault identification

Maintenance personnel can clear, read, or store data into consecutive memory locations

Permits exercise of selected functions in the IOP's using special RD/WD instructions

Enhances diagnosis of I/O system failures

CHART #19   Originated by: Marketing Technology

SIGMA 9

The Sigma 9:

- Continues the field-proven advanced archi-
  tecture of the Sigma family with a large
  multi-processor system
- Provides complete software compatability
  with Sigma 7
- Insures maximum system effectiveness and
  availability through advanced software
  concepts and unique fail-safe design
- Provides multi-dimensional growth in
  core, porting, and I/O processors

Sigma 9 is the system which meets your data
processing requirements today and insures that
your future needs will be served.

SIGMA 9

Extension of Proven Sigma Architecture

Sigma 7 Compatible

Superior Price/Performance

Large Growth Capability

NOTES

INTRODUCTION

This presentation is made up of 20 visuals and accompanying outlines. To orient your audience, you may want to start with the Sigma 9 Management presentation, 67-04-48. An overview will help to place XOS in perspective.

The attached presentation outline will guide you in structuring your talk around the charts. For more details about XOS, consult:

> XOS General Information Manual
> Sigma 9 Announcement Notebook – Section 3.2
> Sales Manual Updates
> XOS Announcement Videotape

CHART #1

Xerox Operating System (XOS) is an advanced
operating system featuring optimum operational
scheduling flexibility, with strong file manage-
ment and a communications management system
with dynamic network restructuring.

Xerox
Operating
System
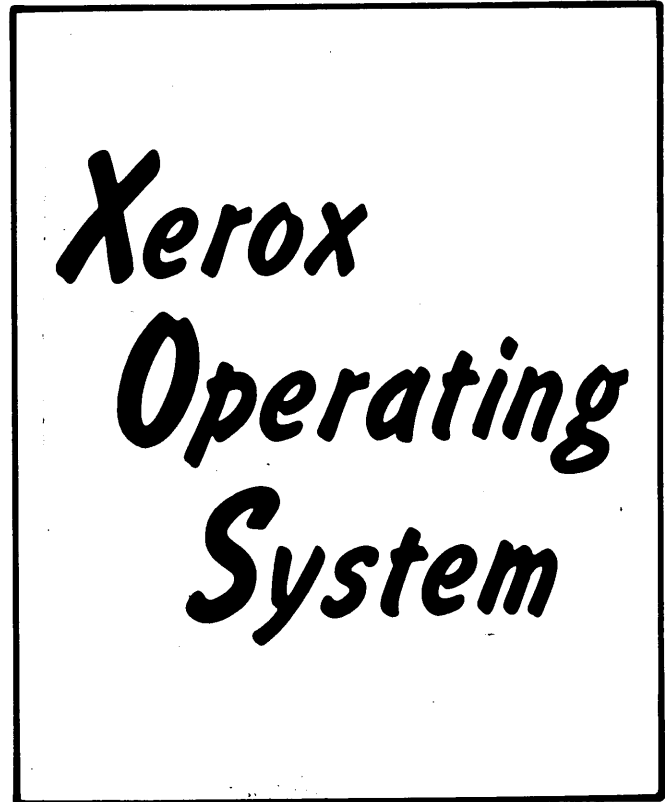
NOTES

CHART #2

ADVANCED MULTIPROGRAMMING OPER-
ATING SYSTEM

- **Maximized Throughput**
  - Multiple jobs in memory
  - Job scheduler
  - Task Management – use of external interrupts
  - Resource management
  - Symbionts – input, output, telesymbionts
  - Overlapped I/O and processing

- **Reliability and recovery**
  - Error Protection – file security – memory map
  - Checkpoint/Restart
  - Automatic Restart after system failure

- **Ease of programming and operation**
  - Simple command language
  - System is flexible
  - Wide choice of processors

- **Installation Control**
  - Priority assignments
  - Generate system to match needs and configuration
  - Operator's control of job stream

**XOS
AN ADVANCED
MULTI PROGRAMMING
OPERATING SYSTEM**

✔ *Maximized Throughput*

✔ *Reliability & Recovery*

✔ *Ease of Programming & Operation*

✔ *Installation Control*

NOTES

CHART #3

MULTI-MODE —

- Local Batch Processing
  - Stress high throughput for "scientific" and business applications
  - Features enable high throughput; i.e., multiprogramming rate, scheduler, etc.
  - Processors available in system
  - Monitor services available

- Remote Batch
  - Stress communications enabling re-mote batch processing
  - Remote batch user is treated just like local batch user once he's in the system

- Real Time
  - Review real time features of Sigma hardware
  - Discuss type of real time applications where XOS could be used

- Time-Sharing
  - More than one user in core
  - Subsystems — Line Editor, BASIC, FORTRAN, Debug
  - User may use on-line or batch modes
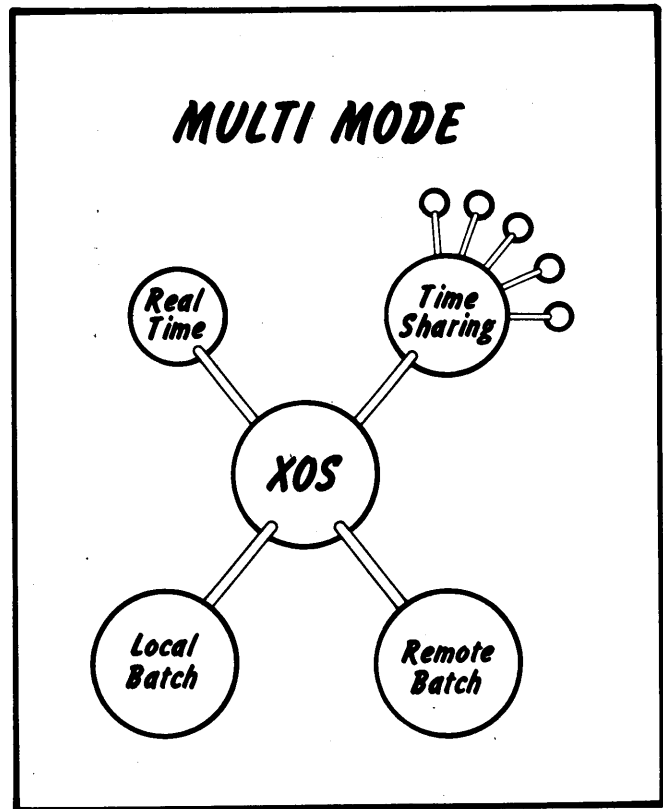
NOTES

CHART #4

MULTIPROGRAMMING FEATURES

- <u>Job Stream</u>
  - Parallel Class
  - Production Class and Subclasses
  - Serial Class
  - Examples of each class

- <u>Job Priorities</u>
  - Class and subclass priorities
  - Priorities within waiting job queues
  - Classes associated with external interrupt
  - Task management controls interrupts

- <u>Multi-Step Jobs and Super Jobs</u>
  - Examples
  - Control
  - Communication between jobs and super jobs

- <u>Resource Management</u>
  - Allocation at SYSGEN
  - LIMIT and SLIMIT cards
  - RESOURCE card

MULTIPROGRAMMING FEATURES

Job Scheduling by Class,
Subclass and Priority

Task Management by Priority

Job Predicate & Conditional
Execution

Resource Allocation by Dedication
or Contention

EFFICIENT, SYSTEM UTILIZATION

NOTES

CHART #5

JOB SCHEDULING (EXTERNAL PRIORITY)

- Job Scheduler
    - Resource inventory
    - Order of class search
        (1) Parallel
        (2) Production or Serial depending
            on way system is generated

- Parallel Jobs
    - Single step jobs
    - Operator initiated
        (1) Media conversion
        (2) Real time tasks
        (3) Cataloged jobs
    - Any single step job can be executed
      as a parallel job

- Production Jobs
    - Multi-step jobs
    - Up to five subclasses
    - Reason for five subclasses
    - Priority relationships between sub-
      classes
    - Priority relationships in waiting
      queues
    - Priorities established at SYSGEN

- Serial Jobs
    - Multi-step jobs
    - Selected by priority on job cards

JOB SCHEDULING

PARALLEL JOBS
- Single Step
- Operator Initiated

PRODUCTION JOBS
- Installation Specified Priority
- Dedicated Resource Option

SERIAL JOB
- Scheduled by External Priority

FLEXIBLE CONTROL
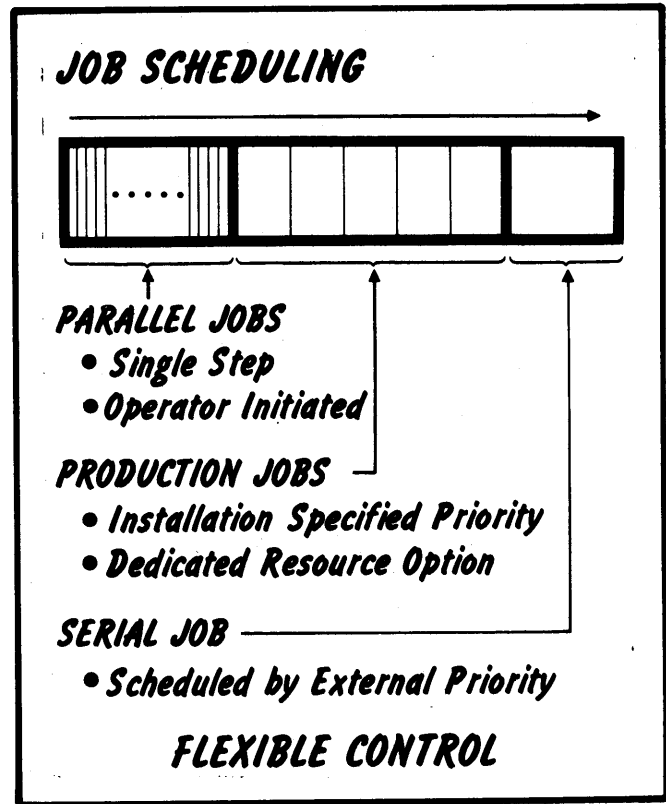
NOTES

CHART #6

TASK MANAGEMENT

- Use of Sigma external interrupts

- Benefits of using interrupts

- Levels used - up to eight

- Activation and control of tasks
  -     Queued on levels

- Waits on an event

- Time slicing for time-sharing
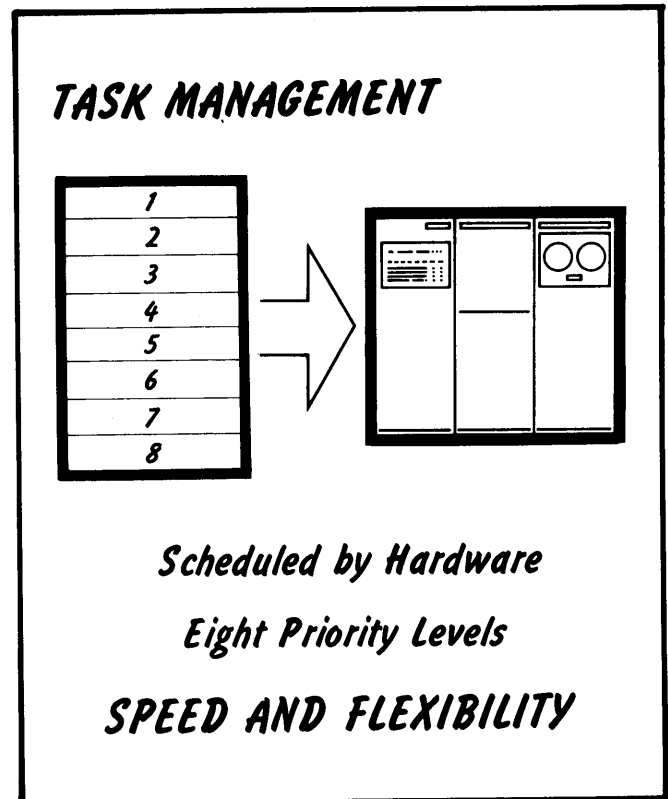
NOTES

CHART #7

JOB PREDICATE AND CONDITIONAL EXECUTION

Description —

- Job step — compile on load or run

- Job — compile, load and run

- "Super" job — many related jobs

- Predicated sequence
  - Necessary that jobs be run in given order

- Conditional Execution
  - Necessary that first jobs successfully complete before last jobs are started

- Jobs communicate through Job Switch Word

JOB PREDICATE AND CONDITIONAL EXECUTION

Step 1	Step 2	Step 1	Step 1	Step 2	Step 3

JOB A · JOB B · JOB C

SUPER JOB

Predicated Sequence

Execution – Dependence

31 Condition Switches
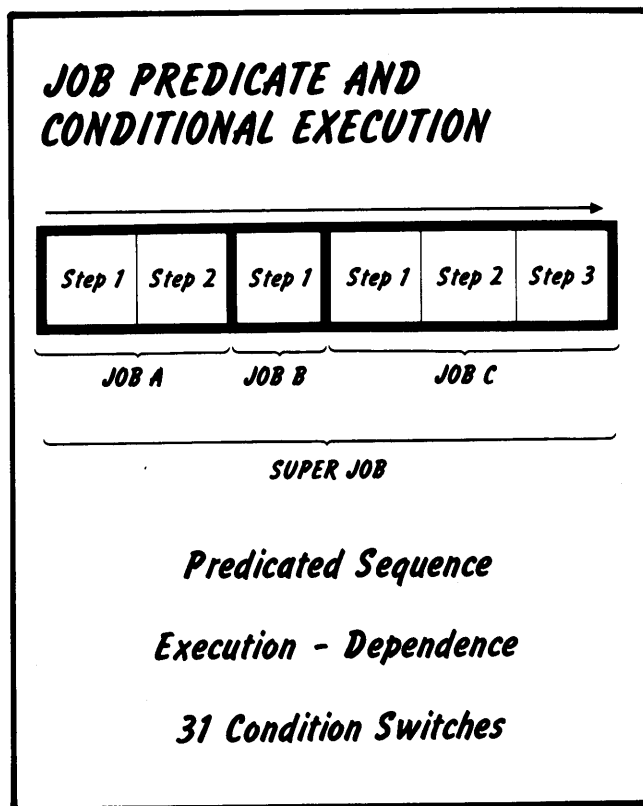
NOTES

CHART #8

RESOURCE ALLOCATION

- Pooled Resources
  - Logical peripheral devices
  - System assigns specific devices to a job
- Reserved Resources
  - Physical peripheral devices
  - User asks for a specific device for his job
- Shared Resources
  - CPU time
  - Core memory
  - Secondary storage
- Control cards affecting resource management
  - LIMIT — gives job limits for CPU time, core memory, and temporary disk storage
  - SLIMIT — gives job step, limits for core memory, and temporary disk storage
  - RESOURCE — specifies the minimum available resources necessary to enable a job to be scheduled

RESOURCE ALLOCATION

POOLED - Any Peripheral
(RESOURCE)

RESERVED - Any Peripheral

SHARED - Core Memory and Rapid Access Storage
(LIMIT & SLIMIT)

NOTES

CHART #9

CONTROL COMMAND LANGUAGE

- **Default Options**
  - Established at SYSGEN
  - Provide simpler control language – installation flexibility
  - May be overridden
- **Modes of Operation**
  - Diagnostic without job execution
  - Diagnostic with job execution
  - Diagnostic with control command cataloging
- **Control Command Cataloging**
  - Simplifies running of frequently used jobs
  - Standardizes running of frequently used jobs
  - Allows for inclusion and modification of pseudo parameters
  - Provides ability to have operator initiated jobs

CONTROL COMMAND LANGUAGE

- **EXTENSIVE DEFAULT OPTIONS**

- **UTILIZATION MODES**

  Diagnostics Without Execution

  Normal Execution

  Cataloging of Control Commands

- **ONE-PASS ERROR DETECTION FOR ENTIRE JOB**

  EASE OF OPERATION

NOTES

CHART #10

FILE MANAGEMENT SYSTEM

- **File Organization**
  - Sequential
  - Indexed sequential
  - Direct
  - Partitioned
- **Record Formats**
  - Fixed length
  - Variable length
  - Undefined

FILE MANAGEMENT

FILE ORGANIZATION
Sequential
Indexed Sequential
Direct
Partitioned

FILE FORMAT
Fixed
Variable
Unspecified

EASE OF PROGRAMMING
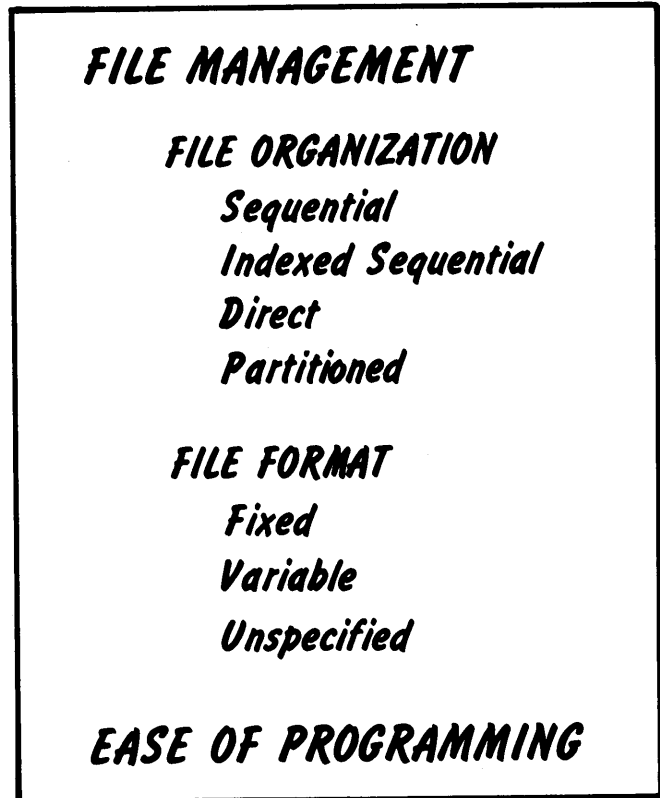
NOTES

CHART #11

VOLUME HANDLING

- Types of removable volumes
  - Private
  - Common
  - Account
  - Parallel or serial volume mounting

- Mounting messages
  - AVR feature

- Label Handling
  - ANS - IBM compatible labels
  - User specified labels
  - Non-standard labels

- Cataloging on disk
  - Account catalogs

- Super catalog
  - Catalog of account volumes

- Other Facilities
  - File concatenation - many files may be combined for use in one job
  - File generations - generation numbers are maintained for files
  - File versions - different versions of each generation may be maintained

---

VOLUME HANDLING

Single File	- Single Volume
Single File	- Multi Volume
Multi File	- Single Volume
Multi File	- Multi Volume

Flexible Mounting Options

LABEL HANDLING

TAPE :   ANS Standard
          User Label
          Non-Standard

DISK :   Standard
          Non-Standard

OTHER FACILITIES

File Concatenation
File Generation
File Version

NOTES

CHART #12

<u>ACCESS METHODS</u>

- <u>Assisted</u> — monitor supplies blocking, de-blocking, buffering
  - Sequential
  - Indexed Sequential
  - Partitioned
- <u>Unassisted</u> — user supplies own blocking, deblocking, buffering
  - Sequential
  - Virtual direct
  - Read direct
- <u>Access modes</u>
  - Move mode — records are read into system's buffer and are moved into user's work area
  - Locate mode – records are read into system's buffer but only a record index is given to user

---

ACCESS METHOD

ASSISTED - *Sequential Access (ASAM)*

  - *Indexed Sequential (AIAM)*

  - *Partitioned (APAM)*

NON-ASSISTED - *Sequential (VSAM)*

  - *Virtual Direct (VDAM)*

  - *Real Direct (BDAM)*

ACCESS MODE

MOVE MODE

LOCATE MODE

NOTES

CHART #13

TELECOMMUNICATIONS MANAGEMENT

- Facilities
  - TAM - Terminal Access Method
  - ATAM - extension to TAM

- Network Definition
  - At SYSGEN
  - Dynamic modification from program

- Telesymbiont

- Applications
  - Remote file exchange
  - Message switching
  - Data collection and distribution

NOTES

CHART #14

PROGRAM CHECKOUT SERVICES

- <u>Debugging aid available at:</u>
  - <u>User program level</u>
  - <u>Command card level</u>
- <u>Facilities</u>
  - Dump
  - Snapshot
  - Conditional tests
  - Logical operations
  - Event count

PROGRAM CHECKOUT SERVICES

DEBUGGING AID AVAILABLE
- *AT PROGRAM LEVEL*
- *BY CONTROL COMMAND*

FACILITIES
- *Memory Dump*
- *Snapshot*
- *Conditional Tests*
- *Logical Operations*
- *Event Count*

NOTES

CHART #15

AUTOMATIC ACCOUNTING

- Complete system accounting
  - Processing
    - (1) CPU time
    - (2) I/O transactions - number of RAD accesses, etc.
    - (3) I/O volume - number of cards read, lines printed, etc.
  - Storage
    - (1) Core
    - (2) Tapes
    - (3) Disk Packs

AUTOMATIC ACCOUNTING

PROCESSING
- CPU Time
- I/O Transactions
- Input/Output Volume

STORAGE
- Core
- Tapes
- Disk Packs

NOTES

CHART #16

RELIABILITY AND RECOVERY

- Error Protection
  - Memory protection – memory map with access protection
  - Parameter validation
    (1) CCI diagnostic checks
    (2) Monitor call parameter checks
  - File security
    (1) Account and password used to lock files
    (2) Legal users are specified in owner supplied list
  - Peripheral reconfiguration
    (1) Operator may lock or unlock a peripheral device
    (2) Exchange of logical addresses between two peripherals of same type

- Recovery
  - Checkpoint/Restart
    (1) Checkpoint to a user specified file
    (2) Tape or direct access media
    (3) Envoked from user program
    (4) Reinitiates program from check-point file
    (5) Execution resumes at specified instruction or at instruction following checkpoint procedure
    (6) Many checkpoint/restarts may be made within single job
  - Suspension and Reactivation — operator may suspend and reactivate a job
  - Activity queue to reflect state of interrupts at crash
  - On a crash, only executing job is lost and only output symbiont files in catch-up mode are lost

---

RELIABILITY AND RECOVERY

ERROR PREVENTION

- *Memory Protection*
- *Parameter Validation*
- *File Security*
- *Peripheral Reconfiguration*

RECOVERY

- *Checkpoint/Restart*
- *Suspension and Reactivation by Operator*

NOTES

CHART #17

SYSTEM MANAGEMENT

- <u>Priority Assignment</u>
  - To Production and Serial classes
  - At SYSGEN time
  - Changeable via operator intervention (serial class only)

- <u>Resource Limit Control</u>
  - Assign resources limits to job classes at SYSGEN
  - Establish core and secondary storage limits
  - Total units assigned may exceed total available

- <u>Operating System Structure</u>
  - Size of resident monitor, 16K resident
  - Management of non-resident segments

- <u>System Configuration</u>
  - Modular SYSGEN to tailored system
  - Peripherals may be locked out of system

- <u>Performance Monitoring</u>
  - Routines accumulate performance statistics

- <u>Account Supervision</u>
  - Operator may add, delete or modify account numbers in super catalog
  - Enables control over active accounts and their files

SYSTEM MANAGEMENT

Priority Assignment

Resource Limit Control

Operating System Structure

System Configuration

Performance Monitoring

Account Supervision

NOTES

CHART #18

SYSTEM SERVICES

- Symbionts
    - Input
    - Output
    - Telesymbionts

- Operator communication
    - System sends mount and status messages
    - Operator may request job status, job initiation, system status, and peripheral control

```
SYSTEM
SERVICES

SYMBIONTS

OPERATOR
COMMUNICATION
```
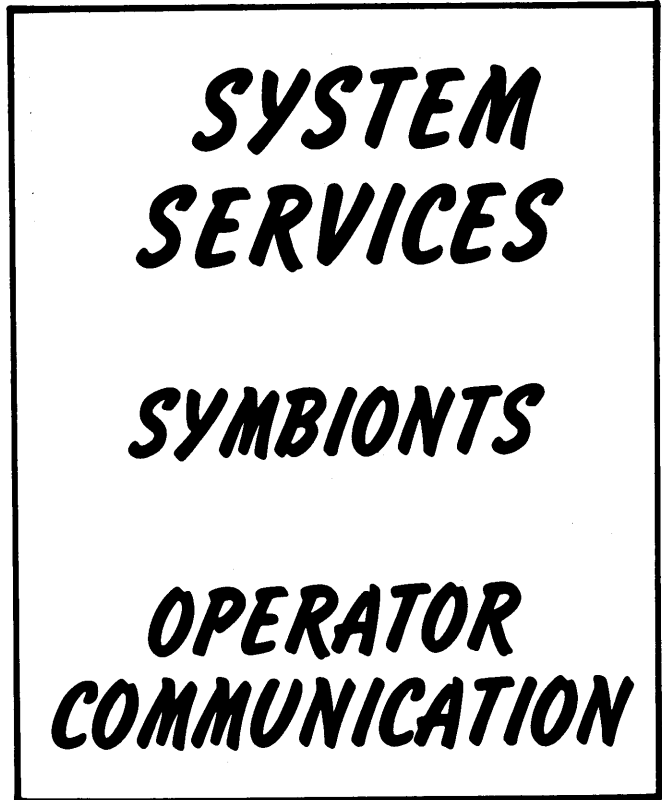
NOTES

CHART #19

PROCESSORS

- XDS COBOL
- XDS FORTRAN IV
- META-SYMBOL
- DMS
- SORT/MERGE
- Linkage Editor
- Media Conversion Program Generator
- Test File Generator
- SYSGEN
- File Management Processors

PROCESSORS

XDS Cobol

XDS Fortran IV

Meta Symbol

Data Management System

Sort/Merge

Linkage Editor

Media Conversion Program Generator

Test File Generator

Sysgen

File Management Processors
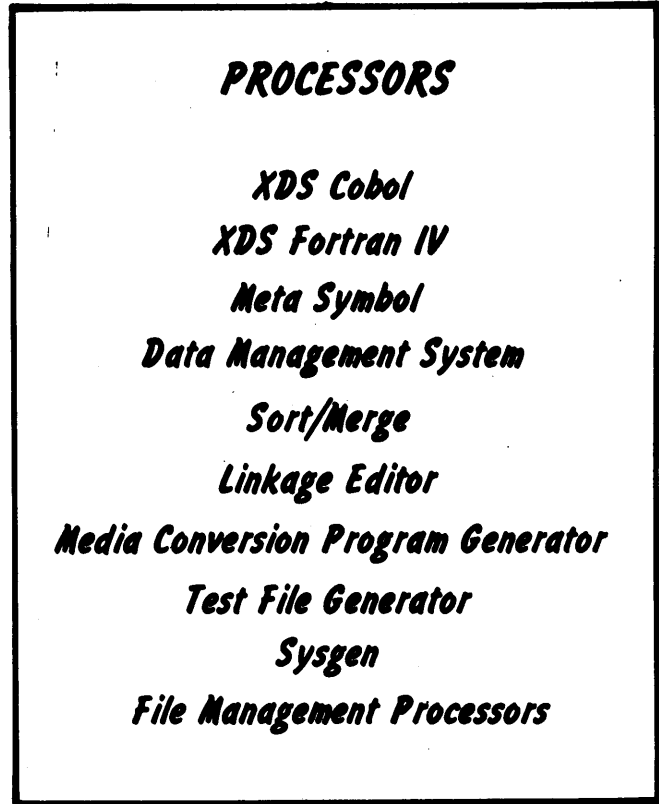
NOTES

CHART #20

XOS — A MULTIPROGRAMMING SYSTEM

- Multiprogramming with high throughput
  - Many jobs active in memory
  - Efficient task, job, and resource management
  - Takes advantage of Sigma interrupts and I/O processing overlap

- Multi-Use
  - Local and remote batch — XOS's strongpoint
  - Real time
  - Time-sharing
  - Communication and transaction processing capabilities

- Installation Control
  - Tailor-made systems through modular SYSGEN
  - Complete control over priority scheme within system

XOS

MULTIPROGRAMMING

Multimode

Throughput & Flexibility

Installation Control
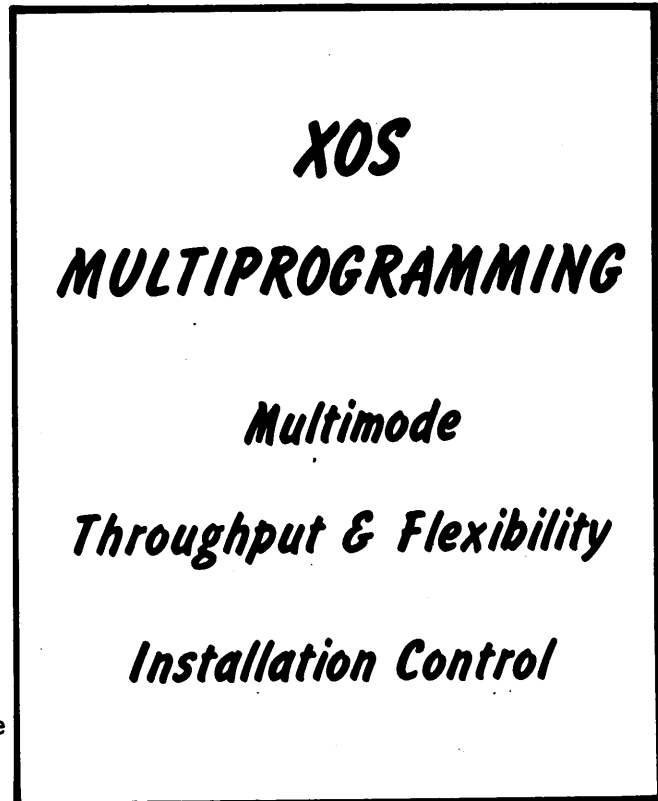
NOTES

INTRODUCTION

UTS is a multi-use operating system designed as a <u>TOTAL</u> system, with both users and installation management in mind.

System users, as a rule, are primarily interested in a service that is easy to use and easy to understand. UTS offers this through a simplified and logical control language for both the on-line user and the batch user. In addition to a simplified control language, the on-line users — presumably less computer-oriented than batch users — have two additional features available that make UTS easy to use:

- The first of these features is Terminal Executive Language (TEL). The majority of on-line processors may be called implicitly through TEL, thus allowing the user to interface with a single executive language for the majority of his requirements. Because he need not concern himself with multiple-levels of control language, this use of implicit capabilities greatly reduces the time it takes him to "learn" the system.

- The second feature that simplifies system usage for the on-line user is Automatic Program Association. This allows the user to be directly connected to a program — either a standard UTS processor or a user-written program — immediately upon logging into the system. In other words, the user may take advantage of UTS's services with no prerequisite knowledge other than how to successfully log into the system.

Not far behind ease-of-use, and perhaps of equal importance to the user, is system responsiveness. Sigma 6/7/9 hardware offers a high speed CPU and large main memory configurations, allowing many users to be in memory at one time, as well as a high speed RAD swapping device to minimize primary/secondary memory transfer times. This combination of hardware features, fully utilized by the UTS software, offers a total system that is truly responsive to the user population.

In addition to ease-of-use and responsiveness, a system which offers multi-use capability is also of importance to users. Illustrations of this are the time-sharing user who may submit jobs to the batch stream for processing and the user who has the ability to create and organize files in the batch mode, and then interrogate and modify them through an on-line terminal. This multi-use capability, coupled with compatibility among the various modes of operation, offers the user a powerful, yet flexible, system.

UTS also offers significant features to system management personnel. Of prime concern to installation management is — Are system resources being used and allocated in the most efficient manner? The system monitoring features of UTS provide an answer to this question and, even more importantly, allow dynamic modification of specific parameters to increase the system's efficiency.

Another main concern of the data center management is security. UTS provides a number of security checks that protect against illegal use or misuse of the system. All UTS users, be they

INTRODUCTION (Continued)

local batch, remote batch, or time-sharing users, must log onto the system and supply their name, their account, and their password (optionally, at the data center manager's request), before they are allowed access to the system. These same security measures protect user files against un-authorized access.

Additional security measures are provided when the UTS system is generated and as additional users are authorized access to the system. At this time, specific privilege levels relating to central site peripheral usage, core memory usage, I/O calls allowable, and so forth, may be assigned to each user on a name and account basis.

Accounting is a critical item to installation management. UTS offers an accounting system that is both comprehensive and flexible. All users are monitored and tracked under a common accounting system that periodically measures units of the system's resources that are being used (CPU execute time, core storage, I/O requests, etc.). Upon logging off the system, these "usage" units are totaled and are related to "billable" units through one of eight possible rate tables. This is the real flexibility in UTS's accounting scheme — the ability to assign different users or classes of users to various rate tables as a function of their demands on the system. The installation management may either change specific charge units within the tables, or the user may be reassigned to an entirely different rate table.

Integrity and reliability of the service offered must also be of concern to installation management. UTS is designed to make the total system as reliable as possible through the use of automatic error recovery techniques, extensive failure analysis features, and on-line diagnostics.

For instance, automatic recovery capability is provided under UTS through continuous and extensive internal consistency checks. These checks are made to anticipate hardware and/or software errors before they propogate through the system and cause a failure. Once an anticipated error is detected, UTS immediately proceeds to an automatic recovery process that allows it to bootstrap itself, thus preventing system failure. In those instances where errors occur that could not be anticipated, UTS takes emergency measures to save the current environment of the system. Extensive analysis programs allow the interrogation of this saved information so the specific reason(s) for a failure may be pin-pointed. While such errors rarely occur, it is comforting to know that, if they do, tools are provided to answer the question — Why?

On-line diagnostics add to the techniques used in UTS to offer a reliable system. These diagnostics allow a Customer Service Representative to dial UTS from a remote location (just as another user) for the purpose of exercising central site peripherals. This can be done without the knowledge of and without affecting other system users.

With this short description, I hope you can see that UTS really is a system that was designed for both users and installation management.

NOTE

The charts comprising this presentation are a combination of new, re-vised, and existing UTS charts. Every salesman and sales office has

INTRODUCTION (Continued)

already received a copy of the existing charts. Not only have charts been added, deleted, and revised, but they have also been numerically re-ordered to illustrate the flexibility of this presentation.

CHART #1   Originated by: Marketing Technology

The Universal Time-Sharing System (UTS) is a
field-proven, time-sharing system offering a
variety of operating modes, extensive soft-
ware, complete accounting, flexible installa-
tion management, and reliable operations on
the advanced architecture of the Sigma 6, 7,
and 9.

U T S

**Universal
Time Sharing
System**
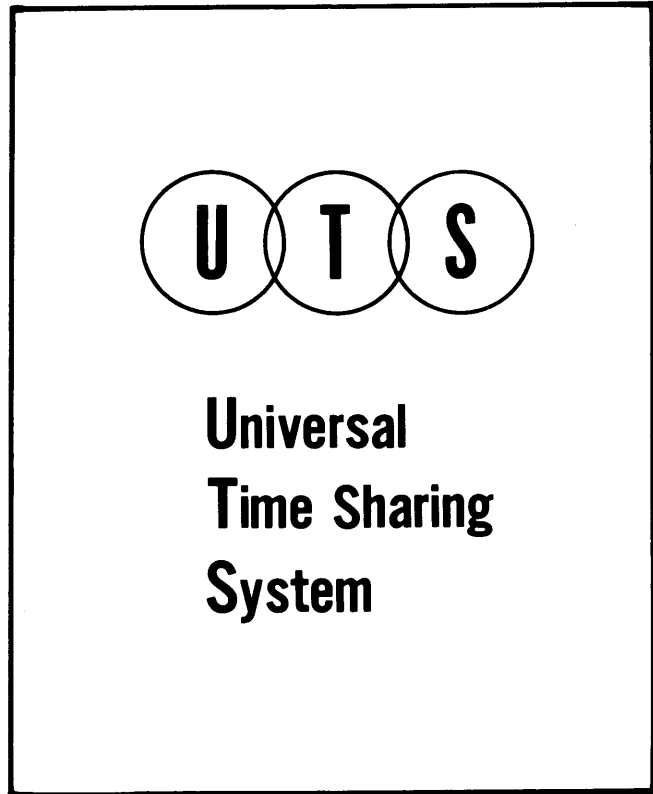
NOTES

NOTE: USE THIS CHART FOR SIGMA 7 AND 9
       ONLY. Sigma 6 does not have real time
       capability.

UTS operates in three concurrent MODES:

- Time-sharing
- Batch
- Real time

Modern computing needs frequently require a
combination of these modes. For example, a
billing system may gather data in the on-line
mode and prepare summary reports in the batch
mode.

In addition to local batch, remote processing is
available via the high speed remote batch ter-
minal. Any number of local and remote batch
peripherals can be used in conjunction with
symbiont operations.

Terminal operations provide for time-sharing,
on-line transaction-oriented data entry, and
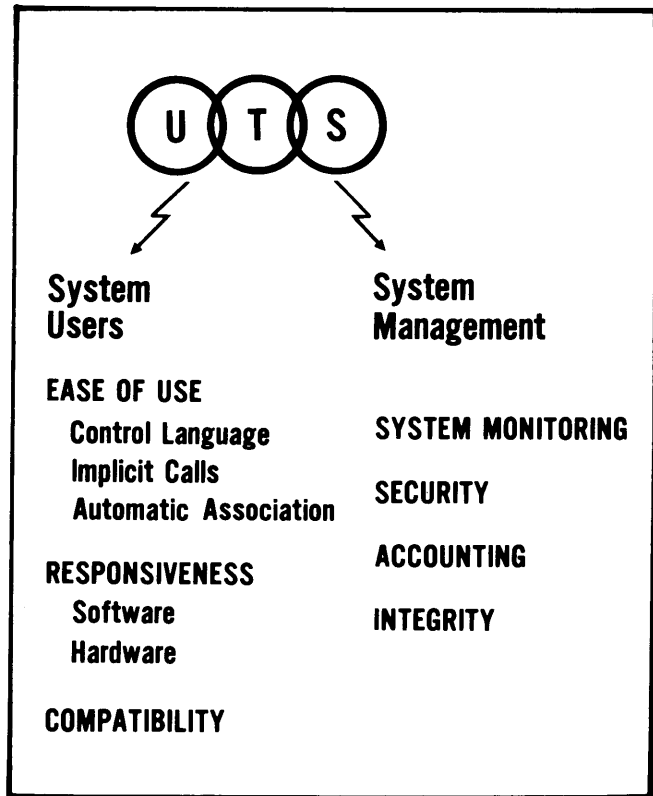remote entry of jobs into the batch stream.

The third level of multi-usage is provided by
the real time capability. Real time programs
may be core resident or brought in from secon-
dary storage by checkpointing the required
core memory. Various system resources may be
dedicated to real time users as required.

Multimode

Batch   **Time Sharing**   **Real Time**

Local
Remote

Conversational
Deferred Batch

Resident
Non-Resident

plus

Symbionts
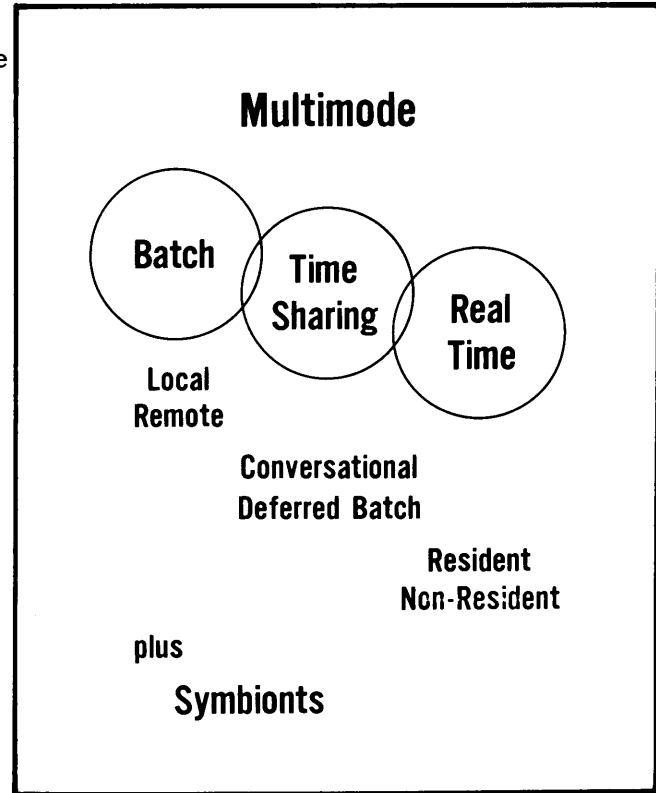
NOTES

NOTE: USE THIS CHART FOR SIGMA 6 ONLY.

UTS operates in two concurrent MODES of op-
eration — time-sharing and batch. Modern
computing needs frequently employ a combina-
tion of these modes. For example, a billing
system may gather data in the on-line mode
and prepare summary reports in the batch mode.

In addition to local batch, remote processing
is also available via the high speed remote
batch terminal. Any number of local and re-
mote batch peripherals can be used in conjunc-
tion with symbiont operations.

Terminal operations provide for general time-
sharing, on-line transaction-oriented data
entry, and for remote entry of jobs into the
batch stream.

Multimode

Local Batch

Remote Batch

Terminal Operations

- **On-Line Transactions**
- **Time Sharing**
- **Remote Job Entry**

plus

Symbionts
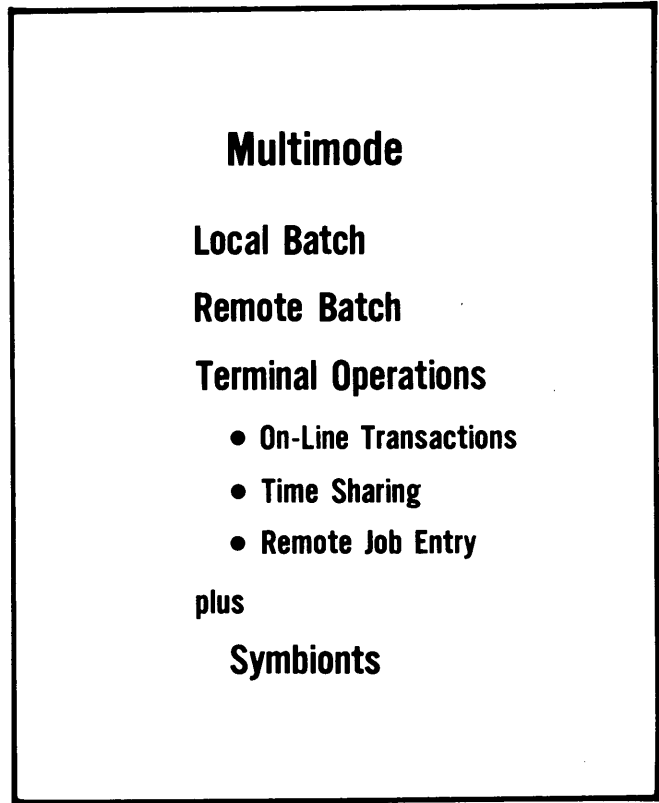
NOTES

CHART #25   Originated by: Marketing Technology

This chart is an overall summary of UTS. It discusses what the operating system can do for the system users and system management. See the INTRODUCTION for a detailed discussion.

U T S

System Users

System Management

EASE OF USE
- Control Language
- Implicit Calls
- Automatic Association

SYSTEM MONITORING

SECURITY

RESPONSIVENESS
- Software
- Hardware

ACCOUNTING

INTEGRITY

COMPATIBILITY

NOTES

NOTE: USE THIS CHART FOR SIGMA 7/9
       ONLY.

The Sigma 7/9 is a third-generation, memory-oriented machine. The CPU is attached directly to memory. All other devices are interfaced to memory with IOP's which also have independent access directly to memory.

The time-sharing terminals are teletypes or similar devices. The swapping RAD contains current users awaiting access to core, the on-line processors, and the monitor. User storage (disk, RAD) contains symbionts, user files, and programs.

Symbiont operations allow for simultaneous I/O (between peripherals and RAD) and computing to occur. Symbionts use the CPU only for initialization. Symbionts use core for buffering I/O.

A variety of real time processes can be interfaced via the Sigma hardware interrupt system. Resident and non-resident programs are supported.

System Architecture

CPU

Real Time

Memory

Swap RAD

User Storage

I/O
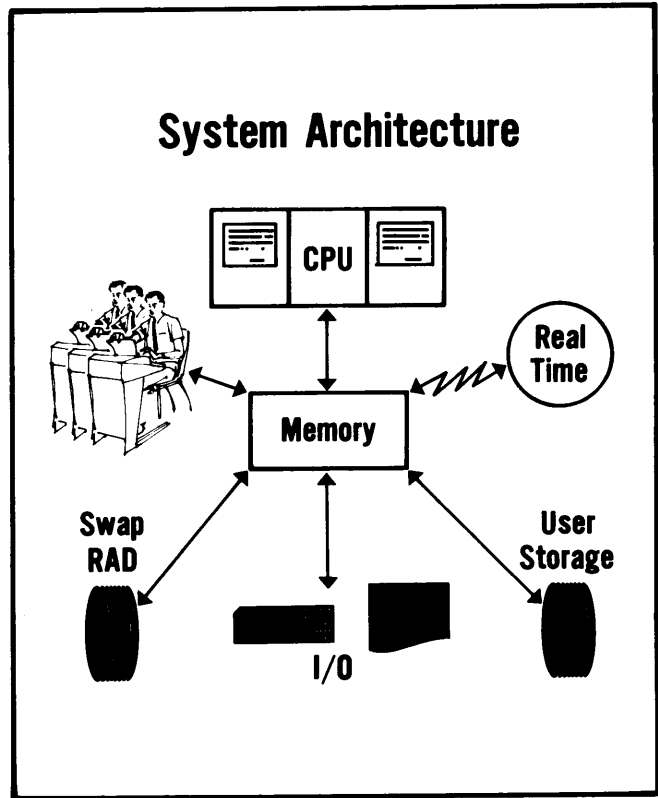
NOTES

NOTE: USE THIS CHART FOR SIGMA 6 ONLY.

The Sigma 6 is a third-generation, memory-oriented machine. The CPU is attached directly to memory. All other devices are interfaced to memory with IOP's which also have independent access directly to memory.

The time-sharing terminals are teletypes or similar devices. The swapping RAD contains current users awaiting access to core, the on-line processors, and the monitor. User storage (disk, RAD) contains symbionts, user files, and programs.

Symbiont operations allow for simultaneous I/O (between peripherals and RAD) and computing to occur. Symbionts use the CPU only for initialization. Symbionts use core for buffering I/O.

System Architecture

CPU

Swap RAD   Memory   User Storage

I/O
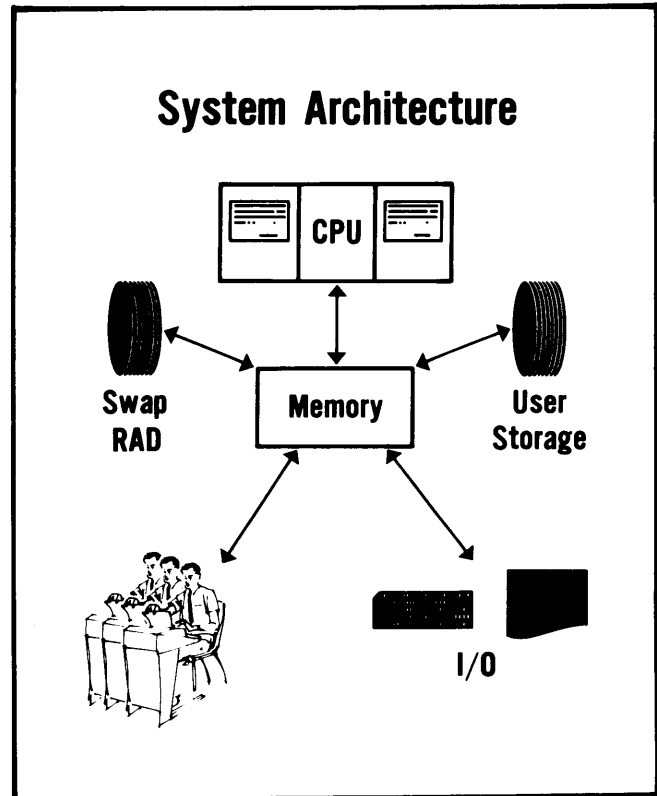
NOTES

CHART #21-1 Originated by: Marketing Technology

A time-sharing system cannot be efficient and responsive unless its scheduling techniques are themselves efficient.

UTS provides a multi-level, queue-scheduling discipline to resolve the following factors:

- What job should be executed next
- What job is eligible for swapping into core memory
- What job must be swapped out

Each user is associated with one of over twenty-five (25) different states. Thus, when a terminal user initiates an event (such as I/O), he will be moved to another state, possibly of lower priority. The scheduler uses this event-driven technique to avoid burdening the system by the overhead associated with unsophisticated polling techniques. It is the heart of UTS, designed to optimize total system efficiency.

Scheduling

PURPOSE
Optimize System Efficiency

DECISIONS
Execution
Swap - In
Swap - Out

METHOD
State Queues
Event Driven

NOTES

CHART #21-2   Originated by: Marketing Technology

The efficiency and response of any time-sharing system is directly dependent on the sophistication of its scheduler. The multi-level queue-scheduling discipline in UTS provides a common algorithm to control two major system functions:

- Selection and organization of users to be swapped in and out of core memory
- Selection of users for job execution

The UTS scheduler assigns each user's program (whether batch or on-line) a unique state. As execution and I/O for the program proceed, events are reported to the scheduler causing a change to another state. For execution, a search is made for the highest priority user currently in core; for in-swap, a search is made for the highest priority user out of core. Selection of users for removal to the swapping device is determined on the order of descending likelihood of the need for the CPU. Thus, the scheduler will maintain in core several compute-bound and interactive users to overlap those users who may be swapping or doing file I/O.

Priority Scheduling

Memory

| U S E R # 1 | U S E R # 2 | U S E R # 3 | U S E R # N |

CPU   HI   HI   LO

Swap
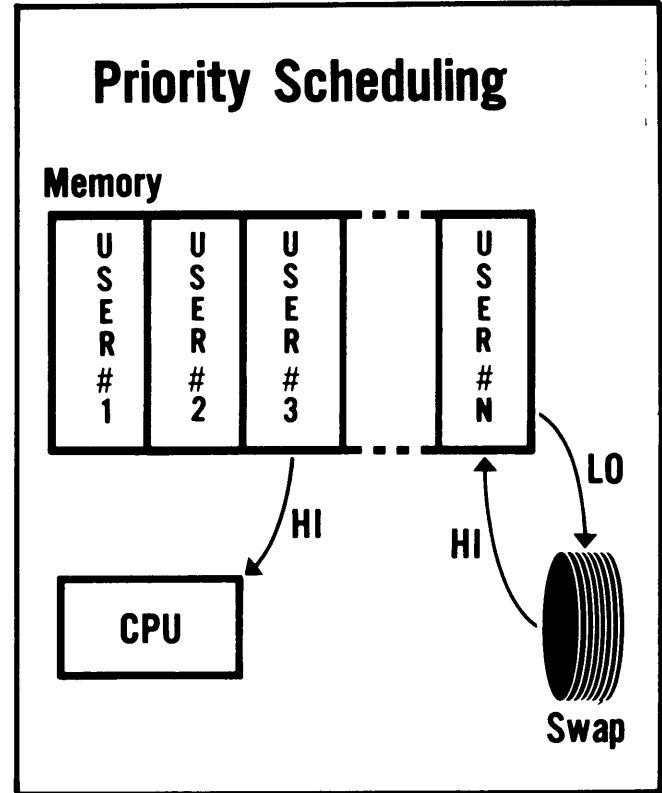
NOTES

CHART #22 Originated by: Marketing Technology

The XDS memory map surpasses all mapping techniques currently used in time-sharing systems. The map translates virtual addresses (used in a program) to the corresponding physical address as shown. Mapping also provides access protection codes for each page of memory. There are 256 registers in the map — one for each page of memory.

The primary use of the map is to fragment programs on a page basis for insertion into core memory. The program operates as if it were contiguous.

Mapping is completely transparent to the user. It is dynamic in that programs are not stored in the same pages after successive swaps. Mapping uses core more effectively than systems which do not fragment and is less time consuming than systems which use software for shuffling programs to condense core memory (consuming considerable CPU time).

Memory Management

Efficient Core Utilization
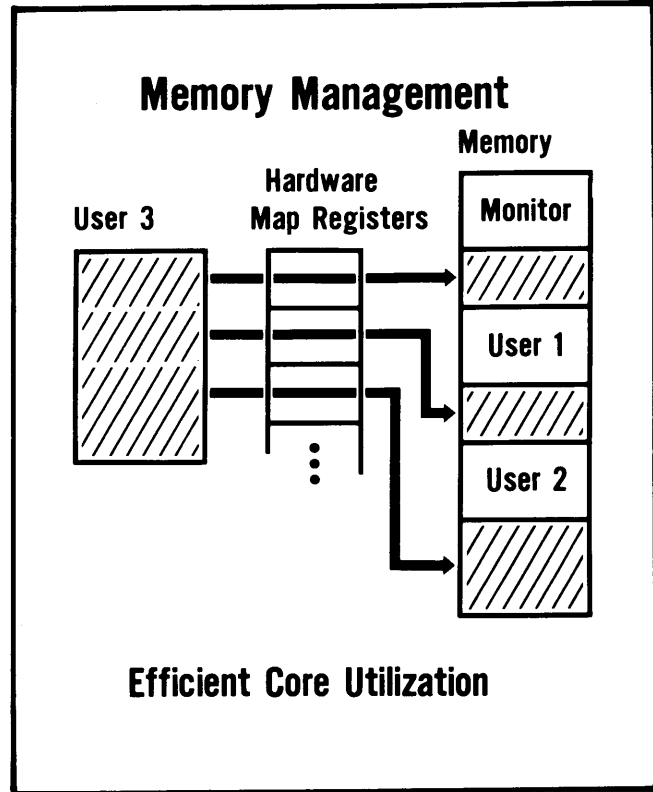
NOTES

CHART #16   Originated by: Marketing Technology

In an advanced system such as UTS, it is desir-
able to refine or tune the system for the best
use of resources. This is particularly important
because the user load is not clearly known in
advance of implementation, and the load will
change with time.

With UTS, the system is constantly being mon-
itored during its use and reports are prepared.
Based upon an analysis of these reports, various
control parameters may be adjusted or <u>tuned</u> to
<u>optimize system performance.</u>

**Dynamic
Resource Management**

System
Performance
Report

Tuning

System
Use

Adjust
Parameters
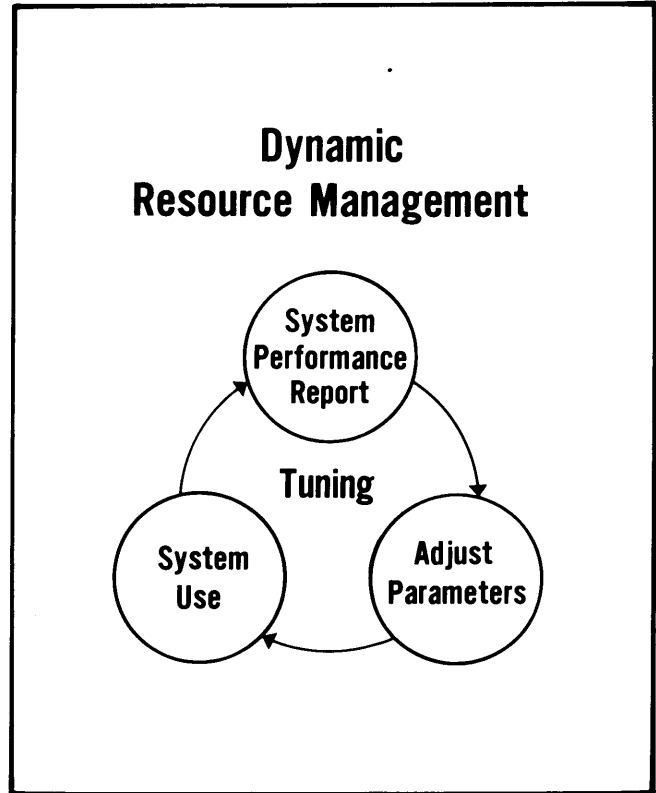
NOTES

CHART #17 (Rev A)   Originated by: Marketing Technology

The system performance report can be user or project oriented. It is based upon a <u>constant monitoring</u> of the system and includes <u>detailed reports</u> in each of the listed areas. As an example, the Summary report includes:

- Number of users
- Tasks per minute per user
- Percent of interactive interactions
- Milliseconds per interactive task
- 90% point for response time
- Execution multiplier
- Users in core
- RAD and tape reads and writes per second

These reports can be printed at preset intervals and/or on a request basis.

System Performance Reports

- **Summary**

- **CPU**

- **Processors**

- **I/O**

- **Terminal**

NOTES

CHART #19 (Rev B)   Originated by: Marketing Technology

The listed <u>parameters</u> are all adjustable <u>dy-</u>
<u>namically</u>. The percent of batch bias reflects
the percent of CPU time given to batch. The
terminal startup/shutdown will terminate on-
line processing and devote the system to batch.
These parameters allow installation management
to <u>allocate resources as desired</u>.

The <u>CONTROL</u> subsystem allows certain items
to be adjusted on a class basis (e.g., "The
amount of core for all on-line users"), while
the <u>SUPER</u> subsystem adjusts on individual users
and account numbers (e.g., "The amount of
core for user X").

Adjustable Parameters

- **User Storage**

- **Number of Users**

- **Time Quantum**

- **Batch Bias**

- **Terminal Shutdown**

NOTES

CHART #18   Originated by: Marketing Technology

User accounting and billing is automatically
provided by the system. Batch and on-line
accounting are identical. Multiple rate tables
may be employed for different types of users as
well as for different times of the day. This is
important for internal accounting or external
billing purposes.

Automatic Accounting

Processing

- CPU Time
- No. of On-Line Transactions
- No. of I/O Transactions
- Terminal Time

Storage

- Core
- No. of Tapes
- No. of Disk Packs
- File Size

NOTES

CHART #3   Originated by: Marketing Technology

Complete compatibility exists between batch
and on-line operations, with regard to many of
the processors, file management, and account-
ing. All on-line processors operate in batch
and most batch processors operate on-line as
well.

For those processors not available in the on-line
mode, EDIT may be used to create source files
which may, in turn, be submitted through termi-
nal batch entry for execution by the batch
processors.

Files created in the batch mode are available
to the on-line user and vice versa. Batch per-
ipherals are available to the on-line user via
PCL. Accounting routines treat batch and on-
line users identically.

Compatible

- **File Management
  System**

- **Language Processors**

- **Utility Programs**

- **Accounting**
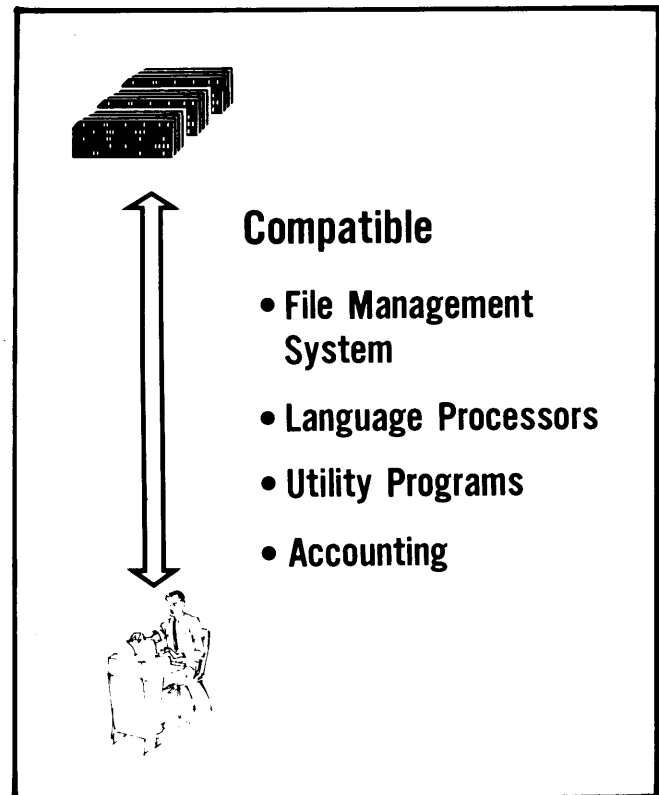
NOTES

CHART #4 (Rev A) Originated by: Marketing Technology

The available language processors are numerous, powerful, and offer a high degree of on-line and batch compatibility.

The processors range from the simple, easy-to-learn, easy-to-use on-line BASIC, through the very comprehensive business-oriented COBOL processor, to Extended XDS FORTRAN IV, which includes statements from almost every other FORTRAN written.

Language Processors

Cobol

Meta-Symbol

Manage

Fortran

DMS

Basic

FMPS

SL-1

NOTES

CHART #8-1

A monitor by itself is of little value unless appropriate language processors and service routines are available for the users.

For business and commercial applications, XDS COBOL offers a powerful and convenient programming language which conforms to the highest industry standards.

The processor itself, being state-of-the-art design, is overlayed in order to allow the user full language capability, while using minimum core. The results are greater capability and less cost to the user.

XDS COBOL

LANGUAGE FEATURES
  Segmentation
  Random Access
  Report Writer
  Library Facility
  Sort
  Table Handling

EXTENDED FEATURES
  Subcompile
  Debugging Statements

NOTES

CHART #9-1   Originated by: Marketing Technology

DMS is a generalized <u>data management system</u> which enables the user to <u>create</u> his <u>data base</u> according to data relationships which the user defines. Data may be structured in virtually any way to suit individual application requirements.

The DMS data base can be structured to support <u>batch processing</u> of data while facilitating <u>transaction-oriented</u> processing as well. With DMS, cumbersome second generation approaches to data management, such as bill of material processors, are replaced with a comprehensive and flexible data management system.

DMS services:

<u>Retrieval</u>

- Selective
- Current
- Economical

<u>Protection</u>

- Unauthorized access
- Destruction
- Audit trail
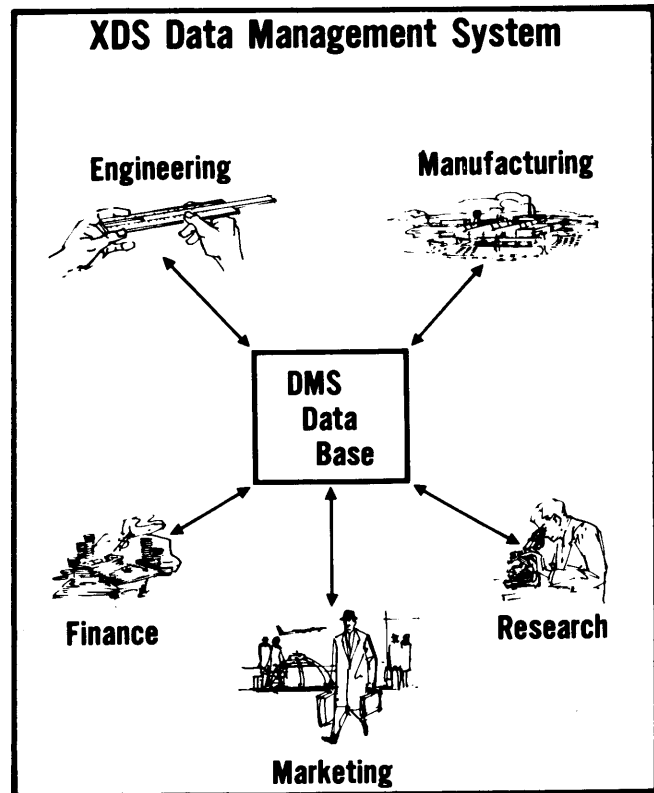
NOTES

CHART #8-2 Originated by: Marketing Technology

MANAGE is a generalized file management system. It is a tool offered by XDS to help solve one of the most prevalent computer-oriented problems — that of presenting data to a computer system and retrieving usable and readable information in an orderly and timely fashion. MANAGE assumes the responsibility for the detailed programming required in creating, maintaining, accessing, and generating reports from data files. This allows the MANAGE user to concentrate more on the specific business problem at hand.

MANAGE consists of four subsystems which may be used independently or collectively. These are:

- Data File Dictionary — describes the data file format and characteristics
- Fileup — creates and updates user files
- Retrieve — locates and extracts requested information
- Report — prepares printed reports from retrieved data

XDS Manage

GENERALIZED FILE MANAGEMENT

Report Generation

File Generation

Remote Access

File Security
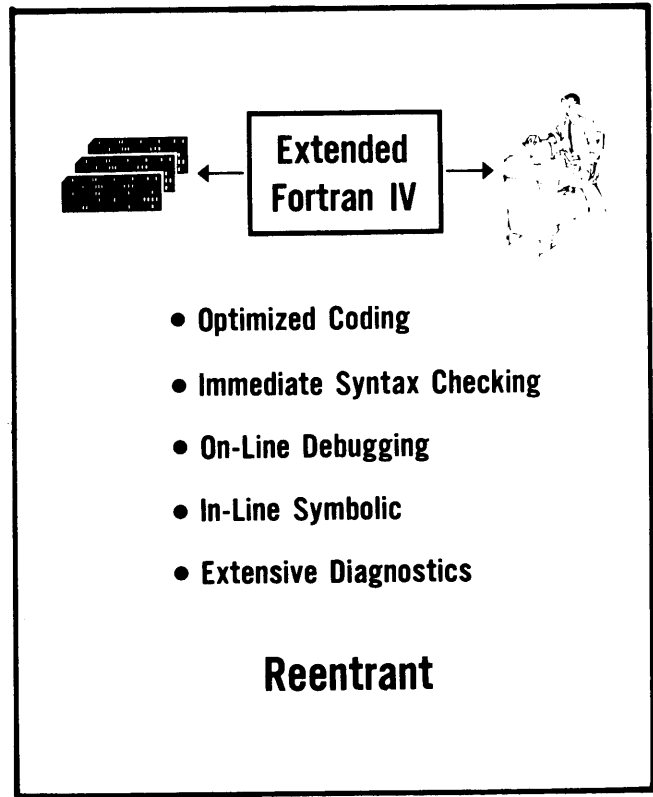
NOTES

CHART #6   Originated by: Marketing Technology

The UTS FORTRAN is a superset of most FOR-
TRAN IV's. A third pass provides code opti-
mization for rapid execution. On-line syntax
checking and line deletion are available. Ex-
tensive debugging facilities are available via
FDP. On-line operation insures rapid accessa-
bility for compilation.

The UTS FORTRAN compiler generates reen-
trant code and is, itself, reentrant. Multiple,
concurrent FORTRAN users require only a
single copy of the compiler in core which all
may share.

Extended
Fortran IV

• Optimized Coding

• Immediate Syntax Checking

• On-Line Debugging

• In-Line Symbolic

• Extensive Diagnostics

Reentrant

NOTES

CHART #5 (Rev A)   Originated by: Marketing Technology

One of the most popular time-sharing languages is the BASIC language because it is easy to use on-line. The extended UTS BASIC processor includes special features for easier debugging of programs, faster compiling and execution, and creation of more sophisticated user programs. Direct statement execution allows the terminal to be used as a desk calculator, permitting step-by-step computation and powerful debugging assistance.

Basic

- **Immediate Syntax Checking**
- **Integrated**
  - **Edit**
  - **Compile**
  - **Execute**
- **Direct Execute**
- **Character String Operation**

Reentrant
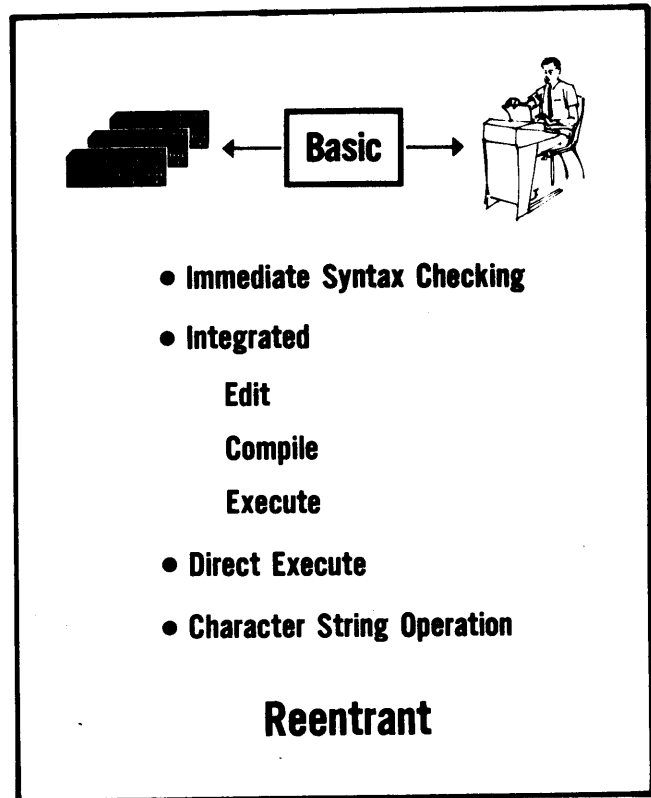
NOTES

CHART #7   Originated by: Marketing Technology

META-SYMBOL is a procedure-oriented macro assembler providing services available in sophisticated macro assemblers plus special features that permit the user to exercise dynamic control over the parametric environments of an assembly.

On-line syntax checking and line deletion are available. Extensive debugging facilities are available via DELTA. On-line operation insures rapid accessability for assembly.

| Meta-Symbol |

- **Procedure Capability**
- **Immediate Syntax Checking**
- **On-Line Debugging**
- **Extensive Directives**

Reentrant
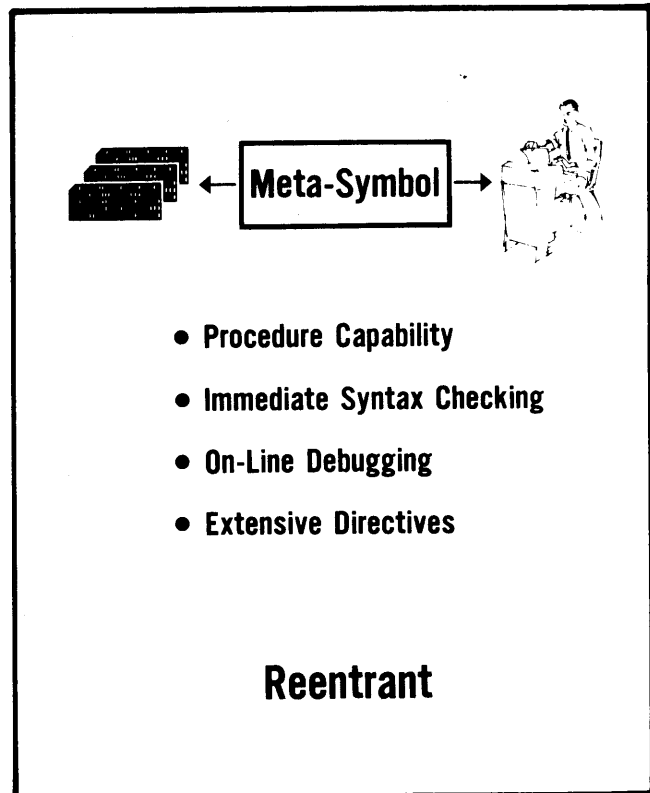
NOTES

CHART #9-2 Originated by: Marketing Technology

SL-1 is a <u>problem-oriented programming lan-guage</u> for the <u>simulation</u> of continuous-time systems and for the <u>solution</u> of differential equations. SL-1 is a superset of the CSSL language. SL-1 provides the user with interactive debugging features and a powerful set of conditional translation features.

<u>FMPS</u> is a comprehensive <u>mathematical pro- .</u> <u>gramming system</u> which is used in the solution of a variety of business and scientific problems; e.g., scheduling, allocation, blending, inventory control, transportation, and math simulations. FMPS operates in a linear or non-linear programming mode.

SL-1

Simplified, Problem - Oriented Simulation Language

Six Integration Algorithms

Facilitates Programming Tasks

FMPS

Gamma III System

Report Writer

Matrix Generator

Separable Programming

Parametric Programming

NOTES

CHART #10   Originated by: Marketing Technology

UTS utility processors are numerous.

The many utility programs provided under UTS
enhance the computer center's operation. These
routines simplify the mundane tasks common to
computer operations; e.g., file editing, card to
tape, tape to print, and many others.

Utility Programs

Delta

FDP

Edit

PCL

Sort/Merge

1400 Simulator

NOTES

CHART #11   Originated by: Marketing Technology

DELTA is very powerful for debugging at the assembly and machine language level. It can be used directly as a processor or indirectly by executing with the debug option. It can be used for modification of programs, insertion of breakpoints, tracing execution, and searching for specific elements in data and programs. Programs may be interrupted during the course of execution for debugging purposes.

Three types of breakpoint control are available: data, execution, and transfer.

Delta

On-Line Debugging Aid

- Program Modification

- Breakpoint Insertion

- Execution Trace

- Element Search

NOTES

CHART #12   Originated by: Marketing Technology

FDP is the most powerful FORTRAN debugger available today. For use in the batch as well as the on-line mode, it operates on programs at the FORTRAN language level. DELTA may also be used to debug FORTRAN programs at the machine language level. In FDP, reference can be made to statement numbers, statement labels, and data elements. Statement and data breakpoints can be inserted so that at preselected event occurrences various options are made available (e.g., stored commands which modify statements or data, display data values, branch trace, etc.). Direct commands are also available at breakpoints, allowing for manual modification and examination.

Dynamic control allows the user to trace the history of execution along with the current status of the program, as well as direct the course of future operations via stored commands.

FDP

- **Fortran Level Debugging**

- **Stored/Direct Commands**

- **Breakpoint Operations**

- **Dynamic Control**

NOTES

CHART #13 Originated by: Marketing Technology

EDIT is very powerful. It allows numerous file, record, and intra-record operations. Files may be built, copied, modified, and deleted. Intra-record operations permit characters and strings of characters to be substituted, overwritten, shifted, and inserted. Substitution of one string by another within a selective set of records is a unique feature.

EDIT may be used identically in either the batch or the on-line mode. File compatibility allows source programs and jobs to be created on-line for insertion into the batch job stream.

Edit

Files & Records

- Build
- Copy
- Delete
- Insert
- Renumber
- Move

Characters

- Shift
- Insert
- Substitute
- Overwrite
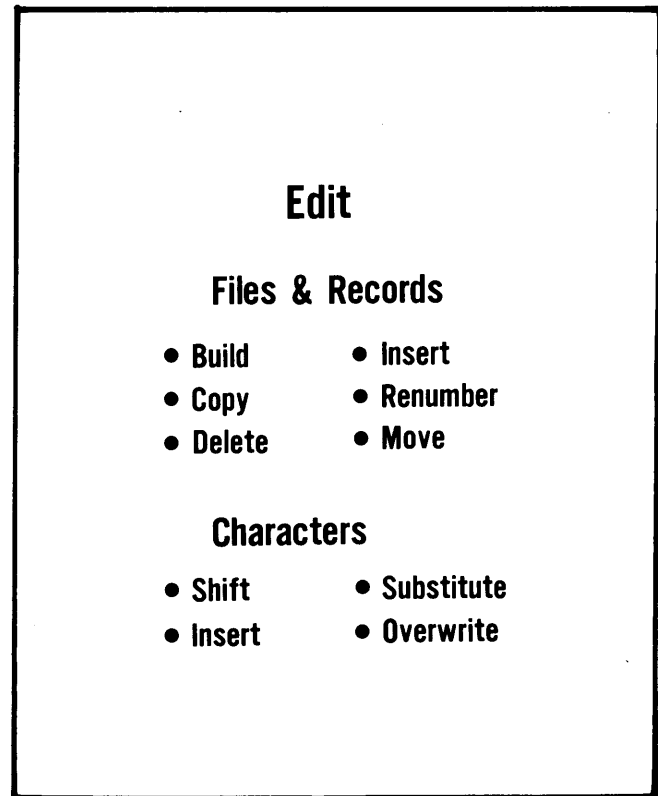
NOTES

CHART #14   Originated by: Marketing Technology

PCL is a powerful subsystem used for operating on files and selected peripheral devices. Files may be copied from one device to another (e.g., disk pack to line printer). During the copy, records may be selected, sequenced, formatted, and converted. PCL allows EBCDIC, Hollerith, and ASCII data codes and BCD, binary, packed, and unpacked binary modes to be specified. Tape operations such as write end-of-file, rewind, space, and remove are offered.

PCL is available to both the on-line and the batch user.

PCL

- **Media Conversion**

- **Tape Handling**

- **File Operations**

    Formatting

    Transfer

    Deletion

    Dump

NOTES

CHART #15   Originated by: Marketing Technology

The SORT/MERGE package provides the user with a fast, highly efficient, device-independent method of sequencing a non-ordered file. SORT may be called as a subroutine from within a user's program, a standard processor, or a batch processing job by control cards. It provides sorting capability for users with magnetic tapes, removable disks, and/or RAD's that contain Sigma 9 or foreign structured files.

Sort/Merge

- **Device Independence**

- **Tape, Direct Access or Both**

- **Subroutine Callable**

NOTES

In summary, the major features of UTS are as
follows:

- Three modes of concurrent operation:
  batch, time-sharing, and real time
- Overall efficiencies inherent in compatible
  accounting, file management, language
  processors, and utility programs
- Unique capability to handle large numbers
  of concurrent users
- Excellent throughput and response due to a
  combination of hardware architecture and
  software, designed for each other and for
  multi-mode operation
- The capability of installation management
  to have control over a complex time-sharing
  system for the best utilization of its re-
  sources

(U)(T)(S)

Multimode

Compatible Features

Many Users

Throughput & Response

System Tuning

NOTES

CHART #24A   Originated by: Marketing Technology

NOTE: USE THIS SUMMARY CHART FOR
SIGMA 6 ONLY.

Major features of UTS are the two modes of
concurrent operation (batch and on-line capa-
bility); the excellent throughput and response
due to a combination of hardware architecture
and software both designed for each other and
for multi-mode operation; the ability of in-
stallation management to have adequate control
over a complex time-sharing system for the best
utilization of its resources; and the overall ef-
ficiencies inherent in compatible accounting,
file management, language processors, and
utility programs.

Multimode

Compatible Features

Throughput & Response

System Tuning

NOTES

4.0   THE SIGMA 9 COMPUTER

Sigma 9, an XDS computer, represents a totally integrated combination of high-performance hardware and efficient multi-use software. It was particularly designed as a multi-use system with time-sharing, local and remote batch, and real time capabilities. Sigma 9 highlights true monolithic integrated circuits and completely modular expansion of capacity, speed, and enhancements.

The Sigma 9 computer system was constructed from a real time control viewpoint and enveloped in a time-sharing architecture. Thus, Sigma 9 may be assigned to several processing tasks simultaneously, in an interrupting environment, while still maintaining the integrity of high priority, real time processes. Its capabilities include basic processing tasks and multiprogramming, and may be expanded to include multiprocessing applications.

Sigma 9 is a multi-dimensional system, designed to operate efficiently in the following types of environments:

- Time-sharing
- High speed, input/output oriented (one or more very high speed I/O operations and/or multiple low speed I/O operations, all taking place simultaneously and independent of the CPU)
- General purpose scientific (computation-oriented)
- Business oriented
- Real time
- Mixed applications (combining several or all of the above environments)

4.1   General Characteristics

The particular features and characteristics which enable Sigma 9 to operate with unprecedented efficiency in transaction processing, real time, time-sharing, and multi-dimensional applications, are summarized on the following pages.

- Sigma 9 is word-oriented (32 bits plus parity) for maximum efficiency, but it is also addressable by 8-bit bytes, halfwords, and doublewords. Using 8-bit bytes, it is compatible with IBM System/360 and System/370 and modern communication codes such as EBCDIC and ASCII

- Sigma 9 incorporates memory protection (locks and keys) and program address protection features. For operation protection, the system has privileged instruction logic (Master/Slave/Master Protected modes)

- Memory mapping is available to provide for automatic dynamic relocation of programs and to control memory fragmentation

- The basic Sigma 9 configuration has two blocks of 16 general purpose registers, expandable to four blocks totaling 64 registers

- Sigma 9 features immediate addressing of operands. Indirect addressing may also be used, with or without post-indexing

- Up to four interval timers are available, three with choice of time base

- Internal traps (some maskable) are provided for error conditions

- Sigma 9 has a powerful instruction set with 111 <u>major</u> instructions. These operate on bytes, halfwords, words, doublewords, single registers, multiple registers, etc. Sigma 9 features (1) fixed point arithmetic; (2) double precision floating point hardware; (3) logical, shift, byte-string, and compare operations; (4) decimal hardware; push-down stacks; (5) automatic conversion instructions; and (6) execute, branch, control, interpret, and analyze instructions. The I/O instructions permit read/write direct of a full word, under program control, as well as full control and communication with input/output processors and their associated peripheral devices

- The real time interrupt system emphasizes extremely fast response time and interruption of instructions with long execution times. Sigma 9 recognizes and acts upon the highest priority interrupt allowed in less than 40 microseconds

- Up to 11 I/O processors are available for standard speed peripherals

- A new high speed RAD I/O processor is available for the high speed RAD's; up to four high speed RAD's may be connected to each high speed RAD IOP with instantaneous

transfer rates of 3.0 million bytes per second

. A wide range of standard and special peripheral devices is available

The Sigma 9 system consists of the central processing unit, one or more core memories, and one or more input/output processors (IOP's), each of which may have one or more subchannels, device controllers, and I/O devices. A typical system hierarchy is shown on the following page.
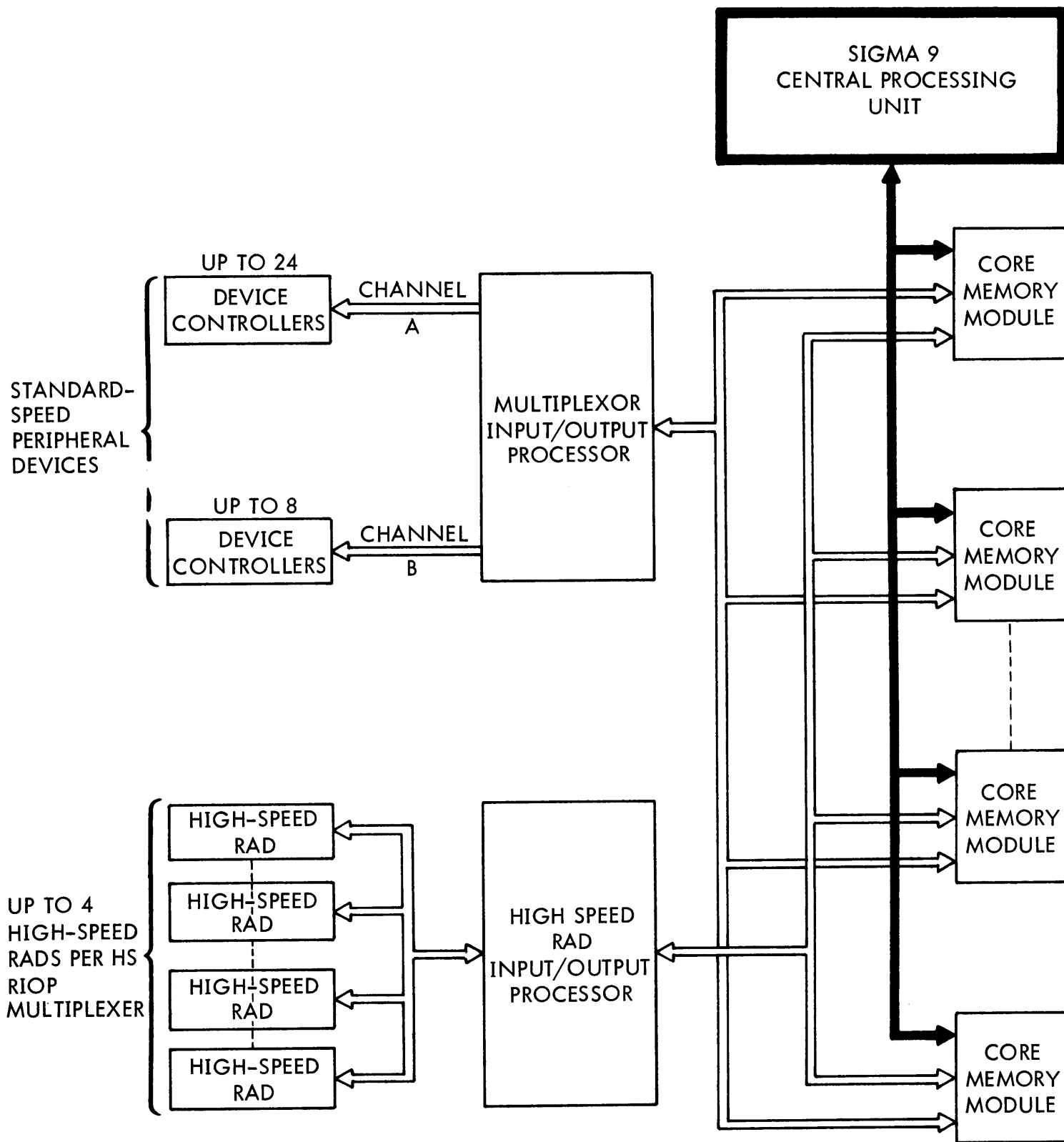
The Sigma 9 CPU is used to address core memory, to fetch and store information, to perform arithmetic and logical operations, to sequence and monitor instruction execution, and to control the exchange of information between core memory and other portions of the system.

Input/output operations are primarily under control of one or more IOP's. This allows the CPU to concentrate on program execution, free from the time-consuming details of I/O operations. Any I/O event that requires CPU intervention is brought to its attention by means of the interrupt system. Normal control and sequencing of the I/O system is handled by the IOP without attention from the CPU.

4.2 Memory Organization

The concepts of universal use, time-sharing, and multiprogramming all contribute to the need for large main computer memories. Sophisticated and increasingly complex computer applications require substantial amounts of main memory for storage of programs and their associated data.

For the Sigma 9 computer, a memory bank is defined as the smallest section of memory which may be independently accessed by a processor. This means that on Sigma 9 a memory bank is

Typical Sigma 9 Configuration

SIGMA 9
CENTRAL PROCESSING
UNIT

UP TO 24
DEVICE
CONTROLLERS

CHANNEL
A

STANDARD-
SPEED
PERIPHERAL
DEVICES

MULTIPLEXOR
INPUT/OUTPUT
PROCESSOR

UP TO 8
DEVICE
CONTROLLERS

CHANNEL
B

CORE
MEMORY
MODULE

CORE
MEMORY
MODULE

CORE
MEMORY
MODULE

CORE
MEMORY
MODULE

HIGH-SPEED
RAD

HIGH-SPEED
RAD

UP TO 4
HIGH-SPEED
RADS PER HS
RIOP
MULTIPLEXER

HIGH-SPEED
RAD

HIGH-SPEED
RAD

HIGH SPEED
RAD
INPUT/OUTPUT
PROCESSOR

16K words. Two banks of 16K words that share port logic are called a memory unit. This arrangement of memory banks and units provide the Sigma 9 memories with two-way interleaving between 16K word banks and four-way interleaving between 32K word units.

Sigma core memory systems use a 33-bit word (four 8-bit bytes plus a parity bit) as the basic unit of information. All memory is directly addressable by both the CPU and the IOP. Since the interchange of data between memories and processors is asynchronous, faster memories can be used as they become available. The speed of a computer system's memory influences both the rate at which instructions are executed and the rate at which input/output data transmissions can be accomplished. The Sigma 9 main magnetic core memory has a cycle time of 900 nsec. Since 32 bits may be read or written on each memory cycle, each memory bank can process over 35 million bits of information per second. Effective data transfer time is commensurate with both the speed of the central processor and the demands for high data transfer rate of real time computation and simultaneous peripheral input/output operations.

Most computer systems generally employ a fixed time relation between all functional elements, where all operations proceed in synchronism with a central timing source. Sigma systems use

asynchronous operation, where the individual functional elements operate on a proceed or continue-when-ready basis. Because Sigma 9 employs asynchronous memory design, a memory cycle may be initiated at any time, yielding a maximum rate of over 1,110,000 cycles per second in each memory bank. Because the system can incorporate multiple memory banks, more than one memory cycle may occur at a given instant; in fact, all memory banks on a system may cycle simultaneously. Further, multiple ports permit the simultaneous access of different memories by different processors. Asynchronous operation of memory banks provides over eight million accesses per second to a 128K word memory. This bandwidth capability is one of the outstanding features of Sigma architecture incorporated in the Sigma 9.

Each core memory in a Sigma system contains two memory ports as standard, each port capable of connection to a separate memory bus. Because a memory module services only one requesting unit (such as the CPU or an IOP) during a single read/write cycle, one port has priority over the other in the event of simultaneous read/write requests; otherwise, the access requests are serviced on a "first come, first served" basis. Optional ports may be added, with a maximum of 12 ports per memory bank. All input busses to memory have built-in priority relationships to resolve simultaneous read/write requests. Thus, each core memory module may be connected to as many as 12 memory busses, with access priority determined by the port to which a particular memory bus is connected. Low numbered ports have higher priority than high numbered ports. Port 1 has the highest priority; port 12 has the lowest priority. All ports on a given memory unit will address that unit with the same starting address.

It is sometimes necessary to prevent any single requesting unit from monopolizing a particular memory module for an extended period of time, even though the requesting unit may have the highest access priority (lowest numbered port) for the memory module. This is accomplished by allowing each port to have two priority levels, the normal priority and a high priority. A processor would normally request the normal priority level; however, under certain circumstances a processor may request high priority access to a given port. In the event one port receives a high priority request, its priority is then higher than the normal priority of all other ports no matter what its position in the normal priority chain might have been. For example, port "N", which would normally have the lowest priority, would be higher in priority than all other ports

when a high priority request is made to it. In the event more than one port is on high priority at the same time, the normal sequence of priority then prevails.

4.3   Instruction Format

Sigma 9 demonstrates that all of the benefits common to a flexible instruction set may be retained without undue complexity or a proliferation of instruction formats and lengths. Sigma 9 provides 111 major instructions, from which thousands of different operations result, all within the single instruction format.

The basic instruction is 32 bits long, exactly the amount of data provided by a single memory access.

I	0	R	X	M

0 1   7 8   11 12 14 15   31
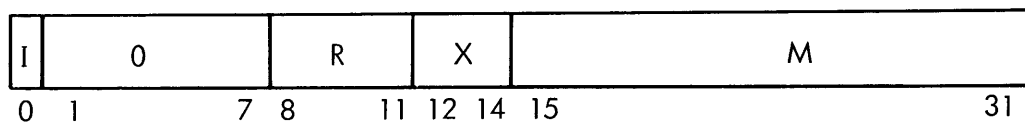
where:

I   =   Indirect Address (1 bit)

O   =   Operation Code (7 bits)

R   =   Register to be used (4 bits)

X   =   Index Register (3 bits)

M   =   Memory Address (17 bits)

In the Real Extended Addressing mode of operation, the instruction format has the following form:

I	0	R	X	E	M

0 1   7 8   11 12 14 15 16   31
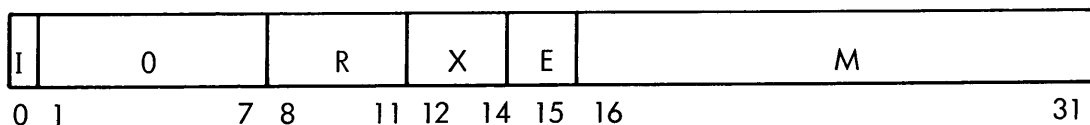
where I, O, R, X, and M are as described above and E is a one-bit Extension Selector Flag.

In a few, but operationally very important instructions, the X and M fields are combined into a single _value_ which is used immediately for computation with no reference to memory for an operand. These are called "immediate instructions".

Simplicity and power are inherent in this single format, a format which tremendously simplifies the internal workings of programs. The space that would be occupied to deal with multiple instruction formats may be used to reduce the size of the assemblers, compilers, and operational programs, as well as to perform other functions.

4.4   Types of Addressing

The Sigma 9 has three major types of addressing: Real Addressing, Virtual Addressing, and Real Extended Addressing. In Real Addressing, all addresses are 17-bit real word addresses. The Real Addressing type is identical to the unmapped type operation on Sigma 7. In Virtual Addressing, the eight most significant bits of the Virtual Address are transformed by the map to a 13-bit quantity which, along with the low order nine bits of the original 17, form a 22-bit real memory address. Since the Sigma 7 has a maximum of 256 pages of real memory, Sigma 9 compatibility with Sigma 7 is provided by the way the map is loaded. In Real Extended Addressing, the instruction address, all reference addresses, all indirect addresses, and all addresses appearing in places other than an instruction are considered to be 22 bits for word addressing.

The actual operation of each type of addressing is described below.

4.4.1   Real Addressing

Real Addressing occurs whenever the memory map is not used and whenever the extended type of addressing is not invoked. This is identical to the unmapped type of operation on the Sigma 7. All addresses are 17-bit word addresses. The high order 5 bits of the full 22-bit memory address are 0 at all times. If Sigma 9 has more than 128K of real memory, only the first 128K words may be referenced with this type of addressing.

4.4.2 Virtual Addressing

Virtual Addressing is invoked whenever the memory map indicator is set (PSD 9=1). When the memory map indicator is set, the mode altered indicator (PSD 40) has no effect on addressing. This type of addressing may be invoked in either the master or slave modes of operation. The use of the memory map on Sigma 9 allows the virtual memory, as viewed by a given user, to be divorced from the real memory and its organization. On Sigma 9, as on Sigma 7, the size of virtual memory is 128K words. The real memory may be up to 4 million words on Sigma 9. In operation, it is identical to the mapped type of addressing on Sigma 7 with the following differences: In translating a virtual address to a real address, the most significant eight bits of the 17-bit virtual address are sent to the map. These eight bits are used to select one of the 256 elements of the map. The contents of that element are then used in conjunction with the low order nine bits of the original 17-bit Virtual Address in order to form a real memory address. On Sigma 9, the contents of each element of the map is 13 bits and, therefore, the resultant address is 22 bits in length for word addressing. This provides the capability for addressing 4 million words of memory.

The compatibility with Sigma 7 operation takes place in the manner in which the map is loaded. When it is loaded by a Sigma 7 program, using only those instructions known to a Sigma 7 program, the lower eight bits of each 13-bit element will be loaded with significant information and the upper five bits will be set to zero. Thereafter, when address translation takes place through the map, the 8-bit virtual page number will result in a real page number, having only eight significant bits, being used for the final real address. A Sigma 9 program which wishes to take advantage of the large memory can load the map with up to 13 significant bits instead of the eight used by Sigma 7.

The full memory of Sigma 9 may be as large as 1024 pages. The virtual memory for any given user at any given time may be up to 256 pages selected from the 1K available.

4.4.3   Real Extended Addressing

Real Extended Addressing is a new type of addressing for the Sigma line of computers. It is in-
voked whenever the memory map is not used (PSD 9=0) and the mode altered indicator is on
(PSD 40=1). Real Extended Addressing is provided on Sigma 9 in order to ease the burden of
operating with a large memory for certain types of programs. Specifically, it is intended for the
operating system when that system is manipulating users directly, and is working with the real
and not the virtual memory of any given user. It also provides a method for writing programs,
real time or other special purpose, which may operate any place within the memory continuum
and not be dependent upon the map for their execution.

The instruction address, all reference addresses, all indirect addresses, and all addresses appear-
ing in places other than an instruction are then considered to be 22 bits for word addressing. Also,
in this type of addressing, the index displacement values are correspondingly longer; the word
displacement is 22 bits, a doubleword displacement is 21 bits; byte displacement is 24 bits, and
a halfword displacement is 23 bits.

A new field is added to the program Status Doubleword called the Extension Address. It occupies
positions 42 to 47 of the PSD. The extensions address is used to supply the extra bits needed to
extend the instruction address and reference address to 22 bits. The following address fields for
other operations are also extended to address the entire Sigma 9 real memory by extending the
address field into adjacent fields that were previously undefined:

   . Indirect addresses are interpreted as either full 22-bit addresses or as 17-bit addresses.
     The proper interpretation depends upon the condition of a flag in bit 0 of the indirect
     word itself
   . For byte string instructions, the source byte address and the destination byte address are
     extended to a full 24-bit byte address
   . The stack pointer is extended to a full 22-bit word address for the top of stack address
   . The value of the return location stored by a BRANCH AND LINK instruction is now a
     22-bit full word address

4.5 Operand Reference

There are a number of methods by which a Sigma 9 instruction word may specify where an operand is to be found. These methods are explained in the following subsections.
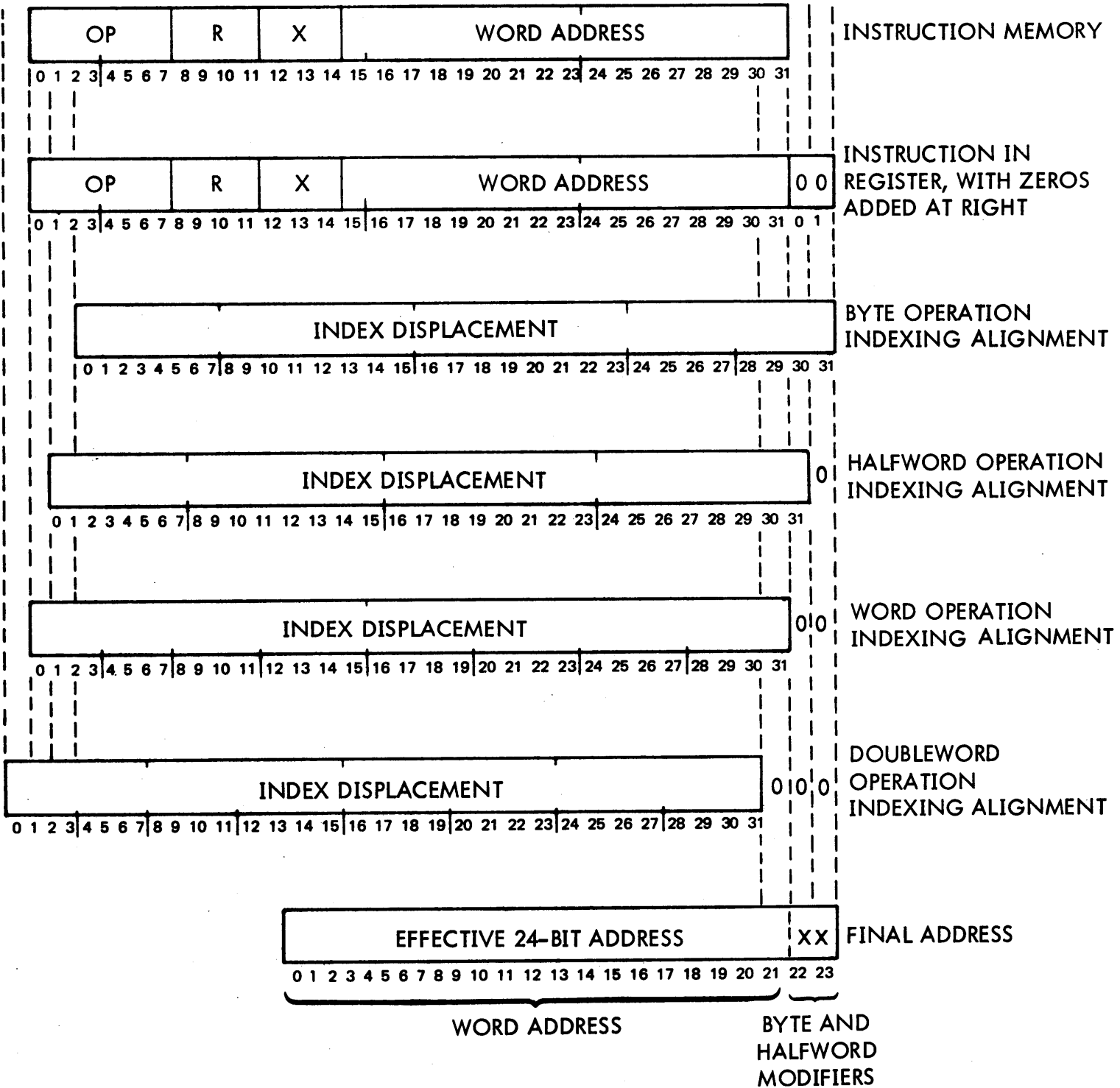
4.5.1 Register Address

Register addressing occurs whenever an instruction produces a direct, indirect, or index program address in the range from 0 through 15 (decimal). The CPU then does not attempt to read from or write into main memory. Instead, the general register (from the current block) with the number that corresponds to this address is used as the operand location. This feature permits the full power of the basic instruction set to be applied in register-to-register operations.

4.5.2 Immediate Address

The operation code of an immediate address type of instruction specifies that an operand is to be found in bit positions 12 through 31 of the instruction word itself, rather than in the core memory or a general register. This permits shorter, faster programs. The address field of this type of instruction is not modifiable by indirect addressing or indexing.

4.5.3 Indexed Addressing

The indexing technique employed in Sigma 9 is unique. Sigma instructions provide for operations on bytes, halfwords, words, and doublewords. These units of information are typically organized in lists which are processed sequentially. The Sigma indexing technique is based upon the concept that the index register should contain an integer value, say K, which permits the Kth item of a list to be accessed, independent of the kind of data in the list. That is, a byte instruction which is indexed would access the Kth byte; a halfword instruction which refers to the same index register would obtain the Kth halfword; word or doubleword instructions referring to the same index register would access the Kth word or doubleword. The figure on the following page shows how the indexing operation takes place.

| OP | R | X | WORD ADDRESS | | INSTRUCTION MEMORY |

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

| OP | R | X | WORD ADDRESS | 0 0 | INSTRUCTION IN REGISTER, WITH ZEROS ADDED AT RIGHT |

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 0 1

INDEX DISPLACEMENT — BYTE OPERATION INDEXING ALIGNMENT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

INDEX DISPLACEMENT 0 — HALFWORD OPERATION INDEXING ALIGNMENT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

INDEX DISPLACEMENT 0 0 — WORD OPERATION INDEXING ALIGNMENT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

INDEX DISPLACEMENT 0 0 0 — DOUBLEWORD OPERATION INDEXING ALIGNMENT

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

EFFECTIVE 24-BIT ADDRESS X X — FINAL ADDRESS

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23

WORD ADDRESS   BYTE AND HALFWORD MODIFIERS

Sigma 9 Indexing

If indexing is called for by the instruction (a non-zero value in bit positions 12 through 14 of the instruction word), the first-level address (or the indirect address if indirect addressing is used) is modified by the contents of the general register (index) called for by the instruction. This final address value (after indirect addressing and/or indexing) is defined as the effective address of an operand.

As the instruction is brought from memory, it is loaded into a 34-bit register which contains two zeros at its low-order end. The displacement from the index register is then properly aligned so that it is treated as an integer relative to the type of operand specified by the OP code. If it is a byte type operation, the displacement is lined up so that its low-order bit is below the right-most of the two low-order zeros in the instruction register. If it is a halfword operand, the displacement is shifted one bit to the left of this position (two bits to the left for a word address, and three bits to the left for a doubleword address). The indexing operation then takes place to develop the final 19-bit address (24 bits for real extended addressing) that is delivered to memory. Higher-order bits of the 32-bit displacement field are ignored in the development of this final address.

4.5.4   Indirect Addressing

Indirect addressing reduces the time and overhead required to transmit data among programs. Furthermore, it permits simple, rapid specification of data address among programs.

If the instruction calls for indirect addressing (a "one" in bit 0 of the instruction word), the word address field is used to access a memory location containing another address. The initial value of this second-level address replaces the first-level address, and is defined as the indirect address. If no indexing is called for by the instruction, the indirect address is the effective address of an operand.
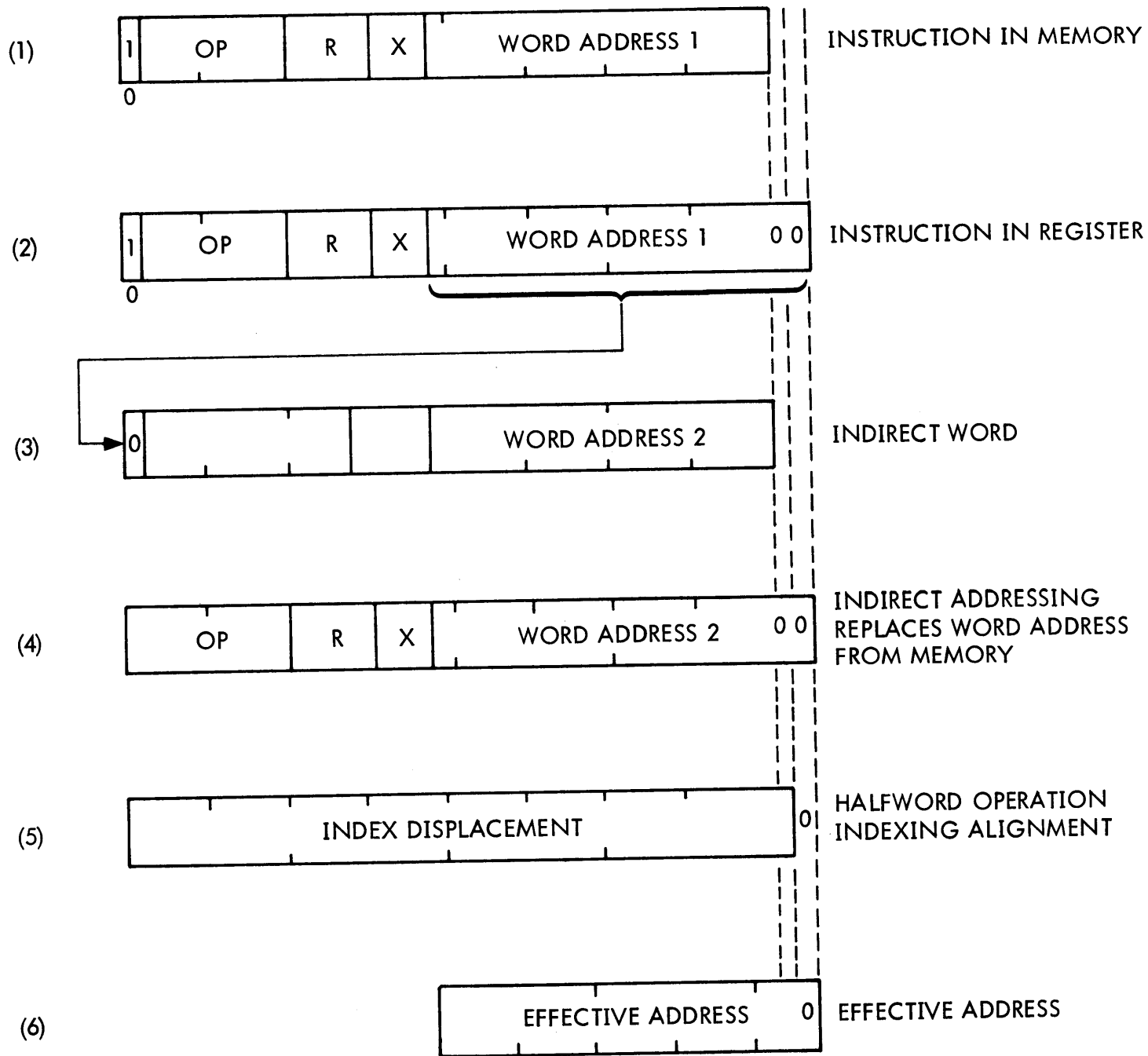
If both indexing and indirect addressing are called for, indexing occurs after indirect addressing. In this operation, called post-indexing, the 17-bit word address of the instruction is used to obtain a word from memory. The 17-bit word address portion of the newly obtained word then becomes the word-address operand, used to obtain the final address. Essentially, it

replaces the instruction word address in the instruction register, and then indexing is carried out in accordance with the previous discussion, dependent on the OP code of the instruction for the alignment of the index register. The figure on the following page illustrates this process for an indirectly addressed, indexed, halfword operation in the real addressing mode. As the figure shows, Word Address 1 is the content of the 17-bit address field in the instruction, as stored in memory. The 17-bit word address 1 on line (2) points to a memory location that contains Word Address 2. Word Address 2 is moved into the instruction register and replaces Word Address 1. The displacement portion of the index register referenced in the instruction is then aligned for incrementing at the halfword level and the effective address is formed. This final 19-bit address, which in this case automatically contains a low-order zero, is then used to access the halfword in memory that is used as the operand for the operation required by the OP code.

Indirect addressing in the Real Extended Mode is handled differently. (Reference figure on following page.) The indirect word is interpreted in two ways, depending upon the condition of the flag in bit 0 of the indirect word itself. If the flag is 0, then the 22 bits from positions 10 through 31 are treated as the 22-bit address of a specific word in the entire memory range. When bit 0 of the indirect word is a 1, then the word is interpreted as having a 17-bit address treated identical to the instruction address in the PSD or the address in an instruction word. That is, the 17-bit address is treated as a 1-bit extension selector flag in position 15 and 16-bit region address in positions 16 through 31. If the extension selector flag in bit 15 is a 0, then the address is in the 0 region. If the extension selector flag is 1, then the address is in the program region (vis-a-vis, the full 22-bit address is generated by concatenating the extension register to the low order 16 bits of the indirect word).

4.6   Register Organization

The use of multiple, multi-purpose registers reduces the number of instructions required for any system and speeds processing. In addition, multi-purpose registers increase indexing flexibility and speed of computation. Commonality of arithmetic and index registers permits address calculations to occur directly in the index registers, saving time and instruction storage. Furthermore, the availability of multiple register blocks offers a tremendous increase in the speed of interrupt response.

(1) | 1 | OP | R | X | WORD ADDRESS 1 | INSTRUCTION IN MEMORY

0

(2) | 1 | OP | R | X | WORD ADDRESS 1 | 0 0 | INSTRUCTION IN REGISTER

0

(3) | 0 | | | WORD ADDRESS 2 | INDIRECT WORD

(4) | OP | R | X | WORD ADDRESS 2 | 0 0 | INDIRECT ADDRESSING REPLACES WORD ADDRESS FROM MEMORY

(5) | INDEX DISPLACEMENT | 0 | HALFWORD OPERATION INDEXING ALIGNMENT

(6) | EFFECTIVE ADDRESS | 0 | EFFECTIVE ADDRESS

Indirect Addressing Sequence for an Indexed Halfword With Real Addressing

INSTRUCTION IN MEMORY

INSTRUCTION IN REGISTER

INDIRECT WORD GIVES 22 BIT
ADDRESS. WHEN BIT 0 IS 0

INSTRUCTION IN REGISTER

INDIRECT WORD

PSD_{42-47} INDIRECT $WORD_{16-31}$

6   +   16 = 22 BITS

ADDRESS FORMED FROM INDIRECT
WORD WHEN BITS 0 AND 15 OF THE
INDIRECT WORD ARE 1.
ALLOWS ADDRESSING OF ALL OF
MEMORY

INSTRUCTION IN REGISTER

ADDRESS FORMED FROM INDIRECT
WORD WHEN BIT 0 IS 1 AND BIT 15
IS 0 IN THE INDIRECT WORD.
ALLOWS ADDRESSING OF FIRST
64K OF MEMORY

16   16 BITS   31

Types of Indirect Addressing in Real Extended Addressing Mode
(No Indexing Shown)

Each Sigma 9 processing unit contains two groups (expandible to four) of 16 general-purpose 32-bit registers. These 16 registers (called a "register block") are reserved for the CPU alone and can be used as accumulators, index registers, or as temporary, "parking registers". Their use is best seen with respect to the following instruction format.
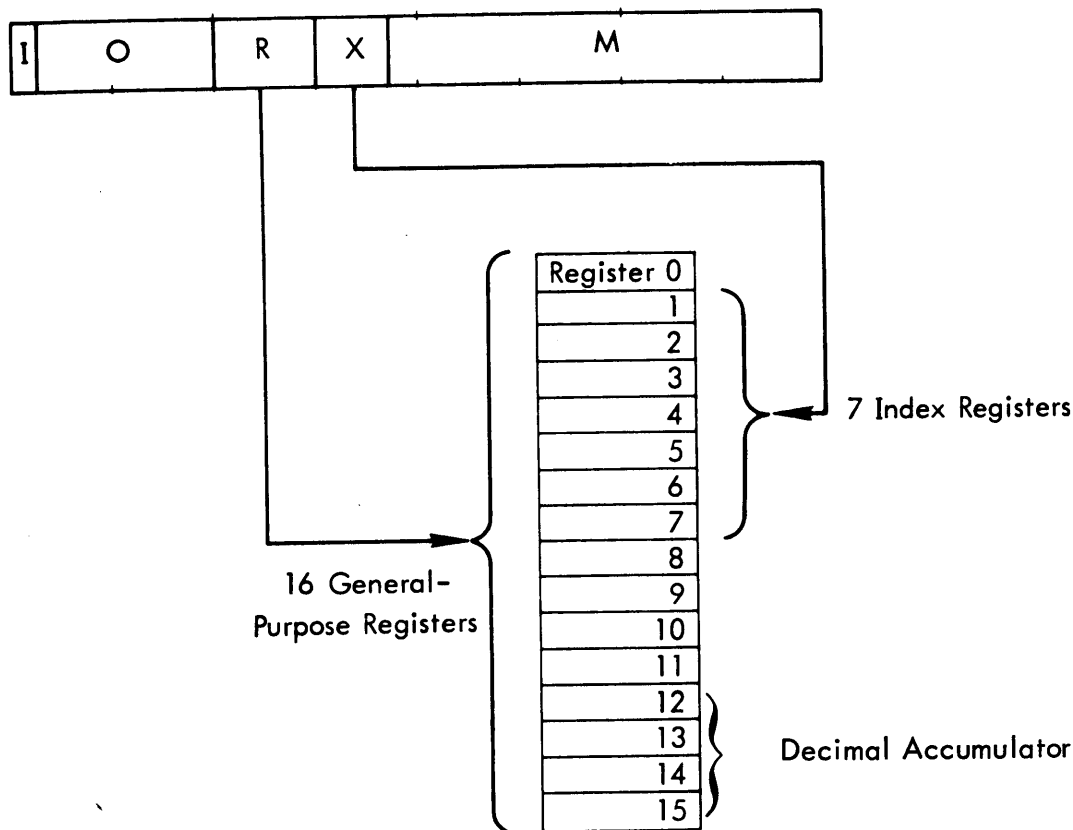
Sigma 9 Register Utilization (Current Register Block)

If the X (or index) field of an instruction is not zero, then the corresponding register (1 through 7) is used as an index quantity. The R (or register) field of an instruction specifies which of the 16 registers (0 through 15) is to be employed (e.g., if R is 9 in a LOAD instruction, the operand is loaded into register number 9). Registers 12 through 15 are used in optional decimal operations. They are all, of course, available for use in other instructions (except as index registers).

A Sigma 9 register block can provide:

. Sixteen separate single precision arithmetic registers (fixed or floating)

. Eight separate double precision arithmetic registers, including double floating point

. Seven separate index registers

. A variable length decimal accumulator, with a maximum capacity of 31 digits plus sign

Other significant effects of the register block approach include the following:

. Arithmetic calculations may be performed in the index registers, enhancing subscript calculation and accessing of multi-dimensional or variable length data arrays

. There is no distinction between fixed point and floating point working registers. Mixed mode arithmetic, floating-to-fixed, and fixed-to-floating conversions are handled easily, without intervening load/store instructions

. For double precision operations, two registers may be treated as a single, 64-bit, double-word register with the most significant word in the even-numbered register (specified by the R field of the instruction) and the least significant word in the next higher odd-numbered register

. XDS automatic programming aids (such as Extended XDS FORTRAN IV) optimize the assignment and use of these registers, both within themselves and for the object programs produced. This results in programs that are faster to compile and execute than those on similar size computers

Sigma 9 design satisfies the need for rapid switching of control among programs to provide the time sensitivity and responsiveness needed by real time and input/output programs. Program Status Doublewords are used to preserve the existing machine state and to establish another state. In general, programs that are initiated require the use of mainframe registers during their execution. To insure the rapid swapping of programs, Sigma 9 has special instructions to store the active registers and load them at the speed of memory. About 45 µsec are required to store all 16 registers in memory and to reload them with 16 words from memory.

Even this small register swapping time may be eliminated if additional optional register blocks are used. The multiple register-block feature permits a completely new set of registers to be

invoked as well as preserving the formerly active register block, all in 6 μsec. The Program Status Doubleword contains a four-bit field, called the register pointer, whose value selects one of up to four separate register blocks, and invokes the selected blocks as "active" for the current program or instruction. This is shown in the following figure.
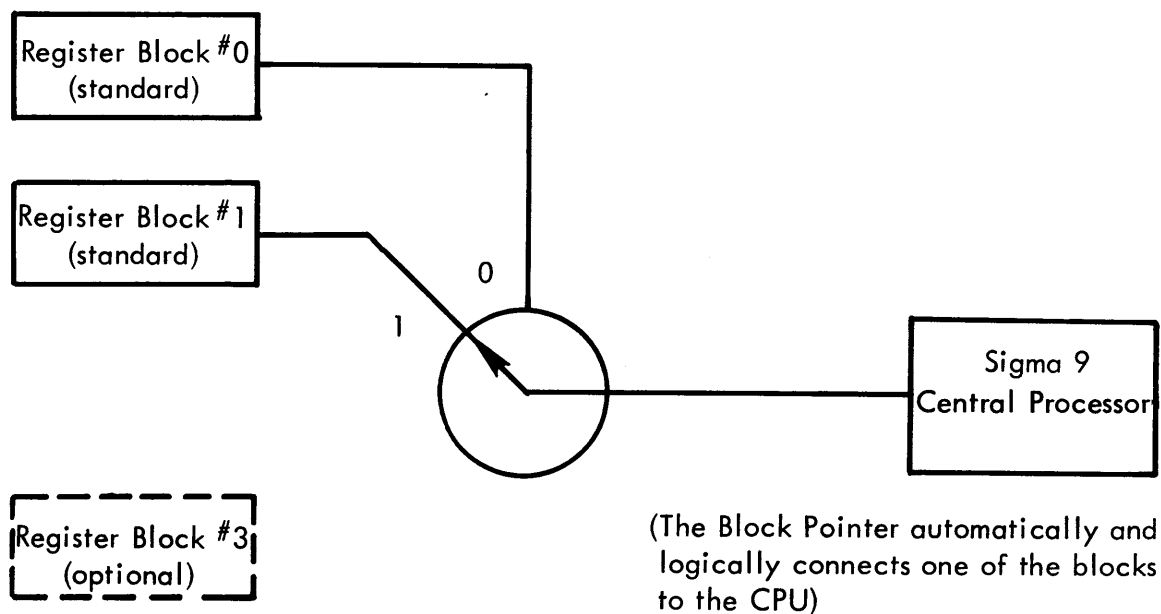
Block Pointer and Register Selection (shown with a Block Pointer Value of 1 (0001))

The pointer is changed whenever an XPSD (Exchange Program Status Doubleword), LPSD (Load Program Status Doubleword), or LRP (Load Register Pointer) instruction is executed. Thus, program control may be swapped without the need to load or store the contents of the fast registers. This enhances real time responsiveness and reduces the overhead usually associated with interrupt and scheduled servicing.

The special Sigma 9 dual addressing feature permits the multiple, general purpose processing registers to be used in the same manner as memory; namely, as the source or destination of a result. To circumvent the need for special instruction formats for register-to-register operations, and simultaneously to permit all memory-to-register instructions to serve as register-to-register instructions, Sigma 9 utilizes the memory address to specify either the core storage location (addresses from 16 to 524,288) or one of the 16 currently active registers (0 through 15).

4.7 Sigma 9 Instructions

The instruction set for Sigma 9 is both an enhancement and extension of the Sigma 7 instructions. Compatibility has been maintained between Sigma 7 and Sigma 9. A Sigma 7 user program (e.g., FORTRAN) will run unmodified on Sigma 9. However, many of the Sigma 7 instructions have been modified to enable them to take advantage of Sigma 9 features not found on Sigma 7. Other Sigma 9 features required totally new instructions. Two of the new instructions are Load Memory Status (LMS) and Load Real Address (LRA). LMS allows you to interrogate the status of a memory bank and to perform diagnostic functions in a memory bank. LRA provides the programmer with a means of determining the real effective address of an operand used in an instruction.

The Sigma 9 instruction set provides I/O-oriented processing efficiency, with byte-string manipulations, absolute and complement loads, selective and multiple register operations and many other significant features, some of which are described in the remainder of this section.

A full complement of shift instructions permits left and right shifts; single and double precision; and logical, arithmetic, end-off, searching, and circular shifts, as well as a floating point shift capability. The OR, AND, and EOR logical instructions provide efficient logical information manipulation. The "Modify and Test" instructions for bytes, halfwords, and words provide efficient means to modify or test operands in memory. The various types of instructions that may be used with Sigma 9 are described in the following paragraphs.

4.7.1 Byte Instructions

To accommodate the merger of diverse applications within a single mainframe, XDS has supplemented the binary, word parallel, primary operating characteristics of Sigma 9 with a powerful byte-manipulation capability. Sigma 9 instructions, for example, permit the system to perform the following operations:

- Load or store any of the possible 2,097,152 byte locations into memory or from a register
- Increment or decrement any byte in memory
- Compare or examine any byte in memory

. Compare two byte strings of arbitrary length

. Move a variable length byte string from one area of memory to another

. Translate from any 8-bit code to any other 8-bit code

. Test a byte or byte string for any or all of eight independent characteristics

Thus, all the basic byte manipulations normally found on character-oriented "business" processors exist in Sigma 9. The byte manipulation capability is vital to efficient operation of compilers, assembly, and input/output programs. Furthermore, Sigma 9 preserves the packing efficiency of main memory in data system applications which require data lengths of only modest precision; e.g., 16 bits.

Three special byte instructions permit Sigma 9 to store or change the contents of the condition code and the flag bits that control the execution of floating point arithmetic instructions. These instructions supplement the power of the Program Status Doubleword instructions and permit the rapid changes that are frequently encountered in programming, such as switching between normalized and un-normalized floating point computation.

4.7.2   Halfword Instructions

The concept of universal use which guided the design of the Sigma 9 required the inclusion of certain features to facilitate real time processes and control, and data acquisition applications. These applications are usually characterized by relatively small units of source and output data (16 bits is usually sufficient for I/O purposes). However, such applications often require internal computation with intermediate precision beyond the basic 16-bit input or output format.

Sigma 9 contains a powerful set of instructions to deal with these 16-bit quantities for input, output, comparison, and data movement. In addition, Sigma 9 automatically provides a convenient method for using 16-bit data in double length (or 32-bit) arithmetic operations. For example, accumulating a 32-bit sum from a set of 16-bit numbers requires no extra programming. Thus, memory is used efficiently and programs do not waste time packing or unpacking 16-bit fields and words. The simultaneous availability of full-word arithmetic involving these quantities guarantees sufficient precision for all computation tasks that involve 16-bit data.

4.7.3 Floating Point Instructions

Floating point arithmetic permits automatic scaling, so that factors with widely varying magnitudes may be calculated directly. In effect, floating point operations remove the programming tedium normally required to align factors properly for arithmetic purposes, and at the same time preserve a maximum degree of accuracy, or significance, for each result.

Sigma 9 contains a group of floating point instructions which permit basic or elementary arithmetic operations in either of two precisions. The short format can express a 7-digit fractional number over the range of $10^{\pm 79}$. For applications which require greater precision or long term significance, a long format yields over 16 decimal digits of precision in the fraction. The choice of format is left to the user; thus, shorter intermediate calculations may use single precision for speed and space economies and complex iterative programs can employ the long format (to preserve a maximum of significance).

The basic Sigma 9 floating point instructions are augmented to provide special kinds of operations and checking that are not commonly found on even large scale computers. For example, floating point addition or subtraction may be either normalized or un-normalized. To aid in significance tracing and conversion between fixed and floating point formats, special circuitry detects an impending or probable loss of significance. Used principally with the short format, this circuitry generates a program trap whenever more than two digits of significance are lost in the final result. This trap serves as a useful clue to potential problem areas and as a pointer to indicate that the long format could be used to increase significance.

Since floating point operations are handled in the primary 16 registers, and since a compatible two's complement format is used for both fixed and floating point numbers, the single or double length instructions for fixed point operations (e.g., compare) may also be used on floating point numbers, tremendously simplifying the programmer's task.

4.7.4 Decimal Instructions

Sigma 9 incorporates a group of instructions to manipulate decimal data and to implement the direct arithmetic operations on decimal numbers.

Sigma 9 uses a sign-magnitude, packed, decimal format, in which two digits or a digit and a sign are packed in each byte. The length of the operation is specified as a number of bytes. This means that operations are faster, since two digits are obtained during each byte access, and bits in memory are not wasted. Instructions which convert between the 8-bit code of peripheral devices and the "packed" 4-bit decimal format eliminate the complex programming that is otherwise necessary to accomplish this function.

The decimal field may be of varying length, from one to 31 digits plus sign. Thus, an overall decrease in problem solution time is realized, since only the exact number of digits required in a calculation need be processed. As with all long instructions on Sigma 9, the long decimal operations may be interrupted at any time. They can be resumed at the point of interruption when the interrupt routine is finished. Sigma 9 has the capability of generating either ASCII or EBCDIC zone and sign codes. The desired form, either ASCII or EBCDIC, is selected by a flag bit in the program status doubleword.

4.7.5 The EDIT Instruction

XDS scientists have found an almost identical incidence of formatting procedures in business, scientific, communication, and concurrent multiple-user applications. Therefore, to optimize all of these seemingly diverse, but really common types of processing, the basic Sigma system includes a versatile, powerful format-editing instruction.

The EDIT instruction, included with the Sigma 9 decimal operations, gives the scientific and real time user capabilities for output formatting which, in the past, were available only to the business equipment user. The instruction requires only one pass over the editing pattern and data to be inserted. Blanks, punctuation symbols, or arbitrary characters may be inserted in the output field along with edited data. The ability to bypass characters in the editing pattern enables punctuation and text to be preserved or inserted as necessary. In addition, the automatic

recording of specific memory addresses is offered to assist in the insertion of such special characters as the floating dollar sign. The Sigma 9 EDIT instruction permits decimal fields with either an even or odd number of digits to be processed.

4.7.6 Immediate Instructions

Many constants used in computer programs (such as memory addresses, increments, or decrements for index register control, data scaling constants, etc.) may be expressed in a number of bits substantially less than the computer's word length. Sigma 9 design permits constant and variable data of this type to be contained in or carried along with the instruction using it. These instructions are called "immediate" instructions. Because both the instruction and its operand are contained in one word and are accessed together, Sigma 9 programs are shorter and faster than similar codes written for other computers.

Sigma 9 immediate instructions have the following format:

OP	R	±	VALUE (two's complement if negative)
0 7	8 11	12 13	31
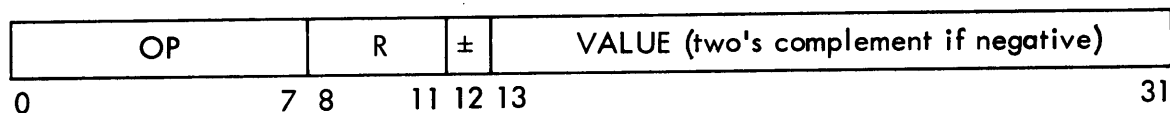
The R field has its usual meaning and specifies which of the 16 general purpose registers is to be used in the operation. A 20-bit value is contained within the instruction itself. This value may be used to state a constant in the range from -524,288 to +524,287. In use, the value is sign-extended to a full 32 bits and then treated as a full word operand.

XDS compilers use the immediate operand accessing feature extensively for address initialization, comparison of computed values with constants, and index register setup and control. Thus, they generate highly optimized codes which are shorter both in space and time than would otherwise be.

4.7.7 Push-Pull Instructions

Two very important concepts concerning the organization of computer programs have emerged from recent intensive studies of time-shared computer use. Each of these has significant bearing on machine design.

The first concept, called <u>re-entrance</u>, concerns the computer's ability to share a single sub-program among several programs and to re-enter at any point required. Re-entrance is a complex problem because control must be switched rapidly among the various main programs in the dynamic environments of time-sharing and real time operations.

<u>Recursion</u>, the ability to repeat a subroutine from its beginning, is closely allied with sophisticated compiling and automatic program generating techniques. Coupled with a compiler's ability to generate coding if and only if a specific set of conditions is satisfied, recursive programs can minimize both the space and execution time of many calculations.

Re-entrance and recursion share a common requirement to "push down" the results of on-going calculations so computation may proceed and, at a later time, to "pull up" previously stored results for use. The essential principle involves the storage of active registers and working space (or a pointer to working space, if indirect addressing techniques are used) and their subsequent re-activation on a last-in, first-out basis.

A number of instructions implement efficient push-down programming techniques. These instructions enable simple and efficient programming of recursive routines and eliminate the need to disable interrupts in order to accomplish housekeeping operations. Each of these instructions operate with respect to a memory stack (a reserved area of memory into which operands are pushed and from which they are pulled) whose top location is indicated by a stack point in memory.

The stack pointer points to the current top of the stack; the contents of that location are pulled from the stack on any pull instruction. When this happens, the stack pointer is decremented by one, so that it points to the new top of the stack. In a push operation, the stack pointer is first incremented by one, so that it points to the next location in the stack, and then the operand is pushed into that location and becomes the new top of the stack.

On every push or pull type of operation, counts are maintained of the number of words in the stack and the number of spaces remaining. If either of these counts indicates that the limits of

the stack are being exceeded, the condition code is appropriately set, the instruction is
aborted, and (unless the programmer has indicated otherwise), the computer traps to a stack
limit error routine.

This push-pull capability, which is incorporated into the basic Sigma 9 processor, also permits
dynamic storage allocation for time-sharing. Thus, programs can receive or release working
storage space as needed. Maximum areas need not be permanently dedicated to programs through-
out the course of their execution. Rather, working space may be assigned as needed, used, and
then released for use by other programs. Core memory is therefore allocated only when neces-
sary.

4.7.8   Bounded Limit Comparisons

In analyzing both real time process applications and general batch programs (including assembly
programs and compilers), XDS design engineers discovered a high incidence of instructions which
caused data to be checked against explicit high and low limits. Although a few older computer
designs have incorporated hardware to perform a high and low limit comparison in one instruc-
tion, the hardware was usually implemented in such a manner that the limits were placed in
working registers and the data in memory.

Investigating further, XDS designers found that in about half the cases, the comparand was the
result of an arithmetic calculation. In these cases, the comparand had to be stored from a reg-
ister, and registers then had to be loaded with the limits before the comparison could take place.
In the other cases, it was found to be more natural to use the limits in registers and the comparand
in memory (e.g., searching for a value which meets certain bounds).

Sigma 9 allows either method to be used. Thus, the total value of the limit comparison is pre-
served without programming complications. The "natural" choice between locating the comp-
arand in a register or in memory is left to the programmer or compiler, where it belongs.

One instruction is required to test against both the high and low limit. The result of the com-
parison indicates if the comparand is:

. Equal to the upper limit

. Greater than the upper limit

. Less than the upper limit

. Equal to the lower limit

. Greater than the lower limit

. Less than the lower limit

4.7.9 CALL's

The very first XDS computers contained special features called "Programmed Operators" to enhance the linkage and communication between a main program and frequently used system programs. These have been extended to provide a generalized mechanism for any type of centralized public service, including not only computation tasks but also such operations as I/O device control, format conversion, un-implemented instructions, etc. On Sigma 9, this feature has been given the name "CALL".

Sigma CALL's provide a highly efficient method of entering 64 special purpose subroutines in a completely natural manner, thus extending the basic Sigma instruction set by an additional 64 functions. Since only the operation code and register field of a Sigma instruction are used to enter a CALL, the index and address fields of the instruction are available to the interpreting program. The CALL may also be used on a dedicated basis for specialized real time applications which require privileged operation.

4.7.10 The Analyze Instruction

Every Sigma 9 computer has a special instruction to minimize address computation within subroutines, service routines, or CALL's. The Sigma 9 "Analyze" instruction is used to locate any word in memory and, interpreting it as an instruction, to calculate the address that would be generated if the instruction were to be executed. In other words, "Analyze" answers the question, "What address would be used by the instruction located at X if it were executed?" If the instruction to be analyzed is of the immediate type, then the value of the immediate operand (rather than an address) is calculated.

This instruction not only calculates an address and stores it in a programmer-designated working register, but also provides additional information for checking and verification purposes. This is true whether the analyzed instruction is of the "immediate" or "addressing" type, and whether the address is that of a byte, halfword, word, or doubleword.

4.7.11 Conversion Instructions

A few previous computers have had specialized conversion hardware, with little or no universality. Sigma 9 now permits a completely generalized bi-directional method of converting between any weighted code and binary, as well as converting from any 8-bit code to any other 8-bit code.
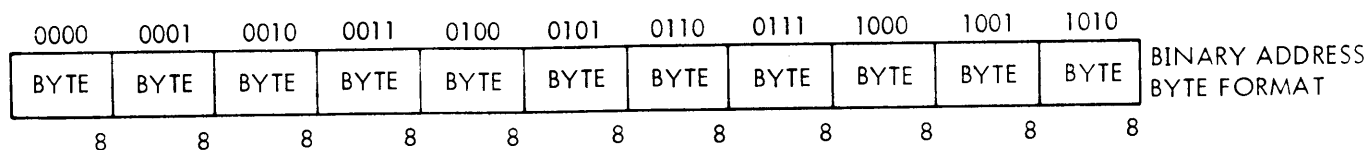
Two Sigma 9 instructions, "Convert by Addition" and "Convert by Subtraction", are used to accomplish bi-directional translation between binary code and any other weighted binary code such as BCD. Conversions are now simple, straightforward, and fast. Typically, a 4-bit BCD-to-binary conversion on a conventional computer requires eight multiplications and eight additions to convert eight decades into binary (neglecting any shifting or housekeeping, which could easily double the number of instructions required). Sigma 9 requires only one instruction for this purpose.

4.7.12 Data Word Formats

The 32-bit Sigma 9 word is treated in a number of data formats by the various instructions of the computer. The basic data formats that are used are shown in the figure on the following page.

4.8 System Environment and Control

To meet the many requirements of real time control and time-sharing, Sigma 9 was designed with many unprecedented control and protection features. These are described in the following subsections.

0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	BINARY ADDRESS
BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE	BYTE FORMAT

8   8   8   8   8   8   8   8   8   8   8

| ±1 | HALFWORD | ±1 | HALFWORD | ±1 | HALFWORD | ±1 | HALFWORD | ±1 | HALFWORD | | HALFWORD FORMAT |

16   16   16   16   16

| ±1 | WORD | ±1 | WORD | ±1 | WORD | WORD FORMAT |

32   32

| ±1 | DOUBLEWORD | ±1 | DOUBLEWORD | DOUBLEWORD FORMAT |

64

| ±1 | BIASED EXP. | 24-BIT FRACTION | ±1 | BIASED EXP. | 24-BIT FRACTION | ±1 | BIASED EXP. | 24-BIT FRACTION | FLOATING-POINT SHORT FORMAT |

32   32

| ±1 | BIASED EXP. | 56-BIT FRACTION | ±1 | BIASED EXP. | 56-BIT FRACTION | FLOATING-POINT LONG FORMAT |

64

| INSTRUCTIONS | ± | 20-BIT FIELD | INSTRUCTIONS | ± | 20-BIT FIELD | INSTRUCTIONS | ± | 20-BIT FIELD |

32   32

Integral Boundaries and Data Word Formats

4.8.1   Master/Slave/Master Protected – States and Privileged Operations

Certain Sigma instructions may be classified as critical with respect to real time, foreground, or other multi-programmed operations. These instructions are characterized by the fact that if they are executed (or improperly executed) by a program, other programs or hardware elements could be adversely affected.

Since the integrity of both programs and equipment external to the computer must be guaranteed inviolate in real time environments, the Master/Slave/Master Protected state feature of Sigma was designed to provide program sovereignty and to preclude catastrophic faults. All critical instructions are specially treated, depending upon whether the computer is in the Master, Slave, or Master Protected state. An active storage element, which can be altered only under certain well-defined and easily controlled circumstances, specifies the machine state.

In the Master state, all instructions are defined as "legal" and may be executed normally. In the Slave state, critical instructions are disallowed; any attempt to execute one of them is regarded as an error, and the Sigma system instantly takes appropriate remedial action. In the Master Protected state, privileged instructions may be executed while still maintaining access protection.

The critical instructions, known as "privileged operations", are allowed only in the Master or Master Protected states. These are the instructions which control:

- . Input/output operations
- . Read direct/write direct
- . Memory protection systems
- . The processor mode itself (i.e., a Slave program cannot directly and unilaterally make itself a Master program)
- . Cessation of computer operation

XDS provides efficient I/O programs which may be automatically appended to user programs, in addition to a set of control programs (monitors) of varying size and complexity to control I/O.

Almost all user programs may be run in the Slave state, with atuomatic reversion to the monitor in the Master state where I/O is required. This feature provides a very important benefit to even the conventional "batch" or "stand-alone" user who does not have real time or foreground operations; that is, programming errors and more easily are more comprehensively discovered and diagnosed. In batch processing on conventional computers, the destruction of a monitor or control program by an undebugged operating program is a common occurrence. This destruction usually requires operator intervention to restore the monitor. In many cases, this precludes a memory dump, complicating the fault isolation and diagnosis of the offending program. Use of the Master Protected/Master/Slave feature and other elements of Sigma 9's comprehensive protection system eliminates this problem.

4.8.2   Condition Code

The ability to make decisions and to select various alternate courses of action based upon these decisions is one of the distinguishing characteristics of stored-program digital computers. This process of dynamic decision-making implies that these computers must have a flexible method of recording decisions and using the recorded results to choose alternate program steps.

The condition code provides Sigma 9 programs with all necessary information regarding the outcome of decisions and the last executed decisional, operational, or data movement operation. The Sigma condition code feature does more than record simple yes/no or true/false answers to decisional instructions — it also providea a dynamic record of all principal computer operations. Depending upon the type of instruction executed, the code records such conditions as:

- Zero/non-zero result
- Positive/negative result
- Arithmetic overflow
- Arithmetic carry
- Normalization indication
- Number of steps performed
- Even/odd parity
- State of I/O devices

The condition code may be interrogated by two symmetric computer instructions: "Branch if Condition Set", and "Branch if Condition Reset". These decisions may be made instantaneously based upon the outcome of preceeding operations.

This setting of the condition code automatically provides two benefits. First, since the condition code is always set, special instructions are not required to record a result. Second, because all results which are useful in making decisions are recorded, special instructions (or operation codes) need not be assigned to test for the multitude of possible conditions that can arise.

Sigma 9 uses the 4-bit condition code in most load/store and arithmetic instructions. Two of these bits indicate whether the result was positive, negative, or zero; the other two independently indicate whether an overflow and/or a carry from the most significant bit position occurred. Sigma 9 allows the user to store the contents of the condition code and to retrieve them later, loading them back into the 4-bit condition code register for testing.

4.8.3   Program Status Doublewords

As computer speeds have increased and programming technology has matured, time-sharing and real time operations have become not only desirable but in many cases necessary adjuncts to conventional, batched operation. Even the simplest conventional systems use some type of monitor program which switches control among various peripheral I/O handlers and a main program, providing an overall control mechanism. As the number of operations (independent programs) increase, the number of transfers among the programs increases; thus, the complexity of the "switching" program grows also.

Whenever control is taken from one program and given to another, the state or environment of the first must be preserved so that computation may be later resumed at the point of interruption. In addition, any prior state of the second program must be reloaded so that computation may proceed properly, possibly at a point of prior interruption. For this type of operation to be economically feasible, and to provide fast response to external interrupts, this program switching must be accomplished as rapidly as possible.

Critical to this switching of control is the preservation of the machine state. The use of Program Status Doublewords (PSD's) enables Sigma 9 to change operating environment completely (to preserve the existing state and establish a new one) in less than 6 microseconds. A 64-bit doubleword is located in memory for every major program or machine environment. This is shown in the following figure.

Sigma 9 Program Status Doubleword
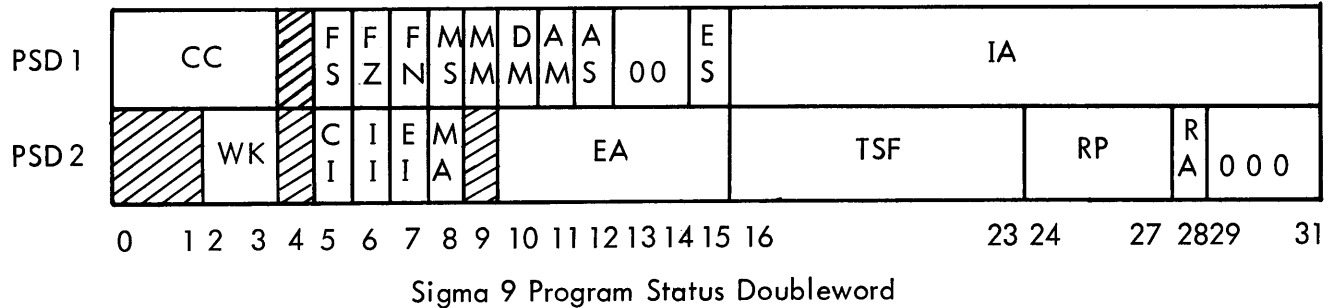
Where:

CC	=	Condition Code
FS	=	Floating Significance Control
FZ	=	Floating Zero Control
FN	=	Floating Normalize Control
MS	=	Master/Slave
DM	=	Decimal Mask
AM	=	Arithmetic Mask
WK	=	Write Key
CI	=	Counter Inhibit
II	=	I/O Inhibit
EI	=	External Inhibit
MM	=	Memory Map
ES	=	Extension Selector
RA	=	Register Altered
TSF	=	Trap Status Field
EA	=	Extension Address
AS	=	ASCII Mask
IA	=	Instruction Address
MA	=	Mode Altered
RP	=	Register Pointer

The environment of a program is completely described by:

.   The current value of the Instruction Address

.   The current value of the condition code

.   The three flags associated with the control of floating point arithmetic

.   The Master/Slave/Master Protected state

.   The map/unmapped control

.   The register altered flag

.   The decimal trap mask

.   The trap status field

.   The fixed point arithmetic trap mask

.   ASCII Mask

.   The write key

.   Extension selector

.   The counter, I/O, and external interrupt group inhibits

.   Mode altered flag

.   The register block selector

.   The extension address

Two instructions handle the loading and preservation of program status. Exchange Program Status Doubleword (XPSD) stores the existing status from active elements into core and loads another status into active elements, all in one instruction. Thus, if an XPSD is located at an interrupt or trap location, a program swap of control is brought about by the single instruction, guaranteeing fast and simple response. Alternately, if one program wishes to initiate another program without preserving the state of the initial program, the Load Program Status Doubleword (LPSD) may be used to give an even faster change of environment. LPSD may be used by an interrupt routine, for example, at the end of its operation, to restore control to the interrupted program. The initial PSD of the interrupt routine (which was loaded by an XPSD in the interrupt location) remains undisturbed and can cause the interrupt routine to be entered at its beginning when a subsequent interrupt is acknowledged. Sigma 9 Program Status Doublewords may be located anywhere in memory, thereby yielding an unprecedented degree of flexibility.

The following table offers a condensed view of PSD functions.

PROGRAM STATUS DOUBLEWORD FUNCTIONS

Designation	Bits	Name	Function
CC	4	Condition Code	Indicates the nature of the results of the previous instruction
FS	1	Floating Signifi-cance Control	Controls computer operation with respect to floating point significance checking
FZ	1	Floating Zero Control	Controls computer operation with respect to generation of floating point zero results
FN	1	Floating Normal-ize Control	Permits or inhibits normalization of the results of floating point additions or subtractions
MS	1		0 = Master 1 = Slave
MM			1 = No map 0 = Map
DM	1	Decimal Mask	0 = Decimal arithmetic trap not in effect 1 = Decimal arithmetic trap in effect
AM	1	Arithmetic Mask	0 = Fixed point arithmetic trap not in effect 1 = Fixed point arithmetic trap in effect
IA	17	Instruction Address	Stores the 17-bit program counter value
WK	2	Write Key	Used for memory protection
CI	1	Counter Inhibit	0 = All interrupts associated with internal clocks are permitted 1 = All interrupts associated with internal clocks are inhibited
II	1	I/O Inhibit	0 = All internal interrupts of the I/O group are permitted 1 = All internal interrupts of the I/O group are inhibited
EI	1	External Inhibit	0 = All external interrupts are permitted 1 = All external interrupts are inhibited

PROGRAM STATUS DOUBLEWORD FUNCTIONS

Designation	Bits	Name	Function
RP	4	Register Pointer	Stores the 4-bit register block pointer
AS	1	ASCII Mask	When = 0 EBCDIC preferred zone and sign codes are generated in decimal instruction execution When = 1 ASCII preferred zone and sign codes are generated in decimal instruction execution
ES	1	Extension Selector	When = 0 The Instruction address is a reference to a location in the first 64K of memory When = 1 The Instruction address is a displacement within the 64K region indicated by the Extension Address field of the Program Status Doubleword
MA	1	Mode Altered	When = 1 CPU is using Real Extended Addressing. Also used in conjunction with Memory Map bit. When both are 1, CPU is in Master Protected state
EA	6	Extension Address	Supplies additional high order bits necessary to form 22-bit instruction and reference addresses when CPU is using Real Extended Addressing
TSF	8	Trap Status Field	When a non-allowed operation trap occurs, the TSF will contain the virtual page address that caused the trap
RA	1	Register Altered Flag	Indicates whether or not any general registers or memory locations have been altered as the result of the execution or partial execution of an instruction that causes a trap

4.8.4   Traps

The time-sensitive demands of real time processing and the need to increase effectiveness in running and debugging conventional batch programs both require automatic detection of and recovery from errors. In the past, even very large computer systems have lacked hardware to

detect or react to error conditions. Operator intervention, undesirable even in batch applications, cannot be permitted in real time environments.

Automatic, non-stop operation without intervention requires two basic hardware functions: the first is the immediate detection of a program failure; the second is a mechanism for taking appropriate remedial action or aborting the offending program, as appropriate. The basis of Sigma 9's comprehensive checking system is the ability to make a rapid response and recovery from an error situation.

A trap is the immediate suspension of a programmed instruction, for specified and predictable causes, coupled with automatic diversion to a program that can deal with the error or faulty program. Whenever an error is detected, the state of the machine is preserved and control is automatically forced to a specified location for its next instruction. The instruction in the "trap" location usually initiates a special subroutine designed to cope with the particular error condition that was encountered. A trap permits, but does not force, the Master state to be entered by the associated trap-processing routine. Several distinct types of trapping are recognized by Sigma 9.

Some of the trapping conditions may be selectively ignored (or "masked") to increase program efficiency in cases where error conditions are checked by the program. To preserve the real time integrity of the system, the trap for any conditions that would result in uncontrolled or unpredictable processor action cannot be inhibited or "masked".

An attempt to perform any of the following operations always produces a trap:

- Using an operation code that has no definition
- Accessing a memory location that is non-existent (e.g., location 131,094 on a machine equipped with only 131,072 words of memory)
- Not completing an instruction within the maximum time permitted
- Executing a privileged instruction in the Slave state
- Violating a protected memory location
- Accessing a non-existent register block

. Executing an instruction in a trap or interrupt location that is not a legal trap or interrupt instruction

. Specifying an odd register in doubleword addition and subtraction, floating point long operations, translate and test byte string operations

. Arriving at a floating point characteristic overflow

. Attempting a floating point division by zero

An attempt to perform the following operations may result in a trap if such a trap is selected. The following are "maskable" trapping conditions:

. Violating the limits assigned to "push" and "pull" stack areas

. Reaching a fixed point arithmetic overflow

. Attempting division by zero

. Arriving at a floating point characteristic underflow

. Arriving at a floating point probable loss of significance

. Reaching overflow on decimal add or subtract

. Attempting decimal division by zero

. Having a non-digit appearing in a decimal arithmetic field (this trap always occurs if an improper digit is encountered in the EDIT instruction)

. Reaching a decimal quotient that is too large

In addition to the above trap conditions, a trap will also occur when:

. A parity error is detected in the map, on a data bus, or in memory

. A trap condition is detected during a trap or interrupt sequence

Another group of traps that cannot be inhibited is associated with the Sigma CALL's. These traps permit the program to easily and rapidly specify certain functional requirements or requests for service to the monitor, control programs, or user-defined service routines.

The trap system on Sigma 9 has the ability to provide the user with significant information concerning the nature of a trap. For example, when a non-allowed operation trap occurs as the result of a map operation which violates the memory protection feature, the virtual page

address that caused the trap is stored in the program status doubleword pointed to by the XPSD instruction in the trap location. This allows the programmer to determine the reason for the access protection violation at a later time. In addition, the Sigma 9 has an indicator, called the Register Altered Indicator, which informs the user whether or not the contents of any general registers or memory locations have been altered as a result of the execution (or partial execution) of an instruction that causes a trap. This feature of the Sigma 9 allows the determination of whether or not the instruction which caused the trap can be repeated after the condition leading to the trap has been corrected.

All instructions that reference a series of locations in the course of their execution which might cross a page boundary will test for trap conditions before execution begins. If an access protect, write lock violation, or reference to non-existent memory should occur during the course of execution, the trap will occur before execution begins. This gives the user an opportunity to correct the erring operation and to continue processing.

4.8.5   Real Memory Protection

Real memory on Sigma 9 may be protected by the use of the locks and keys. A 2-bit write key is contained within the Program Status Doubleword for any operating program. This must match the corresponding lock for each page (512 words) of main memory before a write operation is allowed into that page; the exceptions being, of course, that code 00 in the lock indicates that the page is open for writing by any key and the key 00 indicates that it is a master key and may write into any page regardless of the lock. The write locks, when present, apply at all times to both master and slave programs. Memory protection applies to the first 128K of memory, memory above 128K is assumed to have a write lock of 00.

4.8.6   Memory Map

Efficient utilization and management of computer memory are vital for time-sharing and multi-programmed applications. Generally, programs are swapped between core and fast auxiliary (such as XDS rapid access disk files). The program swapping is triggered either by an external demand or by a programmed scheduling routine. Under these circumstances, memory management becomes

quite complex and main core memory tends to become fragmented. Software solutions, involving the use of index registers coupled with strict programming conventions may partially overcome these problems, but only at the expense of programming restrictions and valuable mainframe time and space.

The Sigma 9 memory map offers the user an automatic method of managing computer memory. The memory map was specifically designed to solve the dynamic relocation and memory fragmentation problems. In the mapping mode of operation, memory is logically divided in up to 256 pages of 512 words each. Similarly, any address generated by a program running in the mapped mode (i.e., the instruction address, the address of an operatnd – called a virtual address) may be broken into one of 512 word addresses in one of 256 storage pages. Associated with each page of virtual addresses is a 13-bit field which controls relocation.

This array of 256 13-bit fields is known as the Sigma 9 memory map. A virtual address is mapped in the following manner:

1. The upper eight bits of the virtual address are used to select one of the 256 mapping registers. The lower nine bits are used in step 3 below

2. The contents of the selected mapping register are accessed

3. The 13 bits from the mapping register are combined with the remaining nine bits from step 1 to provide the resultant 22-bit address. The 13 bits occupy the high order portion of the 22-bit address

Thus, a program may be located anywhere in core and relocated to any other page or group of pages in core merely by changing the map. A privileged instruction, "Move to Memory Control", is utilized by the Sigma 9 processor to facilitate the maintenance and updating of the map. Programs written with contiguous addresses are easily relocated in a fragmented memory. Although such a program may occupy portions of memory that are non-contiguous, the program always behaves as if it occupied fixed, contiguous locations. A program may operate in either the mapped or unmapped mode. The selection is controlled by XDS software or a user program operating in the Master or Master Protected mode.

4.8.7   Virtual Memory Protection

In a time-sharing environment, common computational functions often exist among various oper-
ation programs. For example, consider three programs called A, B, and C. Program A requires
a sine routine; program B requires a sine routine and a square-root routine; program C requires
an arcsine routine, and the arcsine routine requires a square-root routine. This is shown diagra-
matically below:
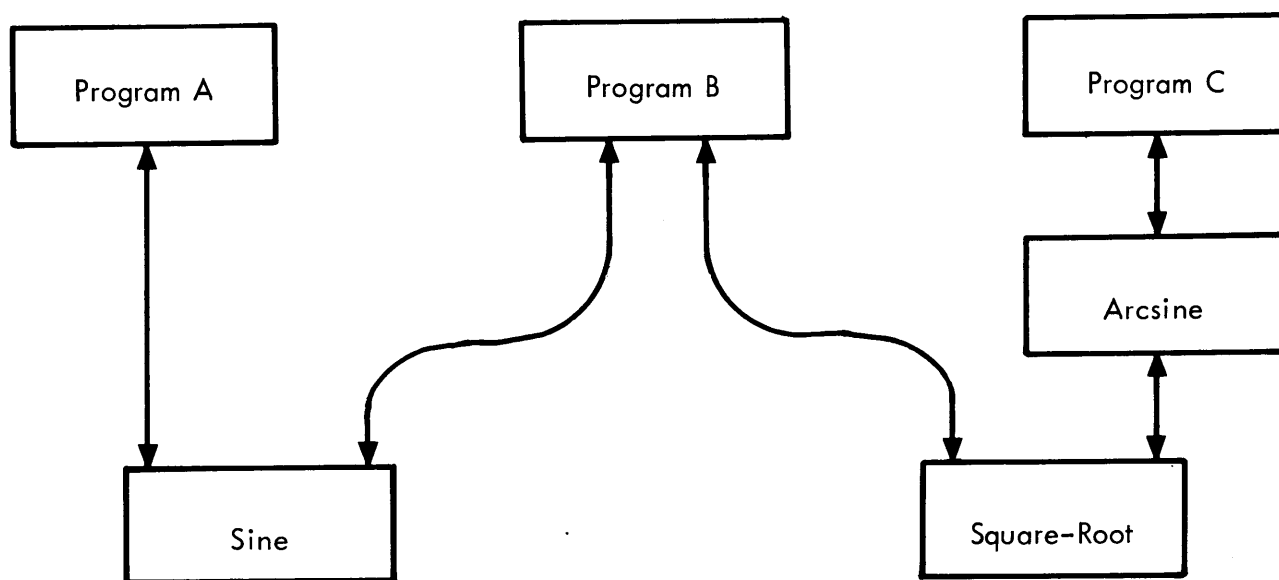
In most computer systems, the sine and square-root routines would appear twice, occupying
valuable core space. The Sigma 9 system requires only one copy of a common routine, with
the Memory Map providing supplementary memory protection so that, as control is switched
between A, B, and C, no subroutine can store results at an address that is not permitted as a
valid address for the program currently in control.

The supplementary protection offered by the Memory Map allows access to common subroutines
and shared data areas without constantly updating and changing the main memory locks or pro-
gram keys. Associated with each page of virtual memory is a 2-bit access protection code.
The access protection code indicates the allowed use or availability of the corresponding page
of virtual memory.

Access protection is in effect whenever the Memory Map is invoked (PSD 9=1) and the machine is operating in the Slave mode (PSD 8=1). Access protection is normally not in effect when the machine is operating in the Master mode. However, as an added feature to Sigma 9, the access protection may be made to apply to a Master Mode program through the use of the Mode Altered bit (PSD 40). When bit 40 and bit 9 are both "ON", the access protection will be effective in the Master as well as the Slave mode.

The four types of access protection are the same as on Sigma 7 and are as follows:

00   Open Access — This code indicates that the corresponding page may be written into, read from, or accessed for instructions

01   Data and Instruction Read — This code indicates that data and instructions may be read from the page but no information may be written into that page

10   Data Read — This code indicates that the page may be used to contain unalterable data only. No instructions, only data, may be read from the page and nothing may be written into it

11   Closed Access — This indicates that this page may not be referenced for any form of data read, instruction read, or write operation

4.8.8   Real Time Clocks

Real time applications require knowledge of the time of day as well as some means of measuring elapsed time intervals. In many real time applications, independent events are occurring that have their own timing sequences. Sigma 9 meets these varied needs with two standard real time clocks, a standard line-frequency timing source, and two optional additional clocks.

The first standard clock provides timing marks at the rate of 500 per second. The second standard clock and the optional clocks may be synchronized to the computer's power source (50 or 60 Hz) or can supply timing marks at the rate of 500 or 2000 per second. In addition, the user may supply timing marks from some external timing source. Any of the clocks may be used as an elapsed time counter or (by programming) may maintain time of day to any desired resolution.

Frequency sources are used to trigger a single instruction interrupt that causes a byte, halfword, or word to be modified in memory. When that information unit equals zero, a second interrupt is triggered.

The availability of more than one clock allows several independent programs to have separate time bases and relative priorities, eliminating the overhead of a central program that would otherwise have to derive several independent timing chains. XDS programming systems use the first standard clock to time the length of programs, permitting precise accounting information to be collected. They also permit programs to be terminated automatically when they have exceeded a specified maximum time, or initiate a program at a specified time of day.

4.8.9 Priority Interrupt System

Time-sharing and real time programs must all recognize and react to external stimuli. Most applications have an inherent hierarchy of importance. Some events are more important than others, and others require instantaneous servicing. The Sigma 9 priority interrupt system was designed with this hiearchical structure; all external stimuli are controlled and governed by this system.

Up to 224 levels of interrupt are available, organized into groups, and each group is divisible into a number of interrupt levels as shown in the following table. An external stimulus is associated with a particular interrupt level. Each level has a unique address, assigned in core, and a unique priority. Hardware automatically handles and identifies the priority and no special programming is required.

INTERRUPT GROUPS

Type	Group	Description
Internal	Override	The eight interrupt levels in this group always have the highest priority in a Sigma 9 system. The power-on and power-off interrupts for the power fail-safe feature have the absolute top priority. In addition to the system fault and memory fault interrupts, this group also contains count pulse interrupts that are triggered by pulses from internal or external clock sources.

INTERRUPT GROUPS (Continued)

Type	Group	Description
Internal (Cont.)	Counter	Each interrupt in the counter group (called a "count = zero" interrupt) is associated with a count pulse interrupt in the override group. When the execution of a Modify and Test instruction in the count pulse interrupt location causes a zero result in the effective byte, halfword, or word location, the corresponding count equal zero interrupt is triggered.
	Input/Output	This group includes the standard I/O interrupt and the control panel interrupt. The I/O interrupt location normally contains an XPSD instruction that transfers program control to a routine for servicing all I/O interrupts. The control panel interrupt permits the operator to initiate a specific routine. All interrupts in the I/O group may be inhibited by means of the internal inhibit feature.
External	2 15	A Sigma 9 system can contain up to 14 groups of interrupt levels, with up to 16 levels in each group. The groups may be connected in any selected priority sequence. All external interrupts may be inhibited by means of the external inhibit feature. Priority of each level within any group is fixed, with the first level having the highest priority and the last level the lowest.

For maximum system flexibility, each interrupt level may be individually disarmed (to discontinue response) and/or disabled (to defer response). This capability permits dynamic reassignment of priorities, even during actual execution of a real time program. Assignment of priorities among the interrupt groups is also flexible. In addition, the Sigma 9 interrupt system permits any interrupt to be triggered under program control. Thus, a hierarchy of interrupts may be nested for later processing, and special systems programs can be checked out before the special equipment has actually been attached to the computer. Instructions with extended execution times may be interrupted every few microseconds to guarantee fast system response to interrupt demands.

A Sigma 9 interrupt level is mechanized by means of three flip-flops. The first two are used to define any of four mutually exclusive conditions: disarmed, armed, waiting, and active. The third flip-flop is used as an enable. These conditions are described in detail in the following table.

INTERRUPT CONDITIONS SUMMARY

Condition	Flip-Flop Configuration			Description
	1	2	3	
Disarmed	0	0	-	When an interrupt level is in the disarmed condition, no signal to that interrupt level is admitted; that is, no memory is retained of the existence of a signal, nor is any program interrupt caused by it at any time.
Armed	0	1	-	When an interrupt level is in the armed condition, it is able to accept and remember an interrupt signal. The receipt of such a signal advances the interrupt level to the waiting condition.
Waiting	1	1	0	When an interrupt level in the armed condition receives an interrupt signal, it advances to the waiting condition and remains in the waiting condition until it is allowed to advance to the active condition. If the level-enable flip-flop is "OFF", the interrupt level can undergo all condition changes except that of moving from the waiting to the active condition. Furthermore, when this flip-flop is "OFF", the interrupt level is completely removed from the chain that determines the priority of access to the CPU. Thus, an interrupt level in the waiting condition with its level-enable "OFF" does not prevent an enabled waiting interrupt of lower priority from moving to the active state.
	1	1	1	When an interrupt level is in the waiting condition and its level-enable is "ON", it may advance to the active condition, provided the following restrictions are met: 1. The CPU is at an interruptible point in the execution of a program

INTERRUPT CONDITIONS SUMMARY (Continued)

Condition	Flip-Flop Configuration			Description
	1	2	3	
				2. The group inhibit (if applicable) is "OFF"
				3. No higher-priority interrupt level is in the active state or in the enabled waiting state with its inhibit "OFF"
				4. The CPU has acknowledged that all of the above restrictions have been met.
Active	1	0	1	When an interrupt has met all the requirements to move to the active condition, the computer acknowledges this and then stores the PSD at the location specified by the XPSD at the interrupt location.

Whenever an interrupt signal is recognized by a particular level, the interrupt system examines the priority of the triggered level (no central processor time or programming is required). The central processor constantly monitors the priority level output of the interrupt system. When it detects a waiting interrupt with higher priority than the current interrupt (if any), it executes its next instruction from a location assigned specifically to that interrupting level. The instruction in the interrupt location causes the existing state of the computer to be saved (if desired) and transfers control to a program that processes the interrupt appropriately.

The Sigma 9 interrupt system is unique in several respects:

. The true priority nature of the interrupt system permits external events to be acted upon in relation to their importance. Most other interrupt systems process interrupts in the chronological order in which they arrive

. Interrupt detection and priority establishment are independent of the computer and its program. All interrupts are examined in parallel and simultaneously. This means that an immediate response is always guaranteed since serial sequential scanning is not employed

. An interrupt can interrupt another interrupt (of lower priority) to provide immediate response when necessary. Other interrupt systems require the program and computer to process one interrupt fully before another can be accepted

. The current status of external interrupts may be examined and saved by Read Direct in-
structions. Once saved, the status may be re-established at a later time by Write Direct
instructions

. The flexible allocation of priority sequence among counter interrupts, I/O interrupts, and
external interrupts permits a user to designate his own priorities to meet his system re-
quirements

. A more flexible form of arming and enabling is incorporated in Sigma 9. Individual levels
may be enabled/disabled or armed/disarmed, for maximum flexibility in control of the
interrupt system and dynamic reassignment of priorities. Enabling and disabling may also
be done on a group basis

. Program-controlled triggering of interrupts permits a program to establish a lower priority
interrupt so that the lower priority segment of a high priority response may be deferred
to a later time. Program controlled triggering also permits programs to be checked before
external systems hardware is ready

. The I/O interrupt system is modular and automatically expands with the addition of each
new I/O device, with no change required in software

. Sigma 9's interrupt system not only has a minimal uninterruptible period but also minimizes
overhead required to preserve the old environment, establish a new environment, and ini-
tiate a useful response

4.8.10   Watchdog Timer

Real time environments require the constant availability of the computer for processing activi-
ties. Clearly, such computers must be protected from "hanging up" or stalling due to hardware
or program faults. The essence of a "hang-up" is that the computer begins an instruction and,
because of some failure, fails to complete that instruction, waiting for an "end" that never ar-
rives.

The Sigma 9 system includes a watchdog timer which constantly monitors the length of time be-
tween interruptible points in the CPU. Should an instruction take more than the maximum time
allotted for its completion, it is automatically cancelled and the processor immediately enters

a routine that determines the cause of the failure. Thus, real time sensitivity is maintained and the overall integrity of the processor is guaranteed.

Sigma 9 can never, under any circumstances, hang up in a condition where it cannot recognize an interrupt. Since it was designed to operate in a real time environment, Sigma 9 must, and is, always responsive to interrupts, except in those cases where the program has decided to disarm individual interrupts.

4.8.11   Abort/Restart and Abort/Continue

The ability to respond to external stimuli is primary for modern computing systems. Sigma 9 has many features to insure recognition of these stimuli in the order of their importance, and other features which insure a rapid response.

Some instructions require multiple machine cycles for execution. Since (in conventional computers) an interrupt is generally recognized only at the completion of an instruction, "long" instructions disallow fast response to external stimuli. Sigma 9 design includes the ability to abort long instructions during the course of their execution to facilitate rapid interrupt recognition. In general, the "long" Sigma instructions may be interrupted every few microseconds so they appear to be normal length instructions repeated many times. If an instruction is aborted, the machine state is preserved so that no ambiguity arises when the instruction is re-started. Thirty microseconds is the maximum that an interrupt must wait to be serviced.

Usually, after the interrupt response program is complete, control returns to the aborted instruction at the point of previous calculation. No extra time is required to process the aborted instruction. This "abort/continue" feature for long Sigma instructions permits fast response to interrupts without lengthening the total computation time. In a few special cases the aborted instruction is re-initialized and re-executed from its beginning. Since they are so few in number and infrequently executed, there is only an extremely small probability of increased total problem execution time.

4.8.12   Power Fail-Safe

The power fail-safe feature, pioneered by XDS for its 9-Series computers in 1962, is a Sigma feature which permits the system to recognize the loss of primary power and come to an orderly halt, preserving active registers in non-volatile core storage while sufficient power is available to perform these few operations.

The two highest priority interrupts are assigned to the Power Fail-Safe feature. Whenever a potential primary power failure is detected, a standard XDS program is activated. This program stores all status information and brings the computer and I/O system to an organized halt. Backup equipment may then be activated by this program if available. When primary power is restored, the program restores the computer to its previous state, "cleans up" the I/O and peripheral systems, and returns control to the program at the point of interrupt, all without operator intervention.

4.9   Reliability and Maintainability Features

The Sigma 9 system has many advanced design characteristics and features which provide for reliable operation and efficient maintenance, incorporating all of the reliability and maintainability design features of the Sigma 7 system. A Sigma 9 system is thus able to perform at a high level of operational availaaility, which means more continuous service to the user.

4.9.1   System Features

The Sigma 9 complement of <u>diagnostic programs</u> utilizes the maintainability features of each of the Sigma 9 units. Interface with maintenance personnel is simplified and activated through a local keyboard printer or via telephone lines to a diagnostic laboratory for rapid analysis.

Diagnostic programs are designed around a multi-level structure consisting of the following capabilities:

.   System verification and testing to determine which unit is faulty

. Unit functional testing to determine the specific function of a unit that is at fault

. Fault location diagnosing to analyze what physical component is malfunctioning. Fault location makes use of the snapshot features

"Snapshot" features provided in Sigma 9 units enable diagnostic programs to retrieve internal registers and control flip-flops not otherwise "visible" to a program. This facility is useful in determining the system status at the time a fault occurs as well as in locating the source of the fault condition down to the smallest set replaceable elements.

Extensive error logging is provided in Sigma 9 systems. When a fault is detected, system status and fault information are available for program retrieval and logging for subsequent error analysis.

A Sigma 9 system is reconfigured through the use of partitioning controls. Each Sigma 9 unit may be partitioned out of the system by selectively disabling it from the system busses. Faulty units are thus isolated from the operational system. Re-enabling the connection allows repaired units to be returned to service.

RIO (Reset I/O), an optional form of the HIO instruction (Halt I/O), provides for programmed I/O reset which operates exactly as though the I/O reset had been initiated with the switch on the Sigma 9 PCP. The addressed Sigma 9 IOP and all peripheral devices connected to it are initialized. This feature is especially useful for diagnostic program effectiveness.

Parity is checked on all data and addresses communicated in either direction on busses between memory units and processors (CPU's, MIOP's, and HSRIOP's). This feature provides fault detection and fault location capabilities which enhance the ability of an operating system or diagnostic program to quickly determine which unit is faulty.

Sigma 9 system units are provided with clock and voltage margining capability which assist the customer engineer or diagnostic program to quickly locate the source of an intermittent fault. Programmable clock margin control is provided. Both clock and voltage margin status are available for program retrieval. NOT NORMAL conditions are indicated on the PCP (Processor Control Panel).

The Alternate Processor (optional) bus feature provides a redundant connection of the IOP's to the CPU's in the Sigma 9 system. This feature may be used to partition IOP's for diagnostic or reconfiguration purposes.

4.9.2   Core Memory Features

The memory system is composed of memory banks each containing snapshot logic which is automatically activated when a memory fault occurs to record the nature and environment of the fault. The contents of the memory snap registers (each 32 bits in size) can be programmatically retrieved by the use of a new instruction, LOAD MEMORY STATUS. This feature may be used by the operating system for error logging or by a diagnostic program to assist in fault locating. Notification of a fault occurrence is via the Memory Fault Interrupt.

Memory Fault Detection covers the following types of fault occurrences:

- Parity errors as information is read out of core, detected in memory bank
- Parity errors detected on addresses received from processors
- Parity errors detected on data received from processors
- Port selection errors (PSE) detected if more than one port is simultaneously selected for one bank. PSE aborts the requested operation without modifying the contents of any memory cell
- Memory bank operational status; (e.g., over-temperature, d.c. voltages out of tolerance, etc.)
- Data Loop Check Fault. Data loop checking is a feature which provides additional fault detection on read operations. As data is gated onto the memory bus for transmission to a processor, it is also gated from the bus back through the input path, clocked into a register and checked for parity. Thus, the integrity of the line driver/line receivers at the memory are tested on every "read" cycle.

The interleaved mode of core memory operation may be disabled for certain diagnostic purposes with a switch located on the PCP.

Clock margins are controlled manually by means of switches or programmatically by special coding of the LOAD MEMORY STATUS instruction.

4.9.3   MIOP Features

The Maintenance Interface Bus (a special mode of the direct I/O bus) is brought to each MIOP from the CPU for maintenance purposes. The MIOP responds in the following way to special WRITE DIRECT and READ DIRECT instructions executed by the CPU:

  . Under RD control, monitors one of 32 selectable groups of MIOP logic
  . Under WD control, provides single phase clock control of the MIOP
  . Under WD control, provides a snapshot of operation whereby a selected display group is stored in a "snap" register at the end of a preset countdown for later monitoring by an RD instruction
  . Under WD control, writes directly into an addressed subchannel
  . Under RD control, reads directly from an addressed subchannel
  . Under WD control, sets the clock margins to Fast, Normal, or Slow rates

Parity is checked on information brought out of each subchannel fast access memory (FAM). A fault is reported to the system via the Processor Fault Interrupt.

A Maintenance Subcontroller (MS) feature on each I/O channel is provided to assist in diagnosing the I/O system. A diagnostic program controls and monitors the MS via the Maintenance Interface and the I/O bus. The following functions may be accomplished:

  . Simulation of a device controller responding to commands sent to it by the MIOP, receiving and sending strings of data bytes
  . Monitoring of IOP bus during diagnostic operations
  . Exercising the IOP at variable rates up to and including its maximum rate
  . Self-testing of the Maintenance Subcontroller logic

4.9.4   HIGH SPEED RAD I/O PROCESSOR (HSRIOP) Features

The Maintenance Interface Bus (a special mode of the direct I/O bus) is brought to the HSRIOP from the CPU for maintenance purposes. The HSRIOP responds in the following way to special WRITE DIRECT and READ DIRECT instructions executed by the CPU:

. Under WD control, selects a phase which, when entered during execution of any HSRIOP operation, will cause the HSRIOP to halt. At this time, the HSRIOP may be "snapped" for diagnostic purposes, via RD control

. Under RD control, "snaps" one of seven selectable groups of internal HSRIOP logic

. Under WD control, provides single phase clock control of the HSRIOP

. Under WD control, selectively sets various fault indicators (e.g., device and memory faults) to simulate actual fault occurrence. This feature allows the diagnostic to test for correct HSRIOP response under these fault conditions

. Under WD control, selectively initiates one of two test modes of the HSRIOP in which the HSRIOP responds to normal I/O instructions while simulating action of the storage units. In this way, major portions of the HSRIOP logic can be diagnosed separately from the Storage Units

Test Mode 1 — In this test mode, the HSRIOP responds to WRITE/READ commands. Data is transferred from memory into the data buffer and sent back to memory. Called the "short loop" test, the memory interface, data buffer, and control logic are all checked in Test Mode 1. Test Mode 1 is initiated via Maintenance Interface WD action.

Test Mode 2 — In this test mode, the HSRIOP responds to WRITE and READ commands. Data is transferred from memory through the data buffer, on through the de-skew logic and then back to memory via the data buffer again. This is called the "long loop" test. Assuming the "short loop" test was successful, the de-skew logic is specifically checked in Test Mode 2. Test Mode 2 is initiated via Maintenance Interface WD action.

4.9.5   CPU Features

The PCP (Processor Control Panel) is divided into two sections. The upper portion (MAINTE-
NANCE SECTION) contains controls and indicators used exclusively by maintenance personnel.
The lower portion of the PCP is used primarily by operating personnel to load, execute, and
troubleshoot programs. A CONTROL MODE switch is provided to disable certain of the mainte-
nance functions during normal operation.

Various phases, control flip-flops, and registers of the CPU and decimal unit can be displayed
on the PCP. A 16-position thumbwheel switch identifies and selects display information during
maintenance activities.

All of the CPU logic that can be displayed on the PCP can be monitored by a program with the
snapshot logic. At a pre-selected clock time of a given instruction, execution selected logic
is stored into a 32-bit "snap" register. The contents of the "snap" register are then retrieved by
a specially coded READ DIRECT instruction. By comparing the "snapped" information with known
correct information, the diagnostic program may accurately determine a specific fault. The fail-
ing component may then be deduced. Snapshot action can also be initiated at the PCP, and the
contents of the "snap" register displayed.

Clock margin control is accomplished manually from the PCP by using the CLOCK MARGIN
switch or under program control with a properly coded WRITE DIRECT instruction. Three clocks
rates are provided — Normal, Fast, and Slow. Manual memory CLEAR and SCAN capabilities
are provided which enable operators or maintenance personnel to rapidly clear, read, or store
selected data into any or all consecutive CPU core memory locations. During the read SCAN
operation, the CPU can be made to halt on a memory parity error, at which time the address
and data of the indicated memory location can be displayed.

The Address Stop feature allows the operator or maintenance personnel to:

.   STOP on any instruction whose virtual address equals the SELECT ADDRESS switch value.
    At the time of the halt, the instruction pointed to by the SELECT ADDRESS appears in the
    DISPLAY indicators

- STOP on any real memory reference indicated by the SELECT ADDRESS switch

- STOP when any word in a selected page is addressed.

Manual interrogation of the DIO bus (including the maintenance interface is provided with information displayed on the PCP. This feature is in addition to interrogation provided via the READ/WRITE DIRECT instruction. Thus, all devices connected to the DIO or MIO may be examined manually by maintenance personnel.

The CPU has a single clock mode of operation which enables maintenance personnel to execute an instruction one internal phase at a time from the PCP.

Capability is provided to selectively override the operation of the instruction W.D. TIMER (watchdog timer) and the operation of the DECIMAL unit to aid maintenance personnel in diagnosing related machine faults.

The registers containing the MAP are checked by parity.

5.0   SIGMA 9 INPUT/OUTPUT

Many basic computing functions which involve reading or writing data, such as sorting, file processing, program assembly, and compilation, are input/output limited. That is, the time required to input and output the data that is processed is greater than the time required to process the data itself. Normally, in these cases, the computer must wait or "idle" while successive reading or writing operations are completed. There are two general approaches to eliminating this idle time.

The first approach involves the use of one or more auxiliary computers to handle input and output operations. Data is passed between these auxiliary processors and the main computer by magnetic tape, a shared disk file, or through a common (usually large and slow) core memory. This approach suffers from high cost and the need for different hardware.

The second approach involves adding input/output channels to the main computer, buffering data onto a high speed drum or disk, and using a small part of the main computer's memory and time to control a number of peripheral devices. For example, a computer of the Sigma 9 class, used in a closed-shop, batched FORTRAN environment, would typically keep two 1500-card-per-minute readers, a 300-card-per-minute punch, and a 1000-line-per-minute printer running at full speed most of the time, while doing computation at maximum speed. This type of multiple I/O operation (known as symbiont processing), to be economically feasible, requires that the incremental cost for each input/output channel be as low as possible. Such simultaneous operation requires that each input/output unit have á separate I/O control and transmission path.

Special emphasis in the Sigma design was given to reducing to an absolute minimum the cost per separate I/O control-and-transmission path. This not only makes on-line, symbiotic processing reasonable, but also enhances multi-terminal communication processing, by enabling the low-cost addition of fully buffered communication devices. Further, Sigma design requires a minimum of programming to handle this type of operation.

The mere provision for such low-cost, multiple input/output channels in a computer is no guarantee that even this cost of incremental hardware for simultaneous I/O is at all justified, much

less, minimized. Traditional approaches to computer I/O design have usually required a large amount of programming to control the transmission of data or to monitor I/O channel activity. Even with a program interruption capability, programs have usually had to be concerned with many details that tended to increase overhead, generally in direct proportion to the number of on-going I/O operations.

The problem then is to reduce this program overhead and free the central processor from routine details, enabling it to perform work of a more productive nature. Central processors ought to be computing elements, rather than input/output controllers.

In the Sigma 9 system, input/output operations are primarily under the control of one or more input/output processors (IOP's). This allows the CPU to concentrate on program execution, free from the time-consuming details of I/O operations. Sigma 9 IOP's require only an initializing sequence from the central processor. Once initiated, each IOP operates independently of the CPU and without the need for further intervention, performing one or several separate functions as required. Any I/O event that requires CPU intervention is brought to its attention by means of the interrupt system. All normal control and sequencing of the I/O system is handled by the IOP.

In the following discussion, certain terminology conventions are used: the CPU executes instructions, the IOP executes commands, and the device controllers or devices execute orders. To illustrate this difference, assume that the CPU executes a Start Input/Output (SIO) instruction to initiate an I/O operation. During the course of the operation, the IOP may issue a command (called Control) to transmit a character to a device controller or peripheral device which, in turn, interprets the character as an order, such as Rewind.

Another distinction that should be clearly understood is the difference between data chaining and command chaining. Data chaining is used to perform scattering or gathering of information sorted within a single logical record. In such an operation, the system gathers (or scatters) data within one record from (or to) more than one region of memory. Command chaining, on the other hand, refers to execution of a sequence of I/O commands, under control of an IOP, on more than one logical record within a physical record. Thus, data chaining operates with respect

to data in a single record, never on data in separate records. A new command must be issued
for each record, even if the operation to be performed is identical to the preceding operation
on another record. Hence, command chaining is required whenever multiple records are in-
volved.

A Sigma 9 IOP is quite similar to a conventional central processor in many ways. An IOP, once
initiated, accesses memory first for information regarding the operation to be performed (com-
mands), and then accesses it (as many times as necessary) for input or output transmissions (data).
Typically, a number of different types of I/O units are required for a Sigma 9 system. The
interfaces between the CPU, core memories, and I/O units are sufficiently generalized that no
redesign of existing interfaces is necessary for new types of I/O capability. The two basic types
of input/output processors are the multiplexor IOP and the HIGH SPEED RAD IOP. Other types of
IOP's may easily be added in the future if the need arises. The Sigma 9 system can incorporate
up to eleven IOP's of varying types.

5.1   Multiplexor IOP (MIOP)

The multiplexor IOP permits slow to moderately high speed data rates to be handled on a time-
multiplexed basis on two channels, Channel A and Channel B. It possesses its own private fast
memory, and handles up to 24 subchannels on Channel A and up to eight subchannels on Chan-
nel B. The multiplexor IOP is characterized by the fact that a central set of registers and
arithmetic elements service a large number of individual peripheral devices. Its private memory,
accessible only to the IOP, takes advantage of the inherent speed of fast memory units. When a
chaining operation occurs, the IOP accesses two 32-bit I/O command words from core memory
and stores this information in its private memory.

A device controller requests service from the IOP by presenting its identification to the IOP.
The multiplexor IOP uses this information to generate an address that accesses the private mem-
ory command word for that device controller. Thus, the IOP acquires the control information
necessary for the data exchange.

Although the multiplexor IOP is the central and most complex element of an I/O system and is shared by all the device controllers, it is essentially a passive unit in the data transmission process, responding when called upon. The device controller is the active element in the system, initiating almost all action during the execution of an input or output operation.

Associated with each device controller (or device) is a logical channel. Each channel provides storage within the IOP for all data required to control transmissions for that device (word count, address, etc.), as well as to sequence the required operations. The design of the Sigma 9 I/O system permits a magnetic tape controller with eight tape transports to use only one I/O channel rather than eight.

Each channel, A or B, of the multiplexor IOP can handle devices with combined transfer rates up to 470,000 eight-bit bytes per second for byte-oriented devices. Devices that can take advantage of the optional 4-byte interface feature can transfer data up to 940,000 bytes per second. Thus, a number of standard speed peripheral devices may all be operating simultaneously on a multiplexor IOP. This approach minimizes the cost of adding additional I/O channels. Greater capability, from 8 to 24 channels, is achieved simply by adding fast memory elements, rather than conventional registers, to the IOP.

The Sigma 9 MIOP offers a Memory-to-Memory Move option that provides the ability to move data from one area of memory to another. This feature allows the MIOP to transfer data within memory without the aid of the CPU once the transfer is initiated. To use this feature, the programmer simply gives the MIOP the starting address of the data to be moved, the destination address in memory of where the data is to be moved, and a count of how much data is to be moved. The MIOP then moves the data unaided by the CPU. The programmer may or may not request the MIOP to give an I/O interrupt when the operation is completed.

5.2   HIGH SPEED RAD IOP (HSRIOP)

The second type of input/output processor is called HIGH SPEED RAD IOP. In contrast to the multiplexor IOP, the HIGH SPEED RAD IOP is designed to accommodate very high speed devices, one at a time. Rather than simultaneity, the HSRIOP offers accommodation for devices which have very

high transfer rates. This is true even when data chaining is required, since the HSRIOP is capable of command look-ahead. Because the HSRIOP can transmit information at the rate of the main memory (32 million bits per second), it permits Sigma 9 to handle such devices as high speed rapid access disk units and other high speed devices. Input/output programs (the commands that are executed by IOP's) are fully generalized, and do not differ for multiplexor or HSRIOP. This simplifies both hardware design and programming and insures the ability to interchange devices among IOP's without changing the I/O command list.

5.3   Direct Input/Output (DIO)

The direct input/output interface (which utilizes the Read Direct and Write Direct instructions) may be used both for internal control functions and for direct communication with other elements of the Sigma 9 system.

This interface allows the transfer of a 32-bit data word between an affected register and an external device. In addition to the data word, a 16-bit address is transferred for control and selection purposes. Each transfer is under direct program control. The same instruction which transfers data and control also accepts status information from the external device, to permit rapid sensing of external conditions. Since this system operates at a maximum data transfer rate, it is generally used for data transfers of short bursts or asynchronous timing to avoid tying up an automatic channel. Direct I/O is also useful when data is to be accepted at medium to high speed, or when each input must be examined immediately as it is received.

5.4   System Interface Units (SIU's)

A successful real time system requires more than just a highly efficient systems-oriented processor; it also requires special hardware and software for dealing with the wide variety of information forms usually found in such applications. XDS offers an extensive array of such special systems, referred to as system interface units. These standard, off-the-shelf units connect analog and digital devices to the computer while fully exploiting the advanced input/output features of Sigma 9. Because the units are modular in design, the user selects only those he requires, reducing not only his initial costs, but also the expense of future expansion.

Because system interface units are standard equipment, they satisfy individual system requirements without the need for special engineering. The resulting benefits to the user include:

- Lower Cost — User costs to perform each function are lower because the systems are standard units in production

- Ease of Design — Because the SIU's offer flexible performance in a variety of standard interfaces closely related to his end tasks, the user can design his own system more simply, rapidly, and efficiently. System implementation time and cost are sharply reduced and productive results achieved much sooner

- Compatible Software — Because the system interface units are standard and use standard software, they are fully compatible with XDS programming systems. Standard I/O handler routines control these units in the same manner as standard peripheral equipment. System interface units are easy to use in real time applications, even where concurrent foreground/background processing is required, because the SIU programs can operate under control of standard XDS monitors. The time and cost involved in the actual use of equipment are minimized, as well as are programming training requirements

- Ease of Maintenance — A full range of standard diagnostic and checkout programs facilitates maintenance. These permit a high level of off-line checkout and maintenance without hampering real time system operation

- Extensive Documentation — Thorough documentation is available to the user even before hardware is installed. The documentation provided with each SIU includes assembly drawings, module location charts, logic diagrams, interface manuals, and theory of operation manuals which contain connection locations, pin numbers, signal definitions, unit loads, detailed timing diagrams, and logic equations

- Greater Flexibility — XDS engineers researched the design of these units with emphasis on their interaction with other system elements. The resulting flexibility of use assures that current system needs are easily met and future expansion and change can be made at minimum cost to the user and minimum disturbance to the operating system

- Concurrent Foreground/Background Processing — XDS system interface units are designed to release the computer from the many routine details involved in real time I/O coupling through direct input/output. Thus, almost all of the system's capacity is available for

useful processing. Concurrent processing of a real time task in the foreground and a general purpose task in the background is economically sound

The available standard XDS system interface units are described briefly in the following table. XDS brochure number 64-28-01, "XDS Sigma System Interface Units", contains detailed descriptions of each unit, as well as installation data and typical systems that can be constructed through use of these standard units.

XDS provides two levels of software support for system interface units: operating system software and maintenance software (including analog calibration and checkout programs and digital I/O programs). These are compatible with software support on all standard XDS products. Because the XDS system interface units are standard, XDS programming systems permit the user to deal with them simply. Even when a real time program is written in a language such as FORTRAN IV, the user may control these units in a variety of convenient ways, including the use of subroutines called by the compiler.

XDS SYSTEM INTERFACE UNIT SUMMARY

Name	Number per System	Function
Analog Input Controller	Unlimited	Provides the interface and control necessary to operate an analog-to-digital converter and a high speed analog multiplexor through a Sigma 8-bit I/O channel. Permits random or sequential sampling of analog inputs at pre-specified intervals.
Analog Output Controller	Unlimited	Provides the interface and control necessary to operate from 1 to 16 digital-to-analog channel controllers through a Sigma 8-bit I/O channel. Each digital-to-analog channel controller can operate 16 digital-to-analog converters. This unit allows analog input to be varied randomly, sequentially, or simultaneously, all at pre-specified rates.
IOP to DIO Adapter	Unlimited	Transforms any Sigma 8-bit I/O channel into an interface identical to a Sigma 32-bit DIO interface. Enables users to perform a program-specified number of 32-bit direct input or output operations, in any combination, via the 8-bit I/O channel.
Digital I/O Subsystem	Up to 8	Generates pulsed digital outputs, transfers data in memory to output registers, and stores the states of input signals in memory. Each fully expanded unit accommodates 960 digital inputs, 480 pulsed outputs, or 480 stored outputs, in various mutually exclusive combinations.
Analog and Digital Adapter	1 via DIO	Provides the interface and control necessary to operate one analog-to-digital converter, a 256-channel analog multiplexor, 256 channels of digital-to-analog conversion, input 48 digital signals, and 32 stored (or pulsed) digital output signals.
Frequency Control Subsystem	Up to 2	Provides for frequency control of XDS analog input, analog output, and digital transfer control units, thus enabling external devices to perform operations at regular, pre-specified intervals. Each fully expanded unit furnishes four independent frequency sources. The frequency of each source may be specified either manually or by the program.

XDS
Xerox Data Systems

5.5   Peripheral Devices

The extensive range of XDS peripheral units makes possible an unusual degree of system productivity and flexibility. Each peripheral device operates in conjunction with a device controller which establishes logical interface lines between the device and the input/output processor. The device controller also sets up timing functions which control the peripheral device's operation, clocking, and data transfer.

XDS also provides standard software that effectively complements Sigma peripheral hardware capabilities. Input/output handlers, as well as diagnostic programs, are provided for each peripheral device. The input/output handlers may operate either with a stand-alone system or be interfaced to a monitor system.

To facilitate maintenance of Sigma peripheral equipment, XDS provides an optional peripheral equipment tester. A plug-in connection on each device controller accommodates this tester. Peripheral devices may thus be exercised and tested off-line, minimizing costly CPU down-time for equipment maintenance.

To achieve maximum reliability standards, XDS designs and manufactures all peripheral electronics. Mechanical mechanisms of most units are of XDS design and manufacture, or were chosen from the most reliable, field-proven equipment available. The chart which follows summarizes the XDS line of peripherals.

XDS PERIPHERAL EQUIPMENT SUMMARY

Equipment	Characteristics
Rapid Access Data Files	Capacities to 6.2 million bytes per unit; transfer rates to 3 million bytes per second; average access times as low as 17 milliseconds. Fixed read/write head per track eliminates time delays associated with moveable-head units.
Removable Disk Storage	Storage capacity of 49.0 million bytes/disk driver; one or two drives per unit; four disk units per controller; transfer rate of 312,000 bytes per second with an average access time of 87.5 milliseconds. Multiple seek operations may be initiated to overlap with a subsequent single read or write operation. Capacity for each fully expanded system is 196.6 million bytes.
Magnetic Tape Units	Seven- and nine-track systems; IBM-compatible; speeds to 150 inches per second; transfer rates to 120,000 bytes per second.
Graphic Display	A comprehensive display system which includes character and vector generators; nominal display area is 10 by 10 inches, with three character sizes of 0.18 and 0.36 inches high. There are two vector generator rates which are automatically invoked depending upon the maximum coordinate increment of the vector. The unit includes a light pen, two keyboards, and action switches for operator control.
Card Equipment	Reading speeds to 1500 cards per minute; punching speeds to 300 cards per minute. Intermixed binary and EBCDIC card codes. Simplified punch programming does not require "corner-turning" logic.
Line Printers	Fully buffered, speeds to 1500 lines per minute; 132 print positions with up to 96 different characters.
Keyboard/Printers	Ten characters per second. Also available with integral paper tape reader (20 characters per second) and punch (10 characters per second).
Paper Tape Equipment	Readers with speeds to 300 characters per second. Punches with speeds to 120 characters per second.

XDS PERIPHERAL EQUIPMENT SUMMARY (Continued)

Equipment	Characteristics
Data Communications Equipment	A complete line of character-oriented and message-oriented equipment to connect remote and local user terminals to common carrier lines.
Remote Batch Terminal	System connects to Sigma directly or via a voice-grade phone line. It includes a 250 line per minute printer, 200 card per minute card reader, 75 card per minute card punch, and an operator's console. The unit can answer incoming calls automatically, provide off-line listing, and operate in a full duplex mode.
Peripheral Equipment Switch	Peripheral devices may be shared by two or more Sigma I/O processing units via the peripheral equipment switch. Switching of a device can be accomplished by either manual or program control.
Channel Interface Unit	Half duplex communications between two Sigma IOP's under program control is provided by the channel interface unit. Transmission rates will automatically fluctuate to absorb the bandwidth available to the IOP's.
System Interface Units	SIU's consist of a line of products that solve the interfacing problems of real time and process control applications. Among other functions, they provide a means of performing analog and digital I/O to devices not normally considered to be computer peripherals.

6.0   SIGMA 9 SOFTWARE

The comprehensive software services available for use with the XDS Sigma 9 computer system enables the user to derive all of the benefits of the advanced architecture of the Sigma 9 hardware. XDS offers a complete range of software elements (language processors, applications systems, and utilities) integrated into a system that fully exploits the Sigma 9 hardware and, yet, is user-oriented.

Being able to apply the computer's vast powers to the problems of modern business is a concern shared by many. Access by local and remote users must be efficient and user-oriented. XDS has taken a giant step toward a solution to this problem by developing the Universal Time-sharing System for use with the Sigma 9. This section discusses UTS as well as the various subsystems available for the XDS Sigma 9.

6.1   Universal Time-Sharing System

The Universal Time-Sharing System (UTS) is an operating system that permits on-line conversational time-sharing, batch processing, and real time processing to operate concurrently on an XDS Sigma 9 computer. UTS features include:

- Conversational time-sharing, local and remote batch processing, and real time processing
- High system efficiency due to memory relocation map, shared reentrant processors, and multiple CPU independent I/O processors
- For on-line users: extensive conversational software, complete file saving, and restoration features, and access to all batch processors (COBOL, MANAGE, FORTRAN, SORT/MERGE, etc.)
- For batch users: access to full system resources through local, on-line, or high speed remote job entry
- For installation managers: a thorough system monitoring and reporting scheme including the ability to control and tune system resources for maximum efficiency. Extensive error checking and recovery features plus integrated hardware/software partitioning of system components for fail-safe operation.

. <u>For all users:</u> comprehensive accounting system, maximum system reliability, extensive file security features, and a complete set of powerful processors

6.1.1   Philosophy of Operation

UTS allows on-line time-sharing users to concurrently create, debug, and execute programs using a variety of powerful and comprehensive language processors and facilities. These include:

. LOG ON/LOG OFF
. TEL (Terminal Executive Language)
. EDIT
. BASIC
. PCL (Peripheral Conversion Language)
. TOM (Terminal Oriented MANAGE)
. XDS FORTRAN IV
. META-SYMBOL
. LINK
. FDP (FORTRAN Debug Package)
. DELTA (Assembly Language Debugger)
. SYMCON (Symbol Control)
. SUPER (Authorizes users and their resources)
. CONTROL (System performance monitor)
. BATCH (Terminal job entry in the batch job stream)

UTS users with batch processing requirements may choose from local batch entry (at the computer site), terminal batch entry (batch jobs initiated from an on-line time-sharing terminal), or remote batch entry (batch jobs using a remote high speed batch terminal). Jobs are processed through symbionts in order to minimize I/O overhead. In addition to the on-line processors and facilities, batch users have these additional processing capabilities:

- XDS FORTRAN
- XDS COBOL
- SORT/MERGE
- MANAGE (File retrieval, update, and report generator)
- DMS (Data Management System)
- 1400 Series Simulator
- FMPS (Linear Programming System)
- SL-1 (Simulation Language for Continuous Systems)
- GPDS (General Purpose Discrete Simulator)

The UTS system is completely integrated in that all processors generate the same format for binary object code. This common format means that no special headings are required in order to have processor compatibility. Any combination of routines generated by an XDS processor can be linked and loaded by the system loader. This further allows multiple linkages at load time providing the efficiency that could be realized by assembling all of the loaded subroutines as one large routine.

Extensive management control facilities are provided by UTS to allow installation management to effectively control system resources and dynamically balance the system in order to maximize performance. These facilities include extensive accounting statistics, automatic periodic system performance monitoring, and dynamic control of system parameters.

Because the UTS monitor is flexible, the term "user" in UTS refers to any user of the system, whether he be a time-sharing, batch, or remote batch user. The monitor rarely makes the distinction between user classes and selective handling is left up to UTS processors which differentiate, according to preset criteria, and act accordingly. This lack of monitor distinction between classes of users allows for the expansion of UTS to its full potential in a multiprogramming environment.

6.1.1.1   UTS Scheduling

The UTS scheduler performs two major system functions: the selection and organization of users to be swapped in and out of core memory, and the selection of users for execution. Each user has one entry in a set of state queues. (A state queue is a chain of users in the same state; e.g., waiting on terminal output). Twenty-eight such queues are maintained by the scheduler to select users for: (1) swapping in, (2) execution, and (3) swapping out. Results of this algorithm allow a flexible means of establishing user priorities and an intelligent selection of users to swap in and out. The queues are also used for entry and use of processors and I/O activities awaiting "wake-up" at a preset time.

The scheduler is driven by events reported throughout the system. An event usually results in a state change for a particular user. Before returning control to the routine which reported the event, the scheduler will check to see if this event has made a swap necessary and, if so, will schedule and initiate a swap before returning control. Given a reasonable ratio of available core to user size (four to one), the scheduler can keep swaps and computing virtually 100% overlapped. Swaps are organized for maximum efficiency which, in the case of a large swap, will reduce latency time well below the average of 17 milliseconds.

6.1.2   UTS Subsystems

UTS provides a wide range of system services and facilities to aid both users and installation management in utilizing the system most effectively. These include: (1) facilities for logging on and off the system and for summarizing system utilization at log-off time; (2) flexible and convenient file management facilities; (3) provision for extensive reporting and accounting of system use by separate account; and (4) provisions for system integrity via diagnostic testing and recoverability procedures.

6.1.2.1   TEL

TEL is the executive service subsystem which conveniently interfaces the user to a variety of UTS monitor facilities as well as other UTS subsystems. In particular, TEL permits direct access

to most of the functions associated with program development in FORTRAN or assembly language. These include:

- Create or modify source program text files (EDIT)
- Compile or assemble source programs into relocatable object modules (FORTRAN IV, META-SYMBOL)
- Link relocatable object-program modules into load modules (LINK)
- Modify object module symbols (SYMCON)
- Load and execute object programs, including searching for library subroutines
- Execute object programs under control of one of the debugging systems (DELTA or FDP)
- Resume execution of programs that have been interrupted or stopped by the user for debugging purposes
- Save core image of program being executed and retrieve it subsequently for continued execution
- Copy and delete permanent files (PCL)

TEL also provides information services to the terminal users — such as his current session accounting charges and the status of available system resources.

6.1.2.2   SYMCON

SYMCON is a convenient tool for program development tasks in which smaller subprograms are put together to form a larger process. SYMCON enables the user to examine internal symbolic references to determine if there are any conflicts in naming between the various subprograms. The user can eliminate those internal names which are no longer necessary and he can change symbol names which may be in conflict.

6.1.2.3   LINK

The LINK subsystem forms executable program load modules from relocatable object modules. It is a one-pass linking loader used to combine a number of program elements into an executable

entity. LINK merges internal symbol tables of the object program elements and searches sub-routine libraries for external references.

6.1.2.4   SUPER

SUPER is available only to installation management for the purpose of displaying user account-ing statistics, defining, or modifying legal users for the system and deleting legal users from the system.

It may be used on-line or in the batch mode. In addition, various parameters may be set on an individual user basis, as follows:

. Maximum core allowed for all on-line users

. Maximum file space allowed for all on-line users

. Maximum number of on-line users to be conveniently served

. Maximum number of tapes allowed for all on-line users

. Batch bias - the percentage of execution time which batch receives

. Batch priority - less than or equal to compute-bound on-line users

. Normal quantum by which users are time-sliced

. Minimum quantum guaranteed each user

Using the information provided by system monitoring, UTS will allow installation management to adjust the above parameters using the CONTROL processor for optimum resource utilization. This process is called "tuning" the system.

Features that contribute to system flexibility include a terminal startup/shutdown facility, a wide variety of parameter adjustments for system tuning, a broad range of hardware configura-tions, and easy upgrading from existing Sigma Batch Time-sharing Monitor (BTM) systems.

6.1.2.5   Terminal Startup/Shutdown

UTS provides a convenient facility which enables the system to discontinue on-line service to terminal users and proceed to process only batch and real time operations. When this is done,

all of the user-allocated core memory is made available to batch and real time processing. This facility also allows on-line terminal operations to be restarted after shutdown so that conversational, batch, and real time processing can again be performed concurrently.

6.1.2.6   System Tuning

UTS installation management personnel can dynamically adjust a number of system parameters (i.e., tune the system) by using the CONTROL subsystem. For example: (1) the minimum and normal time quanta can be dynamically adjusted; (2) the minimum percentage of CPU time to be allocated to batch processing can also be dynamically adjusted to insure that a desired level of batch throughput is maintained; and (3) the number of active time-sharing users which the system will service concurrently is another dynamically adjustable parameter. Reducing this number will improve the response time for those on-line users being serviced. The adjustment of these and other parameters will insure efficient allocation of the system's resources as its environment changes.

6.1.2.7   Configuration Flexibility

The UTS operating system can accommodate a variety of Sigma hardware configurations, depending upon installation needs. The following are key elements that are used to create the optimum configuration for any specific operational environment:

.   Separate storage systems with independent controllers for maximized file I/O and swapping. Separate controllers are recommended to avoid congestion between swapping and extensive file I/O activity

.   High speed RAD (Model 7212) for fast swapping to maximize the number of concurrent on-line users

.   Additional core memory for accommodating larger user jobs or more simultaneous users in core memory. UTS was designed to operate efficiently with multiple users in core; and this will require at least 80K of core memory if a variety of processors are in use

. Additional RAD storage for providing more permanent file space for users and temporary file storage for executing user programs. The XDS Extended Performance RAD (Model 7232) is recommended for this purpose. Additional storage capacity at greater economy is provided by the XDS Removable Disk Storage System (Model 7240/7241/7242).

6.2 XDS Software Systems

To enhance the application of XDS computers to business data processing, scientific processing, and management decision making, XDS offers a field-proven software package. This XDS developed software gives users a total problem solving and data handling capability throughout the entire range of business and scientific applications. The Sigma software package consists of:

. XDS COBOL

. FORTRAN IV Compilers

. BASIC

. Assemblers

. DMS

. MANAGE

. SORT/MERGE

. 1400 Simulator

. FMPS

. GPDS

. SL-1

6.2.1 XDS COBOL

The XDS COBOL compiler offers the user a powerful and convenient programming language facility for the implementation of business or commercial applications. The language specifications are based on the latest definition of COBOL and fully conform to the proposed ANS standard for the various functional processing modules. Only those language elements which

cause ambiguities or are seldom used have been deleted. The compiler's design takes full advantage of Sigma 9's unique features, resulting in rapid compilation of source code, rapid execution of the resulting object code, and the generation of compact programs. The result is a highly efficient programming system requiring a minimum amount of storage.

The following features summarize XDS COBOL advantages as an important tool for the programmer:

- Operates under UTS
- Maximum use of high speed, secondary storage (RAD) file
- Overlay organization for minimal core memory utilization
- Meaningful diagnostics
- High compilation speed
- High execution speed of object programs
- Compact object programs
- Extended language features

XDS COBOL for the Sigma 9 contains many facilities which provide flexibility and ease of use in program development. They include:

- Implementation of the table handling module
- SORT/MERGE linkage
- Sequential access
- Random access linkage
- Segmentation feature
- Report writer module
- Library utilization
- Calling sequences for FORTRAN, META-SYMBOL, and SYMBOL
- Common data storage for independently compiled programs
- Packed decimal arithmetic as well as floating point arithmetic formats
- Data name series options for ADD, SUBTRACT, MULTIPLY, DIVIDE, and COMPUTE verbs

The system provides the user with a comprehensive set of aids to minimize the time required to generate "bug-free" programs. Listings of these programs include: (1) the source language input to the compiler with interspersed English language diagnostic messages; (2) an optional listing of the relocatable binary output, printed in hexadecimal format with an alphabetic mnemonic operation code, and keyed by line number to the source language listing; and (3) a cross-reference listing indicating, by line number, where each data name or paragraph name is defined in the COBOL program, and each place where it is referenced. In addition, at run time, the user may use TRACE and EXHIBIT to follow execution of the procedure division.

The compiler is designed to take full advantage of high speed, random access secondary storage (RAD) file. This rapid overlay service means faster job execution because of minimized I/O delays and smaller core memory requirements.

6.2.2 Extended FORTRAN IV Compiler

XDS Sigma FORTRAN IV is the result of continuing research and development of the FORTRAN language. This system conforms to and accepts the language of the standard version of FORTRAN IV defined by the American Standards Association, as well as providing more complete capabilities and fewer syntactic restrictions.

XDS Sigma FORTRAN IV has been designed for compatibility and, as a result, contains as subsets the following FORTRAN languages:

- ASA, X3.4 Standard FORTRAN
- IOS, TC97/SC5 FORTRAN Proposal
- XDS 900 Series/9300 FORTRAN IV
- XDS 900 Series FORTRAN II
- XDS Sigma FORTRAN IV-H
- IBM 360 FORTRAN IV
- IBM 7090/7094 FORTRAN II and IV

The library routines in XDS Sigma FORTRAN IV, as well as the object code of subprograms produced by the compiler, are fully reentrant, allowing programs to function in a multiprogramming or real time environment. XDS Sigma FORTRAN IV operates under the UTS system.

The following features not found in many other FORTRAN systems characterize XDS Sigma FORTRAN IV:

.   Bit manipulation
.   Control of specialized I/O and interface equipment
.   Efficient coding of critical high speed loops
.   Absolute location references and other non-standard operations

Sigma Extended XDS FORTRAN IV has many special features that facilitate programming and operation. These include:

.   Meaningful Diagnostics: The compiler generally pinpoints detected errors and identifies the cause. Recovery from the error is often possible and the rest of the statement can be checked for further errors

.   Multiply Entry into Subprogram: An ENTRY statement within a SUBROUTINE or FUNCTION subprogram can be referenced by a statement referring to the ENTRY name

.   Expressions in Output List: Sigma Extended XDS FORTRAN IV permits any legitimate expression in an output list

.   Extended Assignment Statement: The assignment statement allows an expression to be assigned to one or more variable

.   Global Variables: Global variables can be used in place of COMMON. Separately compiled or assembled subprograms can refer to the same variable by name statements, thus omitting the order dependence inherent in the use of COMMON. This feature is unique to Extended XDS FORTRAN IV

.   Self-Identified Input/Output: This feature, which is not available in other systems, allows the novice FORTRAN programmer to write I/O programs immediately and the sophisticated user to minimize concern with I/O, since no FORMAT's need to be used

- <u>Extended Relational Operators:</u> Relational expressions can be written in a form consistent with the normal mathematical representation of inequalities, due to the availability of the symbols, ">" and "<"

- <u>Asynchronous Input/Output:</u> In Sigma Extended XDS FORTRAN IV, input or output of data can proceed in parallel with computation or other input/output. Binary information can be entered in a non-standard format

- <u>Memory-to-Memory Data Conversion:</u> In Sigma Extended XDS FORTRAN IV, BCD/BINARY FORMAT, conversion can be performed without any input/output. For example, a card can be read and the user can then DECODE the information in columns 8 through 72, selecting a FORMAT on the basis of a code in columns 1 through 7; or, a record to be output can be built up in sections at various times, while other I/O is going on, and then output whenever desired. This feature is not available in most other FORTRAN systems

- <u>End-of-File Processing:</u> This feature allows the user to specify the action performed when an end-of-file is read. Thus, file manipulation can be accomplished with Sigma FORTRAN IV

- <u>Generalized DO:</u> DO and REPEAT loops can run forward or backward with the control variable in fixed or floating point mode (single or double precision)

- <u>Generalized REPEAT:</u> The Sigma FORTRAN IV REPEAT statement is similar to the DO statement while having much more power. For instance, while the REPEAT range is being executed, the controlling variables and expressions may be re-evaluated

- <u>Generalized DATA Statement:</u> This feature allows subscripts and DO parameters to be expressions that involve outer DO-control variables. These DO loops can go forward or backward, or they can be executed numerous times. Constants of the wrong mode are converted if meaningful; otherwise, they are flagged with a diagnostic message. Repeated groups of constants can be nested to any depth. If there are too few constants, the list is repeated from the beginning, as in FORMAT statements

- <u>Variable Number of Arguments:</u> A subroutine can have a variable number of arguments from call to call by using multiple dummies

- <u>Internal Subprograms:</u> In other FORTRAN systems, the programmer must often place many variables in COMMON to communicate with external subprograms or use assigned and

<!-- XDS Xerox Data Systems logo -->

computer GO TO statements to simulate internal subprograms. In Sigma Extended XDS FORTRAN IV, subprograms can be compiled with each other and with the main program. They then have access to the same variables, constants, and FORMAT statements

. Widthless FORMATs: The input/output routines can determine the proper size of a number at execution time. This feature is unique to Extended XDS FORTRAN IV

. Adjustable FORMAT Specifications: The adjustable FORMAT statement, one of the most powerful features of XDS FORTRAN, saves time and effort when writing FORMAT statements which handle slightly different situations. Input records which are highly variable may be more conveniently handled

. Generalized Array Bounds: Sigma Extended XDS FORTRAN IV permits positive, negative, or zero upper and lower bounds; a lower bound of "1" is implied if no other bound is explicitly specified

. N-Dimensional Arrays: Arrays in Sigma FORTRAN IV can have any number of dimensions. Many other systems restrict the number of dimensions to three or seven

. Mixed Expressions: Numbers and variables of real, integer, double precision, complex, and double complex type can be mixed in any expression in the Sigma system

. Generalized Subscripts: Subscripts in Sigma FORTRAN IV can be any general expression and can be subscripted themselves

6.2.3 Extended FORTRAN IV-H Compiler

FORTRAN IV-H is a one-pass, high speed compiler that operates under UTS. It is designed for maximum compatibility with both ANSI Standard FORTRAN and IBM 360 H-level FORTRAN IV. XDS Sigma FORTRAN IV-H includes a number of features not found in ASA FORTRAN. Among these features are:

. ENTRY statement
. Double complex data
. FORTRAN II READ, PRINT, and PUNCH statements
. IMPLICIT statement
. END and ERROR options on READ statements

- T (tab) format
- NAME LIST input/output
- CONNECT statement
- Object program listing
- In-line assembly language option
- Reentrant library
- Reentrant object code
- In-line symbolic code
- Run-time debug trace of variable assignments under BPM/BTM
- Run-time debug trace of path-of-flow under BPM/BTM

The compiler tables, such as symbol and label tables, are dynamically allocated by FORTRAN IV-H to optimize memory usage. The FORTRAN IV-H compiler offers two kinds of source language debugging capabilities:

- A trace of the statement number of source lines reached during execution of a program
- A snapshot of the values stored into variables as the result of assignment statements

Complete provision is made for interface with both main programs and subprograms written in SYMBOL, MACRO-SYMBOL, or META-SYMBOL.

6.2.4   BASIC

BASIC* is a programming language designed to be easy to teach, learn, and use, allowing users with little or no programming experience to create, debug, and execute programs via an on-line terminal. Such programs are usually small to medium sized applications of a computational nature.

---

* Beginners All-purpose Symbolic Instruction Code was originally developed at Dartmouth College.

Two versions are available, one which operates under BTM, and a more sophisticated version which operates under UTS.

Highlights of both versions:

- Conversational
- Immediate Syntax Checking of input
- Can be used in both on-line and batch modes
- Editing and compilation/execution modes of operation
- Direct execution capability
- Fast in-core compilation
- Efficient object code generation
- Minimum compiler core requirements
- Safe/fast compile option
- Extensive print editing and formatting capability
- Extensive intrinsic functions and matrix operations
- Computer GO TO capability (multiple path branching)
- File input/output capability
- Alphanumeric constants
- Chaining facility
- Variable program size
- Compilation and run-time diagnostics
- Restore statement for re-use of data
- Multiple argument functions allowed
- No fixed limit to the number of constants

Features unique to UTS version:

- Chaining facility includes context preservation
- Character string operations
- All computations carried out in double precision

Many of the above listed features are not available in other versions of BASIC, or have been greatly expanded. Some BASIC features are described in more detail below:

. <u>On-Line and Batch Operation:</u> Provides complete flexibility of operation. Designed primarily for on-line program development and execution, or on-line development and batch production execution. Programs may also be developed in a batch mode to be executed on-line, or developed and executed in batch mode

. <u>Editing and Compilation/Execution Modes of Operation:</u> BASIC provides two user modes of operation. The editing mode is used for creating and modifying programs; the compilation/execution mode is used to run a completed program. This arrangement simplifies and speeds up the program development cycle

. <u>Direct Execution:</u> Statements may be entered via a terminal and immediately executed. Principal benefit — permits on-line debugging of programs under development without employing a special debugging syntax. During execution, programs can be investigated for loop detection, snapshots of variables can be obtained, values of variables can be changed, flow of execution can be re-routed, etc. This unique capability also allows an on-line terminal to be used as a "super" desk calculator. When operating under the UTS monitor, virtually <u>all</u> instructions are directly executable

. <u>Safe/Fast Compile Option:</u> At compile and execute time, the user may specify an array dimension check. In safe mode, statements are checked to verify they do not reference an array beyond its dimensions. In fast mode, this time-consuming check is not made. During checkout, the safe mode could be utilized; fast mode could be used to speed up execution when program reaches production stage

. <u>Print Editing and Formatting:</u> BASIC provides an image statement which utilizes a "picture" of the desired output format to perform editing, TAB capability, and a precision option to indicate the number of significant digits (6 or 16) to be printed

. <u>Alphanumeric Constants:</u> Capability allows the user to read, write, and compare variable alphanumeric data. This is particularly important for conversational input processing. This feature has been greatly expanded in the UTS version

. <u>Chaining:</u> Permits one BASIC program to call upon another for compilation and execution without user intervention. Thus, programs which would exceed user core space can be

segmented and overlay techniques employed via the chaining facility. Under UTS, the chain link option preserves array and string values across chaining

Features unique to UTS version:

. Character String Operations: Character strings up to 85 characters long (SYSGEN parameter) may be assigned, referenced, concatenated, compared, and used in input/ output statements. Strings can be .cleared separate from clearing arrays

6.2.5 META-SYMBOL Assembler

This program is a high level symbolic assembly language processor, compatible with all XDS compiler systems. META-SYMBOL features and facilities compare favorably with those of the most advanced operational assemblers on the largest computing systems. At the same time, it includes as a subset one of the easiest assembly languages to learn and to use.

The language includes general expressions of one or more items, combined by arithmetic and/or Boolean operators. Items can be single elements or lists. META-SYMBOL also has Function (FUNC) and Procedure (PROC) capability, permitting the programmer to specify and use in-line, non-machine operations in symbolic programs without the time penalty imposed by a remote closed subroutine.

Operationally, the system consists of an encoder and a translator. The encoder compresses the source language to about one-tenth its original size before the translator processes it. The translator enables source language modification with optional recovery of the resultant updated source code.

The most important advance in META-SYMBOL is the ability of a procedure to generate coding, dependent upon conditions at assembly time. For example, a generalized subroutine, or MACRO, for a series expansion must always calculate all elements and multiply all coefficients. A general use procedure, on the other hand, can generate conditional codes based on the number of coefficients and the use of those coefficients and constants. If a coefficient is unity in a

specific case, then the associated multiplication can be eliminated in the generated code. If a coefficient were zero, then deletion of an entire term would result. The number of terms for which coding is generated depends upon the number of arguments supplied at assembly time. This capability obviously increases the efficiency of library programs for general use. Coding is generated in-line and is called and provided with arguments with a single call line, exactly as a MACRO. Procedures of general value can be put into the system, then automatically called by name at assembly time.

The following features summarize META-SYMBOL advantages as an important tool for the programmer:

- The operand field can contain both arithmetic and logical expressions, using constant or variable quantities
- Longer programs can be written in segments and the loader automatically links references from one segment to another
- The DO directive allows generation of one line of code many times with address incrementing. In PROCedures, the DO allows selective generation and skipping of areas of code, with parameters as constants, or expressions determined at the time of the particular assembly
- PROCedures allow a MACRO assembler capability of generating many units of code for a given PROCedure call line. Further sophistication provides completely parameterized coding, with PROCedures applicable to many programs
- META-SYMBOL permits full use of lists and subscripted elements
- The call line and its individual parameters can be tested both arithmatically and logically
- A subset of code in the PROCedure can be conditionally generated according to values of given parameters
- META-SYMBOL incorporates definition of mnemonics at a compiler level or redefinition of existing standard XDS machine mnemonics
- The use of dummy parameters is dependent upon the structure of the call line rather than strictly on a name basis
- Nested PROCedures can be used and one PROCedure can call another

- Complete use of arithmetic and Boolean operators is possible in PROCedures
- FUNCtions resemble PROCedures in structure, except that they utilize non-generative code to return values
- Use of the TEXT directive simplifies coding of messages and eliminates the need for character or word count by using delimiters. In addition, Hollerith text can be generated in any DATA statement
- I/O procedures remove the necessity for bit-shuffling in creating I/O instructions of complex structure

META-SYMBOL capabilities, beyond those normally associated with the basic derivatives, can be brought into use gradually as the need dictates and ability allows. Coding practices such as parametric programming and PROCedures and FUNCtions can be phased into use as the learner's competence increases.

6.2.5.1   SYMBOL Assembler

The XDS SYMBOL assembler, which is basically a subset of META-SYMBOL, accepts symbolic input from various media and translates standard XDS Sigma mnemonics and symbolic expressions. It also offers pseudo operations that aid the user in coding and debugging programs. The SYMBOL linkage format is compatible in every respect with the other XDS compilers and assemblers.

6.2.6   Data Management System

A Data Management System (DMS) data base can form the heart of an integrated applications system or a total corporate information system. It is designed to economically extend user capability to handle dynamic system development.

6.2.6.1   DMS Concepts

Elements of pertinent information may be connected by logical pointers within the data base. This allows data to be recorded just once and yet be linked in many ways to other data to permit its use by diverse applications. Information required by separate operating areas of business

may now be retrieved from one common source, thus assuring that all concerned are using the same data. This capability can also eliminate redundant data from files and, in so doing, remove the source of many potentially costly errors. In addition, it can eliminate to a large degree the traditional requirements of sorting and merging to manipulate files.

DMS enables one to create a data base according to relationships among data which are user-defined. These data relationships are established using pointers from one record to another throughout the data base. Because of these pointers, data may be structured and information retrieved in virtually any way to suit individual application requirements. Users control the physical placement of data, through a variety of options that are available, to optimize the efficiency of application processing.

The DMS data base can be structured to support batch processing of data while facilitating transaction oriented processing as well. As a transaction occurs, it quickly updates all records affected in the data base. With this capability, on-line systems can be developed and tested locally, adding terminal access to the data base when the system has been proven. Inquiry of the data base now becomes a natural mode of operation, although standard reports may be produced as always.

By developing and using the DMS data base in a way that reflects individual information flows, users are able to anticipate future system requirements for data. This will allow one to reduce the amount of on-going systems redesign effort in the more dynamic application areas.

The data base can be created or accessed by the programmer using COBOL, FORTRAN, SYMBOL, or META-SYMBOL. The programmer is also provided with a high degree of independence from the data base to facilitate additions or alterations to data with a minimum of program changes. Additionally, DMS relieves the user of the I/O programming burden by performing file processing and data base manipulation for him.

To protect the data base from unauthorized access and potential failure, DMS provides comprehensive access security. Access keys may be required to either read or write the data base, records within it, or even specific fields within a record if desired. For example, a payroll

record may be made generally available, but access to rate and salary grade fields within that record may be locked out to users without the proper level of authority. Additional protection is provided by a journal file where a copy is kept of all records as they appear, both before and after they are updated in the data base. This audit trail ensures ease of recovery from potential failures.

6.2.6.2   DMS Facilities

. File Definition Processor: This is free-standing and operates under control of the monitor. Data is defined by using a data definition language which becomes input to the file definition processor. Data is defined and retrieved in lists and hierarchies, with these two relationships combined to allow extensive networks of data to be used. There is no inherent limit to the number of relationships that can be defined for any one type of record

. Data Base Manager: Functions to service user program requests to access the data base. Invoked by ENTER (COBOL), CALL (FORTRAN), and a BRANCH AND LINK instruction (META-SYMBOL). These calls instruct DMS to add, delete, update, and retrieve data from the data base.

. Utility Routines:
  - File Initialization - used to format the device and prepare it for receiving data
  - File Dump Routine - provides the capability of dumping all or selected portions of an existing DMS data base to a sequential file and/or to a printer
  - Data Base Restore - provides the capability of reloading all or selected portions of an existing DMS data base from a sequential file. Data base restore, used with the Journal Tape, facilitates data base reconstruction if necessary

6.2.6.3   MANAGE

Sigma MANAGE is a generalized file management system for the XDS Sigma 9 computer. It fulfills a long standing business need for a greatly simplified method of using the computer to establish and maintain records on magnetic tape and/or RAD units; selectively retrieve data from these records; and, upon request, generate reports from the contents of these records in a

variety of formats. Sigma 9 MANAGE eliminates the need for specialized programming for these tasks. In contrast to COBOL, which is a language for the programmer, MANAGE provides personnel outside the data processing department with direct and immediate access to information processed and retained by the computer.

Sigma 9 MANAGE functions can be performed without the lengthy time delay common to the implementation of computer programs. Sigma 9 MANAGE has the unique capability of enabling personnel who lack skill in programming to communicate directly with the computer. The full potential value of the computer is thereby approached. The response time to changing management demands for information is reduced from days, or even months, to a few hours.

A focal point in the Sigma 9 MANAGE file management system is the Data File Dictionary, which describes precisely the format and characteristics of a particular data file. Since each Data File Dictionary relates to only one file, an installation has as many dictionaries as there are data files. These dictionaries, along with the generalized MANAGE programs, are retained in the Sigma 9 software library.

The programs that constitute MANAGE can be classified as LOAD and GO file processing compilers. Each is a generalized program oriented around a Data File Dictionary that describes the format of one specific data file. Utilizing control parameters and applicable Data File Dictionaries, these programs perform file creation and maintenance, selective data retrieval, and report generation.

File creation entails setting up a new file. File maintenance entails modifying the contents of an existing file by inserting, deleting, and/or changing records within the file. Data retrieval functions include selecting records (or parts of records) from a file; the selection depends upon a set of criteria that are related by the logical operators and/or their logical negatives. The criteria tests include greater than, less than, and equal to. Data retrieval queries involving files can be processed singly or in batches. Thus, multiple reports (programs) are generated in a single pass of the file, providing automatic multiprogramming of the data base.

The report generation function includes editing of selected data fields into prescribed format and sequence, automatic inserting and alignment of column headings, counting of items within groups, and summarization of data within the report.

6.2.6.4   SORT/MERGE

The XDS SORT executes on the Sigma 9 under control of the Universal Time-sharing System. The input file to be sorted and the resulting sorted output file may utilize any sequential peripheral device (e.g., magnetic tape, disk, RAD, cards, printer, etc.). Intermediate storage can utilize magnetic tapes, disks, RAD's, or any combination of these devices. Complete flexibility of input and output is available. Minimum configuration depends upon the user's choice of input/output services.

The Sigma 9 SORT can be used as a separate run in a batch processing mode, or it can be called by other processors (e.g., COBOL, FORTRAN, or MANAGE), or by assembly language programs. When called in this manner, SORT can checkpoint the host program and utilize the additional storage thus obtained. This permits it to function very efficiently on smaller systems.

The Sigma 9 SORT is designed specifically to make maximum use of the system hardware. The most sophisticated sorting techniques such as replacement selection sorting and read backward polyphase merging are used to create a flexible and highly efficient sort.

The basic generalized SORT is specialized by parameters. These parameters indicate the hardware available, define the file format, indicate the presence or absence of user own coding, select alternate methods of handling errors, define sort keys, indicate if the collating sequence is to be changed prior to the SORT, and specify the direction of the SORT (i.e., ascending or descending).

The first phase of the SORT checks the parameters and optimizes the available hardware. The second phase is the internal SORT which uses a replacement selection tournament technique to increase the speed for the SORT. The third phase uses a polyphase technique to combine the strings generated in the second phase into fewer, but larger strings until only one string (the final sorted file) remains.

During the SORT, six own code exits are provided to enable the user to perform functions such as reading, checking, and writing header and trailer lables, and to modify the data records prior to or after the sorting, but prior to the final merge.

SORT can process multi-reel files with as many as 36 reels per file.

As a separate function, two or more sorted files can be merged using the XDS MERGE program which operates on the Sigma 9 under control of UTS. The input files to be merged and the resulting merged output can utilize any device available in the system.

XDS Sigma 9 MERGE can combine up to eight input files into one merged output file. Files being merged need not reside on the same type of input device. For input files residing on magnetic tape, the MERGE program accepts multi-reel input files and produces multi-reel output files where required. Furthermore, files which reside on more than one physical RAD and/or disk can be merged under control of UTS, which determines the multiplicity of files.

As in SORT, the MERGE program can be specialized by the use of parameters. These parameters define the input/output, specify the handling of header and trailer labels if present, specify the presence or absence of user coding, select error handling options, define the merge keys, indicate the ascending or descending sequences of the MERGE, and specify the collating sequence of the merged files.

MERGE is one unsegmented program that performs internal merging without the need for intermediate storage. User exits are provided to allow the reading and verification of header and trailer labels (if present), label modifications, and data record modification.

The generalized XDS SORT/MERGE package uses improved techniques designed to take advantage of special hardware features such as decimal arithmetic, byte string manipulating instructions, the read backwards feature of 9-track tape drives, and random access devices such as disks or RAD's.

Major features include the ability to process variable length records in monitor format as well as fixed length records in either blocked or unblocked format.

6.2.6.5   1401 Simulator

A 1401 Simulator package minimizes the problems otherwise associated with converting 1401, 1440, and 1460 programs; the simulator provides a compatible level of computing capability, while users are writing improved programs that take full advantage of the greater speed and power of the Sigma 9. The 1401 Simulator runs under direct control of the monitor system. Hence, 1401 programs can be intermixed programs written in any other Sigma language.

6.2.6.6   FMPS — Linear Programming and Package Option

The Sigma Functional Mathematical Programming System (FMPS) is a modular linear programming package that provides mathematical optimization techniques to be used for allocation of resources such as manpower, capital, or equipment.

Some typical problems which submit to solution by FMPS are:

  . Blending
  . Rqw material selection
  . Production capacity allocation
  . Transportation and distribution
  . Inventory Optimization
  . Investment allocation

FMPS uses the high speed auxiliary storage provided by XDS Rapid Access Data (RAD) storage units, as an extension of core memory. Average access time of these fixed-head, high speed units is only 17.5 milliseconds, with transfer rates up to three million bytes per second. Some characteristics are described below:

  . It offers simplified data preparation using the GAMMA III Matrix Generator/Report Writer
  .. It is designed in a modular fashion to give each user only the capabilities he really needs . . . . and to allow for expansion as the user's needs grow

- Time critical sections are written in assembly language to take maximum advantage of unique Sigma hardware features

- Through use of an easily understood, user-oriented control language (a subset of FORTRAN IV), the user has complete freedom to structure his optimizing runs

- FMPS can operate under the XDS Batch Processing Monitor (BPM), Batch Time-sharing Monitor (BTM), or Universal Time-sharing System (UTS) . . . . either at the CPU site or at remote locations via remote batch terminals

- It is economical - with the ability to capitalize on the performance of the RAD unit, the average linear programming user (300-row problem) achieves a price/performance ratio unequalled on other third-generation computers. For each incremental 4K words of core memory made available to FMPS, the maximum problem size is increased by 225 rows

6.2.6.7   FMPS Components

The Sigma 9 FMPS package includes two major components: (1) the basic linear programming system, and (2) the GAMMA III Matrix Generator/Report Writer. The basic system can be used either alone or with GAMMA III.

The Basic System — The basic linear programming system consists of a set of individual procedures, or subroutines (which constitute the basic LP solution algorithm), plus a control language.

The control language is used to direct FMPS through its problem-solving sequence. It provides for procedure calls, data management, and arithmetic and logic testing.

By means of the control language, the user calls in the individual procedures needed to input the problem in matrix form, solve the problem, and output the solution in matrix form. These procedures are of four types: input, optimizing, output, and preservation/restoration. The FMPS linear programming algorithm, which these procedures perform, includes capabilities for multiple pricing, upper and lower bounding, and range constraint. The inversion technique uses a matrix triangularization scheme which is one of the most advanced in the industry.

GAMMA III Matrix Generator/Report Writer — GAMMA III, another optional feature available to FMPS users, provides a simplified, time-saving aid in formatting input data and in producing reports in management-oriented formats.

The GAMMA III Matrix Generator accepts problem-oriented input statements from management in the form of data tables (two-dimensional arrays) and lists (symbolic references that describe the variables and constraints to be included in the current representation of the problem); and automatically formats these statements in the matrix form required for solution by the linear programming algorithm.

The GAMMA III Report Writer prepares reports of the solutions to FMPS problems with full English language titles and in desired format. In the report publication phase, GAMMA III provides controls that enable the user to call for all reports described in the report generation phase or for any desired subset.

6.2.6.8 GPDS (General Purpose Discrete Simulator)

GPDS represents the principal thrust of XDS into the discrete simulation field. In structure, it is similar to the IBM discrete simulator, GPSS, which is the most widely used programming language in the discrete simulation area. GPDS, however, possesses features which are not indigenous to GPSS. These features enhance the attractiveness of GPDS to the simulation community since GPSS/360 then becomes a subset of GPDS.

GPDS is a transaction flow oriented simulation language which functions in a discrete time frame; i.e., events occur at precise measurable instants of time rather than continuously. A feature which is unique to GPDS (and GPSS) among simulation languages is GPDS possesses a unique block diagram scheme which makes it very attractive to modelers and system analysts. GPDS is an extremely versatile language, proof of which lies in the fact that GPSS has been applied to almost every conceivable situation. Among heavy users of GPSS are: (1) the banking and finance industry; (2) manufacturing; (3) educational institutions; (4) hospitals; (5) corporate planners; (6) information system analysts; and (7) urban planners.

One of the principal problems which analysts have encountered with GPSS/360 has been core size. GPSS requires large amounts of core and, thus, if an analyst is dealing with a large system he often runs into trouble with core availability. GPDS has largely eliminated this difficulty through frequent and efficient utilization of random access devices. Among the additional features which GPDS incorporates are:

- Matrix savevalues resident on RAD
- Byte sized savevalues
- Library feature for matrix savevalues
- Multiple initialization of arrays
- Exponential operator
- SNA for absolute clock time
- Text card continuation
- Control over assembly sets
- Alternate method of indirect specification
- Built-in statistical distributions
- More job tapes
- Square root operator
- Inter-transaction communication
- Steady state terminator
- Print columns 73-80 on input
- Premature termination statistics with transaction dump option

- Unlimited random number chain

- Separate DCB for report generator

- FORTRAN/COBOL interface

- Additional parameter and savevalue types

- Blocks stored on a RAD

- Parameters stored on a RAD

GPDS fills a major need of many users who now are recognizing the requirements for this type of software. Its availability establishes XDS as a firm competitor in the operation research field.

6.2.6.9   SL-1

Simulation of continuous and discrete time systems in the scientific and business community is an accepted tool in the solution of both classes of problems. For use with continuous time simulations, XDS provides SL-1. SL-1 is a superset of CSSL*, the standard language specified by Simulation Councils, Inc.

SL-1 greatly reduces the training requirements for programming personnel. It is designed primarily to simplify the work of scientists and engineers who lack special training in programming but who must program a digital or hybrid computer to simulate parallel systems. By removing the communications barrier between man and machine, SL-1 allows the user to take full advantage of the speed and real time attributes of the XDS Sigma 9 computer system. While it serves as a simple, easy-to-use programming tool for the novice user, SL-1 also offers sufficient flexibility and power to aid the sophisticated programmer faced with a major programming task.

The primary function of SL-1 is solving differential equations, a fundamental procedure in the simulation of parallel, continuous systems. To perform this function, SL-1 provides six integration methods and control logic for their use. It automatically sorts the equations to ensure that parallelism is maintained and it includes an implicit operator to handle algebraic loops.

---

*Continuous System Simulation Language

A major feature of SL-1 is its extensive set of macros, which allow the user to simulate a wide variety of linear and non-linear elements through the use of simple, single-operator statements. These prototype statements are inserted into the program each time the macro is referenced by name.

Because of the versatility of XDS Sigma computing systems and the broad applicability of digital and hybrid simulation techniques, applications for SL-1 are essentially limitless. Its macro library concept, which contains a basic XDS macro set, allows the user to define his own macros. Primary current areas of application are aircraft simulation, biomedical models, chemical processing, control system design, fluid-flow analysis, heat-transfer studies, and missile and space systems design.

SL-1 fulfills a two-fold purpose: It facilitates programming tasks for the novice user, and it makes available to the well-trained programmer the flexibility and power of a versatile simulation language.

For the novice user with a simple, straightforward task, SL-1 offers:

- An easily understood form for model-description statements
- A set of operators capable of handling most problems involving differential equations in a simple manner
- Built-in integration algorithms and pre-packaged, standard input/output
- Ability to interactively alter the parameters of a simulation problem between runs
- A complete arbitrary function-generation system
- The proven features of previous, well-accepted simulation languages including sorting of operations to assure proper order of the calculations; expression-based representation resembling that of MIMIC*, DES-1, and DSL/90**; and optional block-oriented representation resembling that of MIDAS and DES-1.

---

* An extension of MIDAS, developed by Wright-Patterson Air Force Base
** A proprietary IBM software package developed for the IBM 7094 computer

The power of SL-1 as an advanced simulation language for the sophisticated programmer results primarily from the following features:

- Open-ended operator set, including powerful macro procedures that enable the programmer to generate his own operators
- Hybrid and real time capabilities useful in a broad spectrum of application areas
- Hybrid and real time features include interrupt operators; clock operators; hybrid input/ output; automatic synchronization with real time; block timing, debugging aids; provisions for simultaneous sampling of A/D lines, simultaneous D/A conversion, and analog check and control; and capability for generating real time solutions
- Conditional processing features to aid in program checkout, permitting the user to include test cases and associated dumps without altering the source program

Macro Capability — An additional feature of SL-1 is its macro capability; macro defines a new simulation operator for the duration of the program in which it is used. A macro can contain any SL-1 statement and other previously defined macros.

A macro is utilized in the following manner: A set of generalized statements define the macro. When invoked, the macro variables are replaced by those in the calling sequence and in-line statements corresponding to those in the macro definition are generated.

An extensive macro library is available to all SL-1 users. It includes the following:

- First- and second-order transfer functions
- Comparator
- RST flip-flop
- Implicit iterative function
- Single-step delay
- Variable delay
- Limiter
- Derivative function

- Dead space

- Hysteresis

- Quantizer

- Resolver

- Step function

- Ramp function

- Pulse generator

- Harmonic wave generator

- Normal-distribution noise generator

- Uniform-distribution noise generator

- Function switch

- Input switch

- Output switch

- Lead-lag

- Mode-controlled integrator

The user can define his own macros, as the need arises, as well as using those in the XDS library. Use of macros drastically reduces programming time, increases efficiency of execution, and enhances the flexibility of the language itself.

6.3   OTHER PROGRAMS

6.3.1   FDP (FORTRAN Debug Package)

The FORTRAN Debug Package provides the conversational user with many powerful features to reduce checkout time. These features, which are dynamically controllable from the terminal at execution time, include the following:

- Program execution control

- Statement stepping

- Conditional breakpoints

- Data breakpoints

. Flow tracing and history recording

. Examination and correction of scalar and array elements

. Branching

. Program restart

. Statement cancellation

. Argument display and automatic checking

In FDP, reference can be made to statement numbers, statement labels, and data elements. Statement and data breakpoints can be inserted so that when pre-selected situations occur, various options are made available. These options consist of stored commands which modify statements or data, display data values, branch, trace, etc. Also available at breakpoints are direct commands which allow for manual modification and examination.

Dynamic control allows the user to trace the history of execution and current status of his program, as well as direct the course of future operations via stored commands.

6.3.2   EDIT

EDIT is a conversational line-at-a-time context editor designed for creating, modifying, and searching data text files. EDIT takes advantage of keyed-file, random access storage for efficient and responsive file manipulation. EDIT functions include:

. Creation, insertion, deletion, re-ordering, and replacement of text lines or groups of
   lines

. Selective printing and renumbering of text lines

. Searching text files by context for matching, deleting, moving, and substituting line by
   line

. New file creation, file copying, and file deletion

. Intra-line editing for convenient modification of portions of a line, including shifting the
   text left or right

. Setting software tabbing controls for text formatting

6.3.3 PCL (Peripheral Conversion Language)

PCL is a media conversion service for moving files of data in various forms from one type of peripheral device to another. It operates in the conversational on-line mode as well as in the batch mode. PCL provides comprehensive facilities that allow the user to:

. Transfer single or multiple files

. Select specific records within a file for sequencing formatting and conversion

. Delete files

. List or dump files

. Call for magnetic tape handling functions

. Copy files to devices (e.g., line printer, card punch, etc.)

6.3.4 DELTA

DELTA is a powerful conversational debugging service for checking and modifying assembly language programs. It permits the use of full symbolic references in an object program to perform the following functions:

. Examine, insert, or modify various elements of a program (instructions, numeric values, encoded information, etc.). Data may be referenced in all types and formats

. Control program execution with insertion of conditional breakpoints into a program, and breakpoint based on changes in elements of data

. Trace execution by displaying information at designated points in a program

. Searches programs and data for specific elements and values

UTS assemblers and compilers generate symbol tables, describing data representation and program location. These tables can be used by DELTA to provide full symbolic referencing.

6.3.5 SYSGEN (System Generation)

SYSGEN permits the Sigma 9 user to generate a monitor system that is specifically designed for his installation, configuration, and requirements.

From a master file, plus any corrections or updates, the system generation program generates an installation-oriented system for a specific configuration. Because it contains only those elements actually needed, rather than all possible peripheral elements, the installation system provides faster, more efficient operation.

6.3.6   XDS Standard Library

The XDS standard library contains appropriate programs of standard mathematics, conversion, and special functions. The library includes those programs required for use with the XDS FORTRAN IV compilers. All library routines may be called by machine language programs. In addition to the standard library, the user can, at system generation time, place any special purpose routines from the XDS Users' Group Library or of his own creation into the standard library.

6.4   HARDWARE AIDS

The integrated Sigma 9 programming system makes full use of the many available hardware features:

.   Push-pull stacks permit dynamic space allocation, subroutine communication, and re-entrant capability

.   Traps are available for simulation of optional instructions not physically present and for error conditions

.   The analyze instruction assists in effective address calculations

.   The interrupt instruction provides increased effectiveness and speed of calculation

.   Call instructions are useful for requesting monitor functions and implementing user-defined subroutines as if they were machine instructions

.   Direct addressing of all of memory simplifies generated code

.   Displacement indexing minimizes the use of general purpose registers and facilitates programming

.   The Rapid Access Data (RAD) file is available for system storage, symbiont buffering, user storage, and other functions

Sigma software exploits the fast access time (17 milliseconds) and high transfer rates of XDS Rapid Access Data (RAD) files. The high speed and large storage capacity allow the RAD to be considered an extension of core memory. Thus, powerful operating systems can be implemented without severe core memory requirements by storing large portions of the operating system on the RAD. Those portions can then be loaded into memory as required very rapidly and overlaying an unused portion of the operating system. This same advantage is obtained when executing large programs.

6.5 XDS USERS' GROUP

As a continuing service for its customers, XDS offers participation in its Users' Group. The primary purpose of this user organization is to facilitate the exchange of technical information between XDS and personnel from installations that have procured computer equipment from XDS.

The significant objectives of the Users' Group are:

. To advance the effective use of XDS computers and systems
. To reduce redundant effort among its membership
. To establish and maintain active standing committees and special interest groups
. To provide channels of communication to assist in the continual interchange of relevant information
. To provide unified feedback to XDS regarding hardware and software needs

In support of these objectives, the Users' Group holds regularly scheduled international meetings bi-annually. Less formal regional, district, and sub-chapter meetings are held periodically, which the Users' Group also endorses and actively encourages.

Another important vehicle of user communication is the monthly publication, Users' News. This newsletter, compiled, printed, and circulated by XDS, keeps customers informed about new hardware and software product developments and schedules, profiles of user installations, new Users' Group members and applications, technical hardware tips, programming techniques and tutorials, standing committee reports, user field problem reports, new publications available, new Users' Group Library programs, and user project or application descriptions.

The Users' Group also distributes updated listings of the Users' Group Program Library, as well as the Standard XDS Program Library list, the installation membership list, standing committee questionnaires, and meeting agendas and proceedings.

Membership in the XDS Users' Group is complementary, both for installation members and individual members, by submitting a written application to the Secretary of the Users' Group. An installation member is the individual most closely associated with an XDS computer installation who is willing to participate as the official delegate of the installation. Any other interested individual within a given installation who is willing to participate in the activities of the Users' Group can apply as an individual member.

The Users' Group is governed by a 3-man Executive Board, composed of a Chairman, Vice Chairman, and Secretary. Each year, the Vice Chairman is elected from the most knowledgeable, experienced, and active members of the Users' Group. After one year in this capacity, the Vice Chairman automatically becomes the Chairman.

Participation in the XDS Users' Group permits the customer to share in hardware and software refinements developed and operated in many different applications, and further makes available the experience of many seasoned XDS computer users.

SECTION IX

SUPPORT

9.0 INTRODUCTION

Included in this section are procedures for arranging Sigma 9 demonstration and benchmark services, and a list of all responsible Sigma 9 personnel (with primary and secondary contacts), including their areas of responsibility.

9.1 DEMONSTRATIONS

All requests should be directed to Dave LeFort, Customer Relations, allowing a minimum of one week advance notice of the demonstration. The priority of objectives to be attained in the demonstration should be provided in detail along with the request.

- Sigma 9 RBM, BPM, and BTM demonstrations will be available for customers starting 2Q71
- Sigma 9 UTS demonstrations will be available for customers starting 2Q71
- Sigma 9 XOS demonstrations will be available for customers starting 3Q71

9.2 BENCHMARKS

Sigma 9 benchmarks will begin 2Q71. The Regional Systems Manager should coordinate his region's benchmark submissions and notify Systems Evaluation of his approval. He should ensure that the following is available with each benchmark submitted:

- A benchmark submittal form
- Listings of the benchmark and its output. (If listings and output are not included, outside computer time will be purchased and charged to the appropriate sales office to obtain them.)
- Objectives of the benchmark (e.g., speed, diagnostics)
- Any restrictions on modifications that can be made

- Estimated running times for competitive systems
- Formats of any foreign tapes needed
- Turnaround time available for the job

When necessary information is unavailable, all work on those programs will be suspended and the Regional Systems Manager will be made aware of the problem.

The Regional Systems Manager will have the responsibility of prioritizing the regional benchmark workload as the load in Systems Evaluation builds to capacity. Where significant proposal or benchmark work is necessary, District and Regional assistance may be necessary.

A summary of information gathered from benchmark activities will be made available to the field in the form of:

- Competitive timings
- Collections of biased benchmarks
- Periodic reports of pertinent problems encountered
- Conversion shortcuts
- Collection of easy-to-use demonstration programs

9.3   RESPONSIBLE SIGMA 9 PERSONNEL

The following people are responsible for the designated Sigma 9 areas:

Area	Primary Contact	Alternate Contact
Configurations	Jay Tavarozzi	Jim Harmon
Pricing	Phil Carley	Mike Gingrich
Contract Services	Jerry Kaminsky	Carl Schroeder
Competitive Information	Kent Cootes	Frank Yee
Plant Visits and Demos	Dave LeFort	Ron Guilbert

Area	Primary Contact	Alternate Contact
Benchmarks	Ed Owens	Burt Brown
Peripherals	Rick Dural	Ted Charter
Display Products	Dave Peltz	Jim Harmon
SIU's	Jay Tavarozzi	Ted Charter
Communications	Dick Gillen	Charlie Mitchell
UTS	Dave Escoffery	Dick Leslie
XOS	Tom Pruitt	Dick Leslie
File Management	Roger Benson	Dick Leslie
Data Management	Wylye Robertson	Leon Carlisle
FMPS and SL-1	Ed Jones	Charlie Jannasch