XOS FACT SHEET

August 10, 1972

XOS Development Section

INTRODUCTION

The XOS design has been optimized for direct replacement of IBM DOS installations. Using the hardware architecture of Sigma, the system has been implemented to minimize system overhead thus enabling high performance real-time and Timesharing support to be implemented within the same structure. The resulting multi-use operating system is unique in the industry. The major design features which have received special attention are:

- File data management system superior to DOS and competitive to OS
  - IBM compatible data organization
  - Data cataloging on the volume with the data
  - Ability to specify Block size as a data set attribute
  - Generation data group support
  - Data access control for security and integrity
  - User control of access methods, blocking, buffering and error handling through JCL or program control
- Ease of use and conversion
  - IBM compatible file formats
  - Assisted access methods
  - Simple yet flexible JCL
  - JCL cataloging capability
- Communications Network Support
  - TAM device independence
  - Multi-drop message mode
  - Point to point character mode
- Performance
  - Multi-buffered I/O
  - Frequency of use non-resident monitor handler
  - I/O scheduling by either priority or minimum disk arm movement
  - Resource management within the job scheduler process selects job mixes for multiprogramming throughput
  - Real-Time design which supplies less than 3 milliseconds response to the typical interrupt
  - The job and task scheduling system takes advantage of Sigma unique interrupt structure to provide a truly event driven operating system
  - Full memory map support
  - Minimum system size 32K
- Reliability and Maintainability
  - Ability to reconfigure system, with devices that are failing off-line, and continue operation or to logically switch devices
  - Modular construction for ease of maintenance
  - Complete technical, functional and design level documentation
  - Debug facilities
  - Patches do not have to be loaded at each system boot
- Timesharing compatibility with Batch
- Real-Time support
- Fast (30 min.) on-line SYSGEN
- Future development

The system is operating in 35 production installations in France and 2 locations in the States. It is considered to be a highly reliable and usable system. Recent visits by a prospect (Lummus) to the XOS accounts in Europe resulted in a very favorably impressed prospect.

Job Classes

The system provides eight separate job classes, all of which may be run simultaneously if adequate resources are available. The scheduling and dispatching priorities are established at will during system generation. However, the normal assignment of priorities is as defined in the sequence below.

The classes are:

Foreground (Class F)	Any number of foreground jobs may be run simultaneously. These are typically operator initiated real-time tasks.
Parallel (Class P)	Any number of jobs may be run simultaneously. These are typically operator initiated utility jobs.
Production	(Classes A, B, C, D, E and T) in which jobs are typically user production or test jobs.

Super Jobs

A feature of XOS is the ability of the user to chain several related jobs of one production class (other than T) into a superjob. The series of jobs constituting a superjob are executed sequentially. Each is executed only upon the proper completion of the preceeding job. If any job of a superjob aborts, all remaining member jobs are ignored. Superjobs may communicate with each other via the job switchboard.

Monitor Residence

The monitor is organized in two parts (with respect to memory residence); a small resident monitor that remains in memory at all times, and a nonresident portion that resides on secondary storage and is brought into memory as needed.

The XOS monitor is divided into resident and nonresident portions. Relatively few of the monitor services are required frequently enough to justify being made resident; the majority are made nonresident, thus saving space for additional user tasks.

The nonresident monitor is physically divided into a number of elements that are independently loaded into memory as required. When one of these elements that was loaded into memory is no longer in use, it remains in memory, but is marked "disengaged". The resident monitor maintains statistics on the frequency of use of these "disengaged" elements, and when additional memory is required, the least frequently used element(s) are overlaid by the program or element that requires space. Using this technique, the system is able to make the most efficient use of "unused" memory and significantly reduce the number of requests for loading nonresident monitor elements.

System Device Residency

The XOS system may reside on either RAD, or disk packs or a combination of RAD and disk packs.

Job Scheduling

When the scheduler is called, it always begins by examining the queue of waiting parallel jobs. All parallel and foreground jobs will be scheduled before any of the other classes are examined. The first job in each of the remaining class queues will be scheduled if: (1) the resource profile associated with the job can be satisfied from the list of system resources currently available for reassignment, and (2) no job from that class is currently active. The scheduler continues examining the first job in each queue until it encounters a job whose resource requirements cannot be satisfied or until it runs out of jobs to examine. When this occurs the scheduler ceases to search the queues and dismisses itself to the idle state. The only case in which the scheduler will scan beyond the first job in the queue, for which resources were not available, is if the job is in the Production class T. In this case the scheduler will examine the rest of the T class and, if a job is found which can be executed, it will be scheduled. This search of the T Production class is made in the order of the job priority given by the user on his JOB card. Multiple T jobs may run simultaneously.

Job Step Scheduling

Within jobs, job-steps are scheduled serially for execution based on available resources. If resources are unavailable a job-step is placed in a hold state until the resources are freed by other tasks. Job steps can be conditionally executed under JCL or program control.

Resource Allocation

Users may optimize the scheduling of their job-steps and resources by means of the LIMIT, SLIMIT and RESOURCE control commands. These commands allow scheduling of jobs prior to the availability of total job requirements.

Task Management

Execution or dispatching priorities of jobs are controlled by utilizing the hardware external interrupts of the Sigma computer, thus reducing overhead. Job classes are assigned at SYSGEN time to hardware interrupt levels. To change tasks or jobs requires triggering of the hardware interrupt. Multiple job classes can be assigned to one interrupt level and, if desired, a time-slicing option may be used to share the CPU resources with the tasks at that level.

Symbionts

XOS maintains symbionts, that asychronously buffer I/O operations on disk - for the card reader, card punch, line printer and remote batch terminal.

Accounting

XOS maintains, via job and job step management, statistics about system and user program performance for purposes of system performance evaluation and scheduling to improve significantly, an installation's throughput. Statistics gathered are:

Volume Accounting
Disk Accounting
Job and job step Accounting

MULTI-BATCH

The items which may exclusively be used by the batch user are:

- The procedures M:STIMER and M:LINK
- Private and account volumes
- Direct addressing of peripheral devices via an assign or reserve

Maximum User Program Size

Virtual = 128K
Real = 107K on a 128K system as specified in the Statistics Section under the Mini-Batch configuration

TIMESHARING SUBSYSTEM

The Timesharing Subsystem is an optional component of XOS which permits an installation to provide concurrent batch processing and Timesharing. XOS Timesharing allows the user to perform Timesharing Subsystem Commands, to execute conversational processors from his terminal such as the Text Editor and Debug, and submit jobs to the batch processing stream.

The interrupt level at which the Timesharing Subsystem executes and the Subsystem characteristics and communications network are defined at system generation time. The common system resources to be allocated to Timesharing - such as memory and disk space - are defined when the Timesharing subsystem is initiated. These resources may be dynamically modified, however, by the central operator at anytime.

The Timesharing Subsystem controls the management of all terminals using the TAM Access Method in character mode. Teletype-like and 2741-like terminals are supported.

The Timesharing Subsystem associates a task with each user and time-slices these tasks. When a task completes its time slice or is waiting for an event (such as terminal I/O completion), it may be swapped from memory to a predefined file on a secondary storage device. This permits a number of tasks to be managed at the same time. When a Timesharing task is swapped into memory, it is placed on the activity chain corresponding to a priority defined at system generation. Each Timesharing user task can take advantage of all batch processing and file management facilities using the Timesharing Commands and the TSAM access method to the standard Timesharing processors - EDIT, Timesharing DEBUG, SLINK, BASIC and FLAG. The user may issue three types of commands to the Timesharing Subsystem from his terminal:

- Primary Commands
- Batch Commands
- Secondary Commands

PRIMARY COMMANDS

These commands direct the Timesharing Executive to perform certain functions directly. Primary Commands are briefly described below.

LOGIN	Initiate a Timesharing session for the user. The user must specify a valid account number and name, and a password, (if his name and account are password protected).
CANCEL	Is used to cancel a request to login to the Timesharing Subsystem and causes an automatic disconnect.
CLOCK	Prints the current time of day on the user's terminal

PRIMARY COMMANDS (Continued)

CHARGE	Outputs the user's accounting log for his session on his terminal.
COMMENT	Permits the user to insert comments on his terminal which are not analyzed by the system.
TAB	Causes tab positions to be established for terminal input
STOP	Permits the user to stop his currently executing program
RESTART	Restarts a program which the user interrupted by keying in an Attention 1
WAIT	Permits the user to stop his session temporarily without performing a disconnect
SAVE	Causes the files specified to be saved for the user.
LOGOUT	Closes a Timesharing session and causes the accounting log for the session to be output on the user's terminal and saved in the system accounting log.

BATCH COMMANDS

These commands permit the Timesharing user to create jobs for batch processing and execute these jobs in the batch processing stream. The Timesharing user can directly access any removable volume or private account volume using these commands. Batch processing commands are summarized below:

CATAL	Causes a group of control cards defining a secondary command to be cataloged. The Timesharing user specifies the name under which his control card set is to be cataloged. He may also delete or replace previously cataloged files, or he may request that his group of control cards simply be analyzed for errors. The secondary command created by CATAL is available to all users under the same account number.
EXECUTE	Permits the user to execute a job step of a cataloged command in batch processing. Upon completion of the job step, the listing log will be output to the user's terminal.

BATCH COMMANDS (Continued)

BATCH	Allows the user to input a complete job for batch processing from his terminal. The user may specify the job class (and priority if the T-class is used), job identification and where the job results are to be output. He may also specify that his output is to be placed in a permanent file. Each user may be executing many BATCH commands simultaneously - the number is defined at system generation.
STATUS	Obtains the status of a job submitted to batch processing from the user's terminal.

SECONDARY COMMANDS

These commands are a group of commands written in the XOS batch control language and cataloged either from the user's terminal or in a batch stream. They are, therefore, definable and extendible by the user. Secondary commands may be cataloged under a user's account, in which case only authorized users of that account may access the command. Or, they may be cataloged under account:SYS in which case they may be used, but not modified by all users. The Timesharing processors are created in this manner.

The name of a secondary command is the name under which the group of commands has been cataloged. A secondary command is initiated by the Timesharing user by giving its name and if necessary, a list which specifies parameter values. When a secondary command is issued, the Timesharing Subsystem first searches for such a partition in the file associated with the user's account. If it finds it, it initiates the execution of the corresponding job; otherwise, the system file is searched.

Any standard batch processor may be run directly from the user's terminal by cataloging the processor as a secondary command, with the input and output operational labels assigned to IN and OUT respectively. However, the procedures M:LINK and M:STIMER are ignored.

The secondary commands permit the user to create his own commands and execute them immediately. He may also use secondary command sets to create new commands at will.

PROCESSORS

The interactive Timesharing processors include the Text Editor, Debug, the Link Editor (SLINK), BASIC and FLAG. Two of these are described below:

The standard batch processors can also be executed in the Timesharing partition.

TEXT EDITOR

The Text Editor is one of the interactive XOS Timesharing processors. It is activated from the user's terminal via the command EDIT. The Text Editor allows the user to create and modify disk resident source files for use by other processors or programs. The user has the ability to:

- Create a sequenced source file
- Copy a specified file or part of a file
- Create a new sequenced file from an old file or part of an old file
- Locate a file and list its characteristics
- Delete a file
- Insert or delete a record or sequence of records in an existing file
- Replace a record or sequence of records in an existing file with a new set of records
- Perform intra-record character string substitution and manipulation
- List a file without line numbers or list only the file line sequence numbers
- List a sequenced file

The Text Editor uses the TSAM, ASAM and AIAM access methods. The files it creates and manipulates are:

- Fixed consecutive, 80 byte record files
- Variable consecutive file with records less than 141 bytes
- Indexed sequential files with records less than 145 bytes (bytes 1-3 are used for the key)

DEBUG

The Timesharing Debug processor is one of the XOS interactive Timesharing processors which may be used in conjunction with Meta-Symbol created programs and the Timesharing Link Editor. It is designed to aid the user in program check-out. The user has the ability to:

- List and modify the contents of memory locations within his program
- Insert instructions or data in his program
- Reinitialize, restart or halt program execution
- Insert and suppress program checkpoints
- Execute his program in single-step mode controlled from his terminal

TSAM

The Timesharing Access Method - TSAM- is the access method used by programs operating in Timesharing mode to perform terminal input/output. TSAM is designed to permit compatibility of programs between Timesharing and Batch. Its operation is identical to ASAM, however, the logical labels must be IN and OUT which designate respectively the terminal keyboard and printer.

On input, the user's program receives the text of the message with corrections effected and without the end-of-message character. If fixed format is declared, each record is completed with blanks.

On output, TSAM performs any tabulations or formatting specified in the data control block. It takes into consideration the physical number of characters per line in order to seperate, if necessary, the message into several lines. TSAM terminates output by positioning the carriage at the beginning of the next line.

TSAM outputs the prefix or prompt character specified by the user without intervention of the user task.

The I/O procedures M:OPEN, M:CLOSE, M:SETDCB, M:MOVEDCB, M:GET, M:PUT and M:DEVICE are available to the user and operate as in ASAM with the additional features cited below:

M:DCB	Permits the user to specify a prompt character which will be output by TSAM each time the user task is ready to accept input (PFX). Also, the user may specify that he wishes to perform his own output formatting (ULC).
M:DEVICE	Allows the user to suppress character echo on input

REAL-TIME

The system makes available foreground User tasks (FUT) for user implementation of real time routines. In BOO, FUT's will operate at a primary interrupt level, equal to that of 'P'-class jobs. In addition, FUT's may ATTACH themselves to any number of Real-Time interrupt levels; these Real-Time routines (ATTACH'ed to external interrupt levels of higher priority than the XOS Task Management levels) will be given control of the CPU (in master mode) upon the occurrence of their respective interrupt level (after the monitor insures that the map is loaded to reflect the virtual image of the ATTACHing FUT and after exchanging the accounting clock so as to charge BST for the upcoming CPU time).

Please note the above terminology: FUT's are operator - initiated tasks which operate at the Task Management interrupt level of 'P'-class jobs; Real-Time routines are "sub-routines" within a FUT load module which are ATTACH'ed to Real-Time (external) interrupt levels and which operate in master mode as "pseudo BST's" (BASIC SYSTEM TASKS). A Real-Time routine may be activated by another FUT (ie., a different operator-initiated FUT) via the M:TRIGGER CAL1. This allows multiple FUT's to communicate with each other. Information may be passed between the FUT's via a common data area (mapped 1:1 in the LOWCORE module) defined at SYSGEN by SYSPROI (FRGD keyword of MONIT PROC). This data area will immediately follow the XPSD instructions generated to handle the Real-Time and Task Management interrupts. Since the number of interrupts used in a system will seldom change, the address of this common data area will not change from SYSGEN to SYSGEN.

Real-Time routines may not issue CAL1's. They communicate with their ATTACHing FUT's via the M:CLEAR PROC (this generates a Branch instruction rather than a CAL1). The ATTACHing FUT may synchronize its operation with that of its Real-Time routines by issuing M:WAIT CAL1's referencing the appropriate ECB (an ECB is defined for each M:ATTACH issued). Multiple FUT's will operate at the same primary interrupt level (ie., that defined at SYSGEN for 'P'-class jobs). Unless 'timeslicing' is specified for this interrupt level (at SYSGEN), FUT's will share the CPU with other FUT's and 'P'-class jobs on a "round-robin" demand basis. If 'timeslicing' is specified, FUT's will be timesliced ("round-robin-with-interrupt") among other FUT's and 'P'-class jobs.

In addition to the above mentioned ATTACHing capabilities, FUT's may do the following:

1. Change from Slave to Master mode and vice-versa; CAL1's may only be issued when in Slave mode.

2. Obtain control in the case of an abort condition (expansion of M:TRAP CAL1).

3. Suspend itself for a period of (real elapsed) time; M:CLOCK used in conjunction with M:WAIT.

4. ATTACH a Real-Time routine to a real-time clock interrupt (COUNTER 1=ZERO)

5. Execute the LRA instruction (when in Master mode).

6. Cause the I/O Supervisor to execute (via M:EXCP) a channel program built by the FUT; this allows support of non-standard peripherals.

7. ATTACH a Real-Time routine to the occurrence of the channel-end interrupt associated with the channel program executed via M:EXCP (see 6 above).

ALSO: If R-T tasks do not need to be core-resident, they may be coded as a portion of the NRM and its core allocation loading capabilities will be available to load these routines when required.

Privileged Procedures

M:MASTER — Changes the status of the executing foreground program to master mode. This procedure must be executed prior to using any of the Sigma 6/7/9 privileged instructions. No CALl instructions, that is, M:procedure reference, may be attempted while in master mode except for the M:SLAVE and M:CLEAR procedures.

M:SLAVE — Returns the status of the executing foreground program to slave mode. In slave mode, any CALl procedures may be used.

M:ATTACH — Associates a specific interrupt with the user's interrupt processing routine. Execution of the procedure csuses the interrupt location to be initialized and the interrupt to be armed and enabled. M:ATTACH may be used to "attach" three types of interrupts to the user's routine:

- External, in which the user specifies the interrupt group and level.
- Counter 1 zero, in which the user may specify regular interruptions to be processed by his routine.
- I/O interrupt associated with M:EXCP processing, in which the user may specify his own processing routine for interrupts from an I/O device accessed by the M:EXCP procedure.

M:DETACH — Causes the specified interrupt or interrupts to be disarmed and disabled and the memory location for the interrupt (s) to be reset to zero.

M:CLOCK — Permits execution of a user routine at regular intervals controlled by the counter 1 and counter 1 zero interrupts. The user specifies a clocking interval relative to the counter 1 zero interrupt frequency. Each time this interval elapses, an event is posted in the event control block (ECB) and the user's routine is entered. A count of the total number of elapsed intervals is kept in the ECB and may be accessed by

4. ATTACH a Real-Time routine to a real-time clock interrupt (COUNTER 1=ZERO)

5. Execute the LRA instruction (when in Master mode).

6. Cause the I/O Supervisor to execute (via M:EXCP) a channel program built by the FUT; this allows support of non-standard peripherals.

7. ATTACH a Real-Time routine to the occurrence of the channel-end interrupt associated with the channel program executed via M:EXCP (see 6 above).

ALSO: If R-T tasks do not need to be core-resident, they may be coded as a portion of the NRM and its core allocation loading capabilities will be available to load these routines when required.

Privileged Procedures

M:MASTER Changes the status of the executing foreground program to master mode. This procedure must be executed prior to using any of the Sigma 6/7/9 privileged instructions. No CALl instructions, that is, M:procedure reference, may be attempted while in master mode except for the M:SLAVE and M:CLEAR procedures.

M:SLAVE Returns the status of the executing foreground program to slave mode. In slave mode, any CALl procedures may be used.

M:ATTACH Associates a specific interrupt with the user's interrupt processing routine. Execution of the procedure csuses the interrupt location to be initialized and the interrupt to be armed and enabled. M:ATTACH may be used to "attach" three types of interrupts to the user's routine:

- External, in which the user specifies the interrupt group and level.
- Counter 1 zero, in which the user may specify regular interruptions to be processed by his routine.
- I/O interrupt associated with M:EXCP processing, in which the user may specify his own processing routine for interrupts from an I/O device accessed by the M:EXCP procedure.

M:DETACH Causes the specified interrupt or interrupts to be disarmed and disabled and the memory location for the interrupt (s) to be reset to zero.

M:CLOCK Permits execution of a user routine at regular intervals controlled by the counter 1 and counter 1 zero interrupts. The user specifies a clocking interval relative to the counter 1 zero interrupt frequency. Each time this interval elapses, an event is posted in the event control block (ECB) and the user's routine is entered. A count of the total number of elapsed intervals is kept in the ECB and may be accessed by

	the user. Multiple M:CLOCK procedures may be issued by the user each specifying a different frequency. Multiple procedures are processed on a first-in, first-out bases.
M:RCLOCK	Stops the interruption of the user program at regular intervals by cancelling the corresponding M:CLOCK request. Up to 5 M:CLOCK procedures may be cancelled with one M:RCLOCK request.
M:CLEAR	Retums to the monitor where the interrupt processed by the user routine is cleared, rearmed and enabled, M:CLEAR is the last procedure executed in the user interrupt processing routine.
M:TRIGGER	Causes an external interrupt specified in an M:ATTACH procedure to be triggered. This procedure also allows communication between two foreground tasks with the aid of the M:ATTACH procedure.
M:INITECB	Permits the initialization of an event control block (ECB). This procedure is executed each time an ECB is used for posting an event. A user task may wait for many events to be posted to the ECB prior to executing an M:INITECB by using the M:WAIT procedure
M:EXCP	Permits the user's foreground program to access I/O devices directly. The user builds an Input/Output Block (IOB) for the device he is using. The M:EXCP procedure requests the Input/Output Supervisor to execute the channel program associated with the user built IOB and optionally, return control to the user. The user may specify that he wishes to process all interrupts from the I/O device.
M:TRAP	Includes an option to aid the foreground user in abort control -ABRT. This option will cause control to be returned to the user for any abort conditions except the following: The user's job exceeded its execution time limit or output pages limit as specified on the !LIMIT command. The operator aborted the job. The job aborted during abort processing. The job executed an M:ERR.
M:OPENAL	Permits the user executing in foreground mode to access the accounting log. This file is consecutive. 80 characters per record, created via ASAM. The reading of the accounting log is destructive. After the execution of an M:OPENAL, the accounting log is reinitialized.
M:LOCK	Permits a resource or a number of resource units to be locked for exclusive use by a user task running in foreground mode.
M:FREE	Frees the resource(s) locked via the M:LOCK procedure.
M:RCB	Generates a resource control block(RCB). This procedure is non-executable.

TELEPROCESSING

Telecommunications Access Method (TAM)

TAM provides the programmer with a collection of user-level services provided by the XOS Communications Management System (CMS) for input/output operations over transmission lines. Some of the automatic functions provided by TAM are:

- Device Controller handling
- I/O and external interrupt processing
- Error detection and retry processing
- Queueing of I/O requests
- Line time-out processing
- Line/terminal/component polling and selection
- Automatic data translation, e. g., ANSCII to EBCDIC. Sysgen definable translation tables of character sets
- Switched and leased lines in simplex, half duplex or full duplex mode
- Blocking of groups of characters received by 7611 during a defined time period

Communications Networks

CMS supports one or more bi-point and multi-point networks definable at system generation and modifiable by program control at run-time.

Polling/Selection Sequences

TAM provides the user with the capability of polling automatically (for input) or selecting (for output) a station and/or component of a station.

Data Access

TAM I/O operations are performed on monitor transmission blocks in buffers managed by the user via the Virtual Sequential Access Method.

Groups of Lines

A group of lines is a set of transmission lines with identical characteristics linked to a given application. These lines are managed as a group to reduce overhead. Multiple groups and multiple applications can be supported simultaneously.

Transmission Modes

- Message Mode — For use with buffered terminals such as the 7670 Remote Batch Terminal
- Character Mode — For use with terminals lacking a hardware buffer such as the teletype

Remote Batch Processing and Telesymbionts

Remote Batch Processing is handled via the telesymbionts. The telesymbionts are system routines that read programs, data and control messages from and send programs to remote terminals a such as the 7670 RBT.

TAM System Procedures

- M:DCB — Enables user at assembly time to introduce any or all of the DCB parameters applicable to TAM

- M:MOVEDCB — Allows dynamic creation of a DCB in the common area by replication of an existing DCB.

- M:SETDCB — Allows modification of DCB parameters during program execution

- M:OPEN — Establishes the connection between the program DCB and the network by

  - verification of the explicit user defined lists and the lines assigned as resources
  - initialization of the network; initialization of the transmission device controllers and the line adapters (character mode)
  - verification of the operational status of the intermediate telecommunications equipment
  - creation in the user program of the list of components or terminals if an implicit list is required
  - creation of the required communications tables between the access method and the I/O supervisor

- M:CLOSE — Closes the DCB and, optionally, the network. The close may be either temporary or definite.

- M:LIST — Requests at assembly time an explicit component or terminal polling or selection list. Lists may be linear or circular.

- M:MDFLST — Requests at execution time modification of a component or terminal list.

- M:WRITE, M:READ — Requests a transmission of data to or from a terminal, respectively. A user may also read in survey mode to detect any attention characters a terminal may have sent.

- M:CHECK — Requests a test for successful completion of a specific I/O operation

- M:DEVICE — Enables the users to specify a transmission code change or to perform a device specific operation such as:

BEL	send an alarm to a component
SUS	suspend transmission from a component
ABO	abort transmission from a component
IND	identify by index into a list the component on which the operation is to be performed
MOD	redefine working mode to EBCDIC or binary

USER CONTROL

XOS Control Command

The XOS monitor receives job descriptions from control commands. They are a means of communication between user and system; they describe the sequence of the different steps comprising a job. Data may be placed after each step for use during execution of that step.

The collection of control commands and data forms a job, which is entered into the system by the input symbiont. Job initiation does not necessarily occur in order of presentation to the input symbiont. The system schedules jobs and job steps by job class, by user-assigned priorities (T class), and by required resources, as described by the control commands.

XOS allows the cataloging of control commands into groups called command sets.

A set to be cataloged is syntactically analyzed and is then filed on the system disk in a specialized file.

The set can be retrieved for execution in two different ways:

1. Insertion into a set of commands. This execution is made with the aid of the !EXEC command. The command set to be executed can represent one or several job steps, or part of a job step.
2. Initiation of a parallel job from the operator control device. The cataloged command set must contain all the commands necessary to execute the same job in a production class.

Batch Commands

!JOB Signifies the start of a job and defines job class, account and user-id. It may be used to catalog a set of control commands.

!RUN Executes a program as a job step. Allows explicit calls to load modules as files or partitions of a file. Allows (via job switch word control) selective step execution. Allows passage of parameters to the called program.

!LIMIT Specifies maximum system resources that may be used by a particular job.

!SLIMIT Specifies maximum core or temporary disk space that may be used by a job step

!RESOURCE Specifies the minimum requirements for shared peripherals required to initiate a job

!EXEC Executes previously cataloged command sets with facilities for passing parameters and conditional execution

!ASSIGN Defines a physical medium and associates it with a program defined DCB (Data Control Block). Basic assignment types are:

- File -labeled permanent files or temporary files on magnetic devices
- Device -files on non-magnetic devices - unlabeled files on magnetic tape volumes
- Indirect-reference to another ASSIGN command and its associated characteristics
- Dummy -Simulation of an input or output file

Assigns may be FRE - released after job step termination - or MTN (maintained - remain in effect over job step until freed or job termination.

Assign command optional parameters are:

- STS — Defines status of file -OLD, NEW or MOD
- LNK — Indicates concatenation of multiple files
- UNT — Defines the volume (tape or disk), account or private account volume on which a file resides. Sub-option OP points to volume defined on a previous assign.
- SQN — Specifies index into series of volumes.
- PAR/MNT — Allows for PARallel or MNT (serial) mounting of volumes.
- DEF — Allows for deferred mounting of volumes.
- DSP — Specifies volume disposition after file DCB closing. RET=remain mounted thru next job step. KEP=remain mounted thru next job. Default is dismount.
- SIZ — Specifies file size allocation characteristics for new files (disk). SEParation allows writing of overflow and index blocks (indexed file) on a separate volume.
- NAM — Specifies a 1 to 17 character file name. Sub-options allow for specification of version, absolute and relative generation numbers.
- CTG — Indicates the volume-id on which a file resides is to be cataloged.
- PRT — specifies file access protection for readers and writers by account and password.
- RET — Specifies file retention period.
- OUT — Specifies output is to printer symbiont associated with the listing logs.
- SLP — Specifies printed output is to be directed to a separate "file".
- NKP — Specifies no catch up mode for line printer symbiont.
- SCP — Specifies card punch symbiont.
- STA — Specifies a remote terminal station for printed output.
- DEV, nn — Specifies a class of device (MT, CR, CP, etc.)
- DEV, ADR, nn — Specifies a logical device address.
- DCB — Introduces a series of optional parameters that optionally define the file characteristics:

  - ORG — File organization (sequential, indexed, partitioned, direct)
  - NBF — Number of buffers
  - MOD — Binary, BCD, EBCDIC, packed, unpacked
  - BHR — Block header length
  - BLK — Block length
  - MXL — Maximum allowable I/O transfer length
  - NBC — No block count
  - DLC — Record delete character
  - FRM — Record form (fixed, variable, undefined)
  - KYL — Key length
  - KYP — Key position
  - REL — Record Length
  - CNT — Page count on each page at specify print position
  - DTA — Column/print position in which data is to begin
  - LIN — Number of lines per printed page
  - SEQ — Sequence number in columns 73-80
  - SPC — Number of spaces between printed lines
  - TAB — Tab character settings
  - VFC/NVF — Vertical format control for printed output

!SWITCH	Allows resetting and setting of each of the 32 bits in a word (job switch word) associated with each job. Switch bits may be tested (and set) by programs or by RUN and EXEC commands for conditional execution Bits 0 and 1 allow for forced step execution and program abort memory dump respectively.
!TITLE	Specifies the printing of a page heading at the beginning of each logical page of printed output.
!MESSAGE	Allows the sending of a message to the operator control device with a WAIT option to temporarily suspend the job until operator action.
!COMMENT	Permits insertion of any kind of commentary in the command deck.
!Processor Call	Allows the invocation of programs cataloged under the system account(:SYS). Parameters may be passed to the called processor.
!DATA	Indicates that job control commands, binary cards or Hollerith (026) cards are included in the following set of data and will be read until an EOD command is encountered.
!EOD	Terminates a set of data passing an "end-of-file" indication to the active program.

Telesymbiont Commands

The remote operator can send commands that request the status of a remote batch job or determine the destiny of job output files. He can also receive operational messages from the telesymbiont and from the operator at the central station. Commands are introduced on cards in the job deck input stream.

/MESSAGE	Allows the remote operator to send a message to the central station operator.
/STATUS	Requests the status of the specified job. The status is output on the remote station printer betweenjobs. The status indicates whether or not the job has been received, is queued or is active.
/HOLD	Designates that all output files for the job are to be held at the central station until requested.
/RELEASE	Directs the system to output any files for the specified job.
/SWITCH	Redirects the output for the station to a remote station specified or to the central site printer.
/RESTORE	Annuls the effects of a /SWITCH command.
/END	Indicates that the sending station will terminate its connection when the output for the specified job has been transmitted to the remote station.
/SUSPEND	Permit the remote operator to temporarily halt.
/CONTINUE	Transmission of a file and continue the transmission of the same file later.

Memory Managemen.

XOS utilizes the Sigma memory map option; user jobs are executed in virtual memory. Most users need not be concerned with virtual memory and mapping since the mapping function is performed by XOS in a way that is transparent to the user. Only the master-mode user need be concerned with the distinction between virtual vs. physical memory.

Space Allocation Procedures

M:GL - Get limits of dynamic space by returning the number of contiguous unallocated whole pages between the highest address of local dynamic and the lowest address of common dynamic.

M:GP - Allocate specified number of pages in local dynamic.

M:FP - Deallocate specified number of pages in local dynamic.

M:GSP - Allocate block of specified number of words in common dynamic.

M:FSP - Deallocate previously requested block of words in common dynamic.

Dynamic Overlay and Program Loading

M:SEGLD - Load a specified program overlay segment into memory as well as all those segments not already loaded which lie on the path of the tree between the calling segments and the specified segments.

M:LDTRC - Dynamically request the loading into memory of, and transfer of control to another program without preserving the calling program. Memory space and local dynamic of the calling program is freed; common dynamic remains unchanged.

M:LINK - Dynamically load and transfer control to another program while preserving the calling program and its local dynamic area on a temporary disk file for a later return.

Program Management

M:TRAP - Enables the executing program to be allowed to handle certain CPU - detected abnormal conditions

PS	Stack overflow
UI	Unimplemented instruction
NI	Nonexistent instruction
NMA	Nonexistent memory address
PSM	Privileged instruction in slave mode
MPV	Memory protection violation
FP	Floating-point fault
DEC	Decimal arithmetic fault
FX	Fixed-point arithmetic fault
CL2	CAL2 instruction
CL3	CAL3 instruction
CL4	CAL4 instruction
NAO	Non-allowed operation
ALL	All of the above

With other options certain combinations of faults may be ignored.

M:RETURN - Allows return from a user routine which may be his main program or from his "abnormal" routine, trap, timer, or operator interrupt routine.

M:WAIT - Permits the user to place his program in a wait state until one or more of up to 255 events occur.

M:WAITL - Permits the user to wait on completion of a specified number of events, of a total number of events outstanding.

M:ERR - Allows user to request abnormal job-step termination; i.e., execution of M:ERR causes the program to be aborted with a specified code printed on the job control file.

Inter-Job and Job-Step Communication

M:SSS - Allows program to set (to 1) one or more of bits 2 through 31 of the JSW (Job Switch Word).

M:RSS - Allows program to reset (to 0) one or more of bits 2 through 31 of the JSW.

M:TSS - Allows program to test the status of one or more bits of the JSW for set or reset conditions.

External Communication

M:KEYIN - Display a message on the operator console and wait for the operator to reply to the message. The reply is transferred to a specified area in the program.

M:TYPE - Display a message on the operator console without a solicited reply.

M:PRINT - Write a record on the system listing log (job control file) normally output to the line printer symbiont.

M:INT - Allow the program to receive an interrupt from the operator via the console interrupt for program communication with the operator.

Time and Date Facilities

M:TIME - Obtain the date and time of day to within one hundredth of a second.

M:GETDAY - Obtain the date (Julian form) during program operation.

M:STIMER - Initialize a job-unique clock counter for a specified interval, activate only while the requesting program is running, and branch to a user- specified routine when the interval has elapsed. The program may specify time units in minutes, seconds or elementary (one pulse on hardware clock 3; normal for XOS in 500 Hz).

M:TIMER - Obtain the time remaining before a clock counter, previously initialized by M:STIMER, reaches zero. Optionally M:STIMER may be cancelled. Time may be returned in units of minutes, seconds or elementary intervals.

Batch Job Submittal

M:BATCH - Converts a user specified file to a symbiont file and submits it to the batch processing stream.

M:STATUS - Obtains the current status of a job which has been submitted to the batch stream.

Debug Aids

XOS furnishes the user with a collection of debugging aids grouped into the system service called Debug which consists of the !DEBUG processor and the Debug procedures. The procedures are also used by the processor. Their functions are described as follows:

- DCB which allows the user to specify a user DCB to be used for the output of requested debug information.
- Postmortem dump which allows the user to specify that portions of his program are to be dumped (in hex) conditionally or unconditionally at the end of its execution.
- SNAP which causes the printing of one or more memory areas before the execution of an instruction at an indicated address.
- SNAPC which allows conditional snaps of memory.
- IF which allows testing of a condition and setting or resetting of an associated flag.
- AND which requests a test of a condition if a flag is set - If true, the flag remains set; otherwise, the flag is reset.
- OR which requests a test of a condition if the corresponding flag is reset. If true, the flag is set; otherwise, the flag remains reset.
- COUNT allows for setting or resetting of a flag depending on the number of times the specified procedure has been executed.
- MODIFY specifies the replacement of one or more consecutive memory words (command form only).
- INSERT specifies the logical insertion of one or more consecutive memory words.

OPERATOR CONTROL

System Initialization

The system disk is loaded from a system save tape at device speed.

The system disk is booted in one of two modes: (1) cold start where all preexistent jobs and symbiont files are deleted, and (2) warm start where symbionts may be reactivated and the job queue re-initiated.

The operator may add, change, or delete user accounts with the !ACCT control command.

Corrections to the system may be added via the system debug processor. This need only be done once since all corrections can become permanent via system SAVE of the system disk(s).

Operator-System-Operator Exchanges

The operator may

- Simulate the AVR signal from a peripheral and optionally re-label an existing labeled volume.
- Label a volume.
- Abort a user job, symbiont or telesymbiont
- Transmit an interrupt to a user job or a symbiont typically to allow a program or symbiont request or receipt of an operator keyin.
- Recall a previously deferred message (deferred by slash command)
- Cancel console interrupt
- Initiate a symbiont or telesymbiont
- Initiate a parallel (previously cataloged command set) job and optionally pass parameters to the job.
- Dismount a currently mounted (AVRed) tape or disk pack.
- Lock or unlock a peripheral or exchange logical peripherals.
- Display the state of indicated system resources or operational components such as:

  job scheduler active queue
  job scheduler wait queue
  all system resources available
  disk pack status
  magnetic tape status
  all peripherals status
  work load waiting to be processed by output symbionts.

Symbiont Control

An operator may

- Suspend a symbiont
- Continue a suspended symbiont at the point of interruption
- Restart a suspended line printer symbiont at the start of the last page being printed
- Delete a current symbiont file
- Terminate a symbiont after the current file
- Abort a symbiont

Telesymbiont Control

A central site operator may

- Lock a telesymbiont after the current transmission
- Reroute output to a new remote station
- Send a message to a remote station
- Abort a telesymbiont
- Delete a remote station
- Restore a deleted remote station
- Display the operational stations
- End a session

A remote site operator may

- Send a message to the central site operator console
- Request a job abort
- End a connection

Peripheral Management

An operator may

- Reserve a specific peripheral per the request of an active job
- Display peripheral status
- Mount or dismount removable volumes
- Pre-label previously labeled or unlabeled volumes
- Disconnect peripherals from system resources
- Control runaway tapes via REQUEST keyin and an abort of the job using the tape drive

Crashes and Recovery

An operator may

- Direct a SYSER (system error) dump to the line printer or a magnetic tape in the event of a system crash.
- Later print the SYSER dump (on tape) in an interpretive format by using the ANALYZE program.
- Recover the system by a simple disk boot in either cold or warm restart mode.
- Check the status of all jobs queued, jobs lost (currently operating before crash), accounts, files, and symbiont output files open at the time of the crash.

Time-Sharing Operations

The XOS Time-sharing subsystem is initiated, controlled and terminated by central operator keyins. Specifically, the operator can initiate and terminate Time-sharing operations, modify Time-sharing parameters and resources, dynamically, display the operational status, current resources and transmission network of the Time-sharing Subsystem.

The operator controls time-sharing operations by means of the OPERATE and INTERRUPT keyins.

The OPERATE keyin is used to initiate the Time-sharing Subsystem and define initial resources: Transmission lines, maximum number of concurrent users, memory size of the time-sharing partition, number of system disk quanta to be reserved for temporary files, number of pseudo volumes to be allocated for time-sharing usage.

The INTERRUPT keyins are summarized below.

NR	Permits the operator to modify resources dynamically - memory, temporary disk and pseudo volumes.
DR	Displays the resources dedicated to the time-sharing subsystem and the number of current users.
DL	Displays the transmission network.
UL	Permits the operator to unlock a transmission line which has been locked via a UNIT keyin.
HH	Terminates the time-sharing subsystem.

INSTALLATION CONTROL

XOS System Generation

An XOS SYSGEN is the process by which a system conforming to an installation's hardware configuration and scheduling needs is created. The SYSGEN process is performed under control of an XOS system- either on dn installation's existing XOS system or, in the case of a new installation, on the minimum system provided on the Master Release Tape.

The XOS SYSGEN is performed by standard processors using standard file management techniques. A standard XOS SYSGEN required about one hour. Subsequent system changes can be accomplished in 15 to 30 minutes. An existing XOS installation can perform an entire XOS SYSGEN as a set of batch jobs during normal system operation. The output of an XOS SYSGEN is a Start-up Tape containing the monitor, processors and libraries for the target system. The start-up tape is booted into the target system, accounts and user files added and a system save tape (DISK DUMP) created of the new XOS system.

The series of batch jobs used to create sn XOS system may be divided into three phases. The first phase - SYSPRO - consists of five Meta-symbol assemblies which through procedures create load modules describing the system configuration, scheduling needs and monitor structure. The second phase - SYSEDIT - links the load modules created in phase one with the monitor load modules on the Master Release Tape and creates a bootable image of the target system within a standard XOS disk file. Phase three - SYSREL - adds the XOS processors and libraries to the image file on disk rebiasing them for the target system. This image file is then copied to tape using the FMGE processor. This tape is the start-up tape for the target system.

A few of the procedures which the user may modify during phase one - SYSPRO - are briefly described below. However, default values exist for all procedures and they need not be specified. Some of the procedures

- Permit the user to define the external interrupt structure to be used by Task Management for scheduling and any interrupts to be used for real time, telecommunications, etc.
- Define the batch job classes, the resources available to these classes and limitations on jobs executing in these classes - including memory, temporary disk space, execution time, maximum card and page output.
- Define the symbionts for the system, their names, permitted memory space and their device residency.
- Define the characteristics of the Timesharing Subsystem such as scheduling level, time-slice per user task, number of batch jobs permitted for each user, system resources.
- Defines the peripherals and telecommunications network.
- Define the translation tables for the telecommunication network and any special function codes required by the installation. An installation may also modify the standard translation tables.
- Describe remote batch stations and their components.
- Define the system resources.

- Define any standard operational labels and their default assignments.
- Define the monitor structure, which modules are to resident and non-resident.

Account Control

XOS maintains the supercatalog and allows the system manager to

- -Define new account numbers and specify characteristics of their account volumes
- -Modify the parameters of the account volumes already known to the system.
- -Remove existing account numbers from the system
- -Change an account's catalog from one account volume to another
- -Change an account's catalog from a pseudo-volume to a removable account volume or inversely.

Whenever XOS is quiescent, the operator may modify the supercatalog by using a !ACCT card deck. The system processes these cards immediately and outputs a summary of the current status of the supercatalog via the printer.

System Patching and Debugging

XOS provides a system debug facility to display system resources, modify the system and aid the systems programmer in locating system problems. Commands to the system debugger may be entered via the operator's console or card reader. The system debugger has the following facilities:

- Define a new symbol (not in the REF/DEF stack) as a constant.
- Clear the debugger's symbol stack
- Modify the contents of one or more core locations in a Monitor module and updates the disk image
- Insert one or more instructions in a monitor module and updates the disk image
- Restore an instruction that was modified by and insert or ENTER command
- Can modify any system disk block and may be used for patching processors
- Catalog all subsequent system debugger commands through END for deferred execution
- Allows a call to debugger to be inserted at the specified location
- Output the core location or locations specified either on the printer or console
- Causes a hexadecimal of the system disk blocks specified
- Causes a transfer of control to other debug commands
- Exits the debug facility.

FILE MANAGEMENT

The XOS File Management System (FMS) is comprised of a collection of system programs responsible for the movement of data between memory and external storage for user programs and system tasks. These programs provide the facilities to locate data, manage buffers and external storage, read data, and write data.

FMS provides a set of services to coordinate the transfer of information between user programs and data files:

- FMS handles all types of physical files consistent with the I/O devices on XOS systems. These include unit-record devices, magnetic tapes, disk packs, and RADs.
- For magnetic tapes and disk packs FMS handles all combinations of single or multiple volume files or multi-file volumes.
- FMS handles both standard and nonstandard labels on magnetic tape. The standard tape label is ANS compatible. For nonstandard labeled files (user labels), the entire volume is treated as data.
- In order to achieve flexibility, FMS supports a variety of file organizations and record formats. File organizations include:
  - Sequential
  - Indexed sequential
  - Direct
  - Partitioned

  Record formats include:
  - Fixed, Variable and Undefined lengths

  Fixed and variable formats on tape are ANS compatible.
- FMS provides file-sharing and file protecting functions. Shared files may be read by several tasks or processes concurrently. However, in order to write on a shared file, the user must obtain exclusive use of the file. A shared file may be protected by the file owner against unwarranted access. This protection is achieved by means of a password specified at the file's creation and by a list of users who are authorized to read or write the file.
- FMS permits file concatenation. This facility enables the user to logically connect several data files into a single consecutive file. FMS will automatically process from the end of one file to the start of the next file without any intervention from the user.

XOS provides facilities for six different methods of file processing, referred to as access methods. These access methods are divided into two groups according to the general techniques involved in their use.

The assisted access methods operate at the logical record level and are characterized by a high degree of system service and control: Record blocking/deblocking, error checking, volume switching, etc. They are:

- Assisted sequential access method (ASAM), intended for the creation and sequential processing of files on any type of media.

- Assisted indexed access method (AIAM), intended for the creation and direct-access processing of indexed files.
- Assisted partitioned access method (APAM), intended for the creation and processing of files that are segmented into partitions.

The basic access methods operate at the physical record (block) level and are characterized by a high degree of user control and relatively little system intervention. They are

- Virtual sequential access method (VSAM), intended for the creation and sequential processing of files, at the block level, on any type of media.
- Virtual direct access method (VDAM), intended for the creation and direct-access processing, at the block level, of files on direct-access storage media.
- Basic direct access method (BDAM), intended for access to a private or unlabeled direct-access volume by relative sector addressing.

Volume Classifications

Standard Volume - contains a volume header with an ANS standard volume - id and an account number. Organizations may be mono-volume file, multi-file multi-volume.

Non-Standard Volume - does not conform to XOS (ANS) standards for volume formats and may be processed in DEVice mode.

Common (Public) Volume - a tape or disk volume that doesn't belong to any user account. It may be used for temporary or permanent files. It becomes private after the creation of a permanent file.

Private Volume - a tape or disk volume that belongs to a given user account.

Account Volume - a direct access volume that contains the account catalog for a given account.

Pseudo-Volume - an account volume that resides in a dedicated portion of secondary system disk storage.

Cataloged Files

The file identification and identification of a volume on which a file resides may be cataloged for future reference by file name only.

Generation Data Groups

A set of cataloged files known by a single name, each member of which is distinguishable one from another by an absolute generation number.

Closed Loop - volumes in the defined generation group are rotated so that (by default) the oldest volume is used for output and the newest is used for input.

Open Loop - volumes in a generation group are (by default) new volumes for output and are the most recent volume for input. The oldest volume is "pushed" out of the loop when a new one is created.

Volumes may be referenced by default (file name only), by absolute generation number or by relative generation numbers.

Volume Sharability

When a volume is defined as sharable, one or more users can access one or more files under one or several accounts on a given volume.

File Sharability

When a volume is sharable, the files residing on that volume may be defined as sharable as follows:

1. More than one DCB can be open to the same file for concurrent input mode processing.
2. The system controls multiple access to a single file whenever more than one user wishes to modify the file by queueing requests for opens.
3. Account authorization and passwords apply.

Creation and Modification of DCBs

M:DCB – allows assembly time creation of a partially specified or complete DCB

M:MOVE DCB – allows dynamic creation of a DCB by execution time replication of an existing DCB

M:SETDCB – allows execution time modification or completion of a DCB prior to opening and modification of error and abnormal return addresses subsequent to opening

ASSIGN COMMAND – allows run time specification of certain DCB parameters which modify the DCB at open time

M:OPEN – effects both explicit DCB modification and as specified by the ASSIGN command and/or implicit modification by information contained in the label of a file opened for input

Execution Time DCB Assignment

M:ASSIGN – Allows during program execution to

(1) define a temporary file and assign an operational label to it
(2) define a permanent file on a physical resource

Processing of Files

M:OPEN – activates link between a DCB and a physical file. The modes are:

Input mode (forward reading)
Backward reading
Output mode (forward writing)
Update mode (reading and modification)

M:CLOSE - suspends activity of a DCB and the processed file. Types of close:

-temporary close. DCB link maintained. Subsequent open may be in different mode.

-definite close. Cancels DCB link, but maintains job file link. Resource not released.

-definite close. Cancels DCB and job-file link. Resource not released.

-definite close. Cancels DCB and job-file link. Resources released.

Some sub-options are:

catalog the file if CTG specified or if on an account volume

delete existing file or suppress cataloging of new file

M:GET - Get next record. Valid for ASAM, ISAM and APAM. Permits reading of logical records either to a program defined buffer (MOV mode) or to a monitor buffer (LOC mode) with a pointer supplied.

M:PUT - Put next record. Valid for ASAM, ISAM and APAM. Permits writing of next logical record in file being created or updated. MOV mode or LOC mode are allowed.

M:TRUNC - Permits termination of operations on a partially processed block and passage to the next sequential block for processing.

M:DELREC - Permits deletion of last logical record access by an M:GET.

M:CVOL - Permits explicit switching to the next sequential volume of a file.

M:NOTE - Obtains pointer to current block/record position for subsequent use by an M:POINT

M:POINT - Permits repositioning within a file to a record pointed to by information obtained from a previously issued M:NOTE.

M:DEVICE - Allows requests for device dependent operations. Options are:

- send message to operator for change of print forms
- request page ejection during printing of a file
- position a magnetic tape file by one block backspacing, one block forward spacing, position to first block, and position behind last block.

M:STOW - Permits storing or deleting of principal and synonym partition keys into the directory of a partitioned file.

M:FIND - Permits positioning to a partition boundary selected by either a principal or synonym key in a partitioned file.

M:READ - Permits reading of the next sequential physical record (VSAM) on a program determined by physical record (VDAM). BDAM is by relative disk sector number.

M:WRITE - Permits writing of the next sequential physical record (VSAM) on a program determined physical record (VDAM). BDAM is by relative disk sector number.

M:CHECK - Tests a given I/O operating for proper completion placing the issuing program in a wait state if necessary, to await such completion. Applies to VSAM, VDAM and BDAM.

Abnormal and Error Handling Routines

A program DCB may specify certain routines which are to handle events or errors in I/O processing such as:

- Programming errors
- Job initialization errors
- Abnormalities in file content
- Device related errors
- Transmission errors
- Bypass of errors
- Passwords
- End of file, or volume
- File expiration date
- User label processing
- File or key non-existence or existence
- Sequence errors

PERFORMANCE

Several comparative performance job streams have been run on XOS and other systems. Some of the results are summarized below. Details of the following may be obtained from P. H. Johnson.

F320 Business Job Stream

An internal business system (F320) was run on XOS-AOI, BPM-FOI and UTS-COO. The following represents the total elapsed times for each system:

BPM	UTS	XOS
120 minutes	124 minutes	68 minutes

Dow Chemical Benchmark

Times given for this benchmark are for XOS on a Sigma 6 vs. OS/MFT-II on a 360/40 for a series of commercial applications.

	XOS - Sigma 6	OS/MFT-II - 360/40
Elapsed	18 minutes 45 seconds	37 minutes 30 seconds
Execution	12 minutes 23 seconds	36 minutes

COBOL Compilations

A set of tests were performed with seven COBOL compiles. The jobs were executed under XOS, UTS and BPM. Times are expressed as elapsed throughout.

XOS	BPM	UTS
7 minutes	13 minutes	14 minutes

Wichita State University

A series of COBOL and FORTRAN programs were compiled, linked and executed under several hardware vendor configurations and operating systems. The results expressed in total job elapsed times are given in seconds below for a stream of 7 jobs.

XDS Sigma 7	XOS	299.4
CDC 3100	MSOS	952.94
CDC 3200	MSOS	1341.27
GE 415	DPS	1896.00
GE 415	DAPS	1230.00
IBM 360/30	DOS	1234.20
IBM 360/40	DOS	1141.20
IBM 360/40	OS/HASP	1077.60
IBM 360/44	DOS	1051.20
IBM 360/44	OS/HASP	1093.20
IBM 360/50	OS/HASP	703.80

IBM 360/50	OS/HASP MFT IIRelease 16	703.80
IBM 370/145	DOS POWER	603.00
IBM 370/145	DOS Release 25	904.00

Assembler Performance Test Timings

BPM Meta-Symbol Sigma 7	10.834	minutes
UTS Meta-Symbol Sigma 7	8.9705	minutes
XOS Meta - Symbol Sigma 7	8.01	minutes

Lummus Benchmark

JOB	SYSTEM	WALL CLOCK TIME	BILLABLE TIME
Commercial Mix	OS-360/65-ASP	57.00 minutes	47.56 minutes
Commercial Mix	XOS Sigma 6	56.10 minutes	44.20 minutes
Multi-Feed Fractioner	Univac 1108		1.08 minutes
Multi-Feed Fractioner	XOS Sigma 6		4.85 minutes
Petroleum Blends and Cuts	CDC-6600		.89 minutes
Petroleum Blends and Cuts	XOS Sigma 6		4.88 minutes
Three Dimensional Space	Univac 1108		3.08 minutes
Three Dimensional Space	XOS Sigma 6		6.57 minutes

Bureau of Customs

Compute bound job.

SYSTEM	COMPILE-LOAD	EXECUTE
Sigma 6 (DBM)	2.142	19.129
Sigma 6 (UTS-B00)	.768	20.099
Sigma 6 (XOS)	2.030	19.290
Sigma 9A(UTS-A03)	.559	11.993
Sigma 9B(UTS-B00)	.768	12.626
Sigma 9A(XOS)	1.800	11.600
360/50	10.000	11.000

Xerox Region Conversions

Compile/Load - (15 programs)

DBM	59.92 minutes
UTS	74.80 minutes
XOS	38.00 minutes

Execution - (2 sorts, 6 programs)

DBM	11.30 minutes
UTS	10.25 minutes
XOS	7.30 minutes

STATISTICS

XOS 800 Core Size Requirements[1]

I.	Minimum Batch	39.5K
II.	Minimum Batch (a) plus Remote Batch (b)	
	Minimum Batch	39.5K
	Remote Batch	5.5K
		45.0K
III.	Minimum Batch (a) plus Real-time (c)	
	Minimum Batch	39.5K
	Real Time	4.0K
		43.5K
IV.	Minimum Batch (a) plus Terminal Batch Entry (d)	
	Minimum Batch	39.5K
	Terminal Batch Entry	25.5K
		65.0K
V.	Minimum Batch (a) plus Minimum Time-sharing (e)	
	Minimum Batch	39.5K
	Minimum Time-sharing *	28.5K
		68.0K
VI.	Minimum Batch (a) plus Full Time-sharing (f)	
	Minimum Batch	39.5K
	Full Time-sharing	28.5K
		68.0K
VII.	Minimum Batch (a) plus Remote Batch (b) plus Time-sharing (f)	
	Minimum Batch	39.5K
	Time-sharing	28.5K
	Remote Batch	5.5K
		73.5K

* XOS Timesharing Only System

Basic Monitor with 1 symbiont	17.5 K
Minimum Timesharing	28.5 K
	46.0 K

[1] Response to specific configuration sizing request from Marketing (see special features section for minimum 32K system.

600 Core Size Requirements (cont.)

a. Minimum Batch

Basic Monitor (with IMT controller and 4 drives; 1 DM controller and 2 drives; DEBUG trace stack)	12.0K
NRM Area	1.5
SST Work Space	1.5
DEBUG Patch Area	1.0
Card Reader Symbiont (when active)	1.5
Card Punch Symbiont (when active)	1.5
Line Printer Symbiont (when active)	1.5
Total Basic Monitor Size	20.5K
User Size (Minimum Batch)	
User Context & Monitor Service Work Area	2.0K
FLAG, COBOL, METASYM	17.0
	19.0K
Total Minimum Batch	39.5K

b. Remote Batch (7670)

TAM - Message Mode		
Resident Module	1.8K	
I/O Tables (2 7670's)	.7	
		2.5K
2 Telesymbionts (when active)		3.0
		5.5K

c. Real Time

Real time User Context & Monitor Work Space	2.0K
Real time User Program	2.0
	4.0K

d. Terminal Batch Entry

TAM - Character Mode		
Resident Modules	2.6 K	3.5 K
I/O Tables (8 lines)	.9	
Timesharing Task		
Resident Modules	1.5 K	
DRAGON Task	2.0	5.0
T/S Exec.	1.5	
T/S User		
User Context & Monitor Service Work Space	2.0 K	17.0
User Program (EDIT)	15.0	
Total Terminal Batch Entry		25.5 K

XOS 800 Core Size Requirements (cont.)

e. Minimum Timesharing

TAM - Character Mode (8 lines)		3.5 K
Timesharing Task		5.0
T/S User		
User Context & Monitor Service		
Work Space	2.0 K	
FLAG, BASIC, EDIT	18.0	
		20.0
Total Minimum Timesharing		28.5 K

f. Full Timesharing

TAM - Character Mode		3.5 K
Timesharing Task		5.0
T/S User		
User Context & Monitor Service		
Work Space	2.0 K	
FLAG, BASIC, EDIT, METASYM, COBOL , DMS	18.0	
		20.0
Total Full Timesharing		28.5 K

XOS Supported Hardware

The following is a list of all hardware supported by XOS in the B00 release . The list is in two major groups: Computer Hardware and Peripheral Hardware.

Computer Hardware by Computer Series

Sigma 6 - Model 8310A-G

Model #	Description	Maximum Quantity Supported
8311	Two additional Real-time clocks	1
8316	Additional Register Block	3
8318	Floating Point Arithmetic	1
8321	Priority Interrupt Control Chassis	14
8322	Two Interrupt Levels	8 per 8321
8364	Memory Port Expansion	4
8370	MIOP with 4-byte Interface	4
8375	IOP Expansion Option with 8 Channels	1
8376	Additional 8 Multiplexor Channels	2 per 8370 or 8375
8385	Selector IOP	

Sigma 7 - Model 8401

8411	Two Additional Real-time clocks	1
8413	Power Fail-safe	1
8414	Memory Protect	1
8416	Additional Register Block	3
8418	Floating Point Arithmetic	1
8419	Decimal Arithmetic	1
8421	Priority Interrupt Control Chassis	14
8422	Two Interrupt Levels	8 per 8421
8461	Memory Bank	8
8462	Memory Increment	1 per 8461
8464	Memory Port	6 per 8461
8473	Multiplexor IOP	5
8475	4-byte Interface Feature	1 per 8473
8476	Additional Eight Subchannels	2 per 8473
8485	Selector IOP, Model II	

Sigma 9 - Model 8610A-E

8611	Two additional Real-time clocks	1
8616	Additional Register block	3
8621	Priority Interrupt Control Chassis	14
8622	Two Interrupt Levels	8 per 8421
8664	Memory Port	Up to 10
8670	Multiplexor IOP	5
8671	4-byte Interface Feature	1 per 8670
8672	Additional Eight Subchannels	2 per 8670
8675	MIOP Channel B	
8684	MSRIOP Bus	

Peripheral Hardware

Model #	Description
7012	Keyboard/Printer with Controller
7025	Remote Keyboard/Printer - 35 KSR (10 char/sec., 12 char/in)
7027	Remote Keyboard/Printer with paper tape
7121	200 Cards/Min Reader
7122	400 Cards/Min Reader
7140	1500 Cards/Min Reader
7160	300 Card/Min Punch
7201	RAD Storage Control Unit (Medium Capacity)
7202	RAD Storage Unit - .75 Megabytes
7203	RAD Storage Unit - 1.5 Megabytes
7204	RAD Storage Unit - 3.0 Megabytes
7211	RAD Storage Control Unit (high speed)
7212	RAD Storage Unit - 5.3 Megabytes
7231	RAD Storage Control Unit (extended performance)
7232	RAD Storage Unit - 6.2 Megabytes
7236	Extended Width Controller (for 7232)
7240	Removable Disk Controller
7241	Extended Width Interface Feature (for 7240)
7242	Dual Spindle Removable Disk Storage Unit
7246	Single Spindle Removable Disk Storage Unit
7260	Dual Disk Storage Unit
7261	Single Disk Storage Unit
7265	Dual Disk Storage Unit
7266	Single Disk Storage Unit
7315/6	9-track Magnetic Tape Unit (800 BPI, 60 KB)
7320/2	9-track Magnetic Tape Unit (800 BPI, 60 KB)
7320/3	9-track Magnetic Tape Unit (800 BPI, 120 KB)
7330/2	Phase Encoded Tape Unit (1600 BPI, 60KB, 120KB)
7361	7-track Magnetic Tape Control Unit (low cost)
7362	7-track Magnetic Tape Unit (556 BPI, 20 KB)
7365	BCD Option for 7361
7371	7-track Magnetic Tape Control Unit
7372	7-track Magnetic Tape Unit (200, 556, 800 BPI, 60KB)
7374	BCD Binary Packing Option for 7371
7440	Buffered Line Printer (628-795 LPM)
7441	Buffered Line Printer (820-1100 LPM)
7446	Buffered Line Printer (1000 LPM)
7601	Message Oriented Communications Equipment
7611-7623	Character Oriented Communications Subsystem
7630,7631	Communications Controller and Expansion Unit Package (COC)
7670	Remote Batch Terminal (half duplex only)
2741	IBM Terminal with upper/lower case printing
1033	Dual Access for 7260 Disk
1035	Dual Access for 7265 Disk

The following available features of XOS insure system reliability and maintainability:

- System reconfiguration
  Devices can be varied on or off-line under operator control and may be logically switched.

- Error log file to record hardware errors

- System debug facilities
  Analyze program for itemized and formatted core dumps
  Trace capability to monitor events within the system
  A system debug capability to patch, dump or trace events within the system
  Capability to selectively add patches for permanent storage or optionally at system initialization

- Modularity of the system
  Centralized system tables
  Standardized naming conventions
  Systems modules grouped into functional elements
  Error codes identified by functional element identification
  File management system designed to prevent file loss

- Complete functional, technical and design documentation

- Flexible sysgen capabilities
  Allow adding new processors without re-sysgening
  Allow varying software and hardware configurations without requiring a complete resysgen

- Recovery
  Automatic timesharing recovery
  Complete listing of jobs in execution when the system crashed
  Recovery of and lost disk space
  All symbiont files that were closed are maintained
  Cold start or restart capability. Typical restart time is approximately one minute

SPECIAL FEATURES

Among the features of XOS certain enhancements have been made to the base level of the system for the B00 version. Some of these are:

New I/O Supervisor - This subsystem provides two major improvements in I/O performance.

- Optimized scheduling of disk I/O to minimize arm movement. The queued requests are segmented by device and selected by IOS according to the current arm location.
- Multi-channel access to the same device allows I/O requests to be scheduled via an alternate channel if the primary channel path is busy.

Additional Device Support

- 7260 Disk Drive
- 7265 Disk Drive
- 7446 Printer (all features supported)
- IBM 2741 terminal with full upper/lower case printing capabilities
- TTYs with type-ahead capability

Minimum System Reduction

The system can be both sysgened and operated in 32K.

Additional Enhancements

- Load modules in partitioned data sets.
- Input error bypass.
- Prep of volumes from an AVR keyin.
- Accounting log improvements for job step accounting.
- Tape catalog improvements.
- Output file ID on operator console.
- Lost disk file space recovery
- Extended catalog which allows up to 255 volumes per multi-volume file and up to 8192 files per account or volume.
- Source library control system.

APPLICATIONS AND PROCESSORS

Standard XOS Processors and Utilities

The following processors and utilities are available under XOS:

FMGE - File Management Utility that will

- copy files
- save files
- restore files
- compress (EBCDIC) data into files
- display (decompress) compressed files
- include (add) partitions to partitioned files
- extract and/or delete from partitions files
- list account or volume catalogs
- delete files (one or all)
- list file contents on a line printer
- cards to magnetic medium/magnetic medium to cards

PREP - Preparation of Removable Volumes

- writes volume header labels on disk or tape volumes for subsequent use as standard labelled volumes

REORGP - Reorganize Partitioned Files

- recopies partitioned files
- listing all key names and synonyms and deleting all partition records whose principal keys have been deleted
- lists partition keys and synonyms in order of creation

REORGI - Reorganize Indexed Sequential Files

- disk to intermediate tape
- disk to disk
- disk to intermediate tape and to disk
- tape (sequential or REORGI created) files to disk
- partial processing of files

DEFG - Generation Group Definition

- creates and maintains file/volume generation groups
- OPEN or CLOSE loops
- list all entries
- delete entries

GENER-MEDIA CONVERSION GENERATION

- generate specifically tailored utility
- generate generalized utility
- user input/output own-code exits
- card to tape/disk/printer/card
- tape/disk to printer/card/tape/disk
- record selection
- print file restart by block number or user defined key

GEF - Test File Generator

- through COBOL-like language generate one or more sequential files in any user defined format or pattern

DEBUG - Program Execution Control

- request program dump automatically after program abort or explicity
- SNAP (print) registers and core locations conditionally or unconditionally
- modify or insert instructions into a load module

LINK - Program Linkage Editor

Some of the features are:

- Create executable programs from assembler or compiler generated object modules.
- Create library load modules for insertion into executable load modules in subsequent link operations
- Define a load module tree structure
- Request a load module map
- Modify or insert instructions
- Redefine externally defined symbols
- Create load modules as partitions of partitioned files
- Reference program libraries

OTHER COMPILERS AND PROCESSORS

- SORT
- MERGE
- METASYMBOL
- COBOL (ANS)
- EXTENDED FORTRAN IV
- FLAG
- BASIC
- GPDS
- DMS
- RPG

New Utilities System

- Supercedes and expands capabilities of FMGE, REORGP and REORGI utilities
- Functions will allow copying of files with record field and character format control comparing of files, copying and comparing disk packs, copying and comparing tape volumes, saving and restoring and comparing files to tape, and listing of direct access catalogs

SLM - Source Library Maintenance

- Allows a user to maintain a complete set of source programs with editing and update facilities.

FUTURES

Automatic checkpoint/restart
Dynamic system reconfiguration
Multi-tasking
IRBT Support
Removable volumes for Timesharing users
Dual operator consoles
Greater than 128K support
Transaction Procession Option (TPO)
TAURUS Support
Multiprocessor Support
Memory roll-in/roll-out
Provide the operator with the ability to directly control & modify job scheduling
Enque/Deque facility
Share private disk volumes and account volumes between Sigma 9 computers
OCP Support
SKD support via TAM
Processors: APL, TEXT
RMA Extensions
Forms Control
Performance Monitor