XDS Memorandum

TO Rig Currie

DATE November 11, 1970

FROM Norm Bryga EXT 4302 MAIL STA C6-15

REF NB 172

SUBJECT XOS AND XMS OPERATING SYSTEMS EVALUATION

SUMMARY

I am pleased to present the report documenting our evaluation of XOS and XMS. This report is essentially the same in format and content as the flip chart presentation we prepared for the management meeting on October 21.

We did not draw any conclusions or make any recommendations relative to the XOS - XMS alternatives. We have, however, several related recommendations which are a direct result of this study.

RECOMMENDED ACTION

1. A project should be initiated to create brochures marketing the existing communications capabilities of UTS for business information systems.

2. A project should be initiated to recommend enhancements to UTS required for business information systems.

3. The functional similarity of XDS keyed random file management to IBM's ISAM should be exploited from a marketing point of view, and inherent advantages should be stressed.

4. A project should be initiated to identify enhancements to XDS keyed random files to improve performance during sequential processing of disk-resident files.

BACKGROUND

1. UTS communications capabilities can be applied very effectively to implementing network or dedicated information systems. UTS functions are different from BTAM and QTAM, thereby creating the impression that UTS communications services are not adequate for these applications.

2. UTS provides only a small part of the services required for business information systems. Extending UTS communications functions, would strengthen the multi-use capabilities of UTS. IBM's TCAM should be evaluated as a possible guideline for UTS extensions.

3. There is a great deal of functional similarity between ISAM and XDS keyed random file access method. Marketing should exploit this capability and stress inherent advantages where they exist.

4. The XDS keyed random files are "randomly" distributed on disk files, resulting in excessive disk arm movement during sequential processing (frequent in BDP environments).

E. N. Bryga

mlj

Enclosure: Operating Systems Evaluation Report

xc: Mike Gingrich
Ed Keh
Bob Kemp
Linc Miller

MARKETING DIVISION

OPERATING SYSTEMS EVALUATION

XOS and XMS

CONTENTS

COMPANY PRIVATE

INTRODUCTION

Background

A memorandum from W. F. Glavin on October 8 requested that the pros and cons of XOS and XMS be evaluated prior to any decisions being made to proceed with either or both programming systems. This report is the result of the Marketing Division's study conducted by Norm Bryga, Jim Hargrave, Ed Keh, Bob Kemp and Linc Miller.

Marketing Objectives

The evaluation of any operating system requires an appreciation of the corporation's marketing objectives. For the purposes of this study, we accepted the Marketing Division's proposed strategy, which is:

(a) Maintain and strengthen our traditional real-time and scientific position

(b) Pursue extensions to our traditional markets via multi-use applications where real-time, time-sharing and/or scientific processing is important.

(c) Establish a growth base in the BDP and on-line, transaction oriented marketplace.

Evaluation Method

The objective of the study team was to collect available facts necessary to support Management's XOS-XMS decision. We avoided attempts to draw conclusions, make recommendation, or to bias the information by applying judgment or confidence factors. These, we felt, were part of Management's decision making process.

We identified seven areas which required evaluation. These are: (1) availability, (2) performance, (3) functional characteristics, (4) hardware considerations, (5) expandability, (6) conversion, and (7) logistics (supportability). In addition to the knowledge of XOS and XMS possessed collectively by the study committee, we proceeded to:

(1) meet with the XOS proponents,

(2) meet with the XMS proponents,

(3) evaluate and consolidate our information unto flip charts,

5. EXPANDABILITY

	XOS	XMS
Time-Sharing	Difficult (Estimated 9 man-years minimum, may be implemented by CII)	Exists (UTS)
Communications Management	Difficulty is unknown, but is estimated to be equally difficult under XOS and XMS	
Sigma 9 Extended Memory	XDS effort	Easier, use UTS extensions
Sigma 9 Multi-Processor	Easier, since we could utilize SIRUS 8 concepts	More difficult, but would utilize UTS extensions

6. CONVERSION

XOS	XMS
Easier to convert IBM user to XOS	Easier to convert XDS user to XMS

7. LOGISTICS (SUPPORTABILITY)

XOS	XMS
More difficult and expensive because of outside vendor dependency, translation, merging XDS and CII updates, separate libraries, etc., in addition to that required for UTS.	Would be an extension to UTS, therefore would not require a major additional support and training effort.

(4) present our information to a Marketing Division staff meeting attended by Bill Glavin and George Boyd,

(5) present our information to a joint meeting of the XOS and XMS proponents to calibrate and correct our information,

(6) present part of our information to Don Shaw, Frank Yee and Errol Forkner to test our understanding,

(7) present our information to a management committee considering the XOS-XMS alternatives.

We believe the above procedure has allowed us to meet our objective of presenting highly reliable information.

Information Sources

In the process of collecting information; we interviewed:

George Boyd (Software Product Planning Manager)
Ed Bryan (Programming Development - UTS Manager)
Dan Cota (Programming Development Director)
Buddy Doeppel (Programming Development - UTS, XMS Designer)
Fred Haney (Programming Development - XMS Manager)
Doug Heying (Programming Development - BTM, FOO File Management)
Max Mueller (Program Management - XOS)
Bob Sharpe (Programming Development - XMS, IBM OS Spec)
Wendell Shultz (Program Management - Technology Manager)
John Weaver (Software Product Planning)

The following were the major documents used in the study:

All the published translated XOS (MMP) documents
Most relevant XOS memos and correspondence
Interim Final Report on MMP Evaluation
Memorandum, Wendell Shultz, September 29 on MMP Evaluation
XMS - AOO Functional Specifications
Conformance Report, XMS-AOO, Review Board, September 16

SUMMARY OF ANALYSIS

The following is a summary of the analysis performed on XOS and XMS. The details supporting the statements in this section are included as Section B to this report.

1. AVAILABILITY (First Release)

XOS

Objective date - August 1971

(Initial Release to include multiprogramming; Telecommunications Access Method; disk residence; some XDS program products; no time-sharing; no checkpoint-restart; no real-time)

Risks-

- XDS' success is highly dependent upon CII dedication
- MMP-2 may continue to be unstable
- Translation is a very large problem
- Learning XOS code may be difficult
- Size of delay between CII release and XDS release is unknown
- Work plan non-existant

XMS

Objective date - August 1971

(Initial Release to include multiprogramming; UTS communications services; all XDS program products; time-sharing; basic checkpoint-restart; no disk residence; no real-time)

Risks-

- XMS is planned as an extension of UTS. The UTS architecture may be a constraint in achieving the XMS objectives.
- UTS may continue to be unstable
- Complexity of integrating XMS, FOO FM, and UTS is unknown
- Work plan exists, but past conformance record is poor

2. PERFORMANCE

Insufficient data is available to properly evaluate the performance of XOS and XMS. It was generally agreed that:

(a) XOS is designed to optimize the turnaround of the highest priority jobs, at the expense of resource utilization (total throughput);

(b) XMS is designed to optimize the resource utilization of the system (total throughput), at the expense of the turnaround of the highest priority jobs;

(c) We should not expect throughput to vary by more than 10%; and

(d) Benchmarks could be devised to show either XOS or XMS to be faster.

3. FUNCTIONAL CHARACTERISTICS (Initial Release)

	XOS	XMS
Multi-batch Facilities	Good	Better Features
File Management	Better than XMS, probably faster	Functionally Equivalent to XOS
Communications	Yes (IBM-like approach)	Yes (UTS services different than IBM's and does not support polled terminals now)
Time-Sharing [BTAM]	No (planned)	Yes (UTS)
Real-Time	No (planned by CII)	No (Depends on Marketing's priorities)
Symbionts	Functionally equivalent	More convenient to use
System Management		More flexible
Checkpoint-Restart	(planned for later release)	Basic functions available

4. HARDWARE REQUIREMENTS

	XOS	XMS
Core Residence	14K (min) to 22K (Typical)	About 6K larger (20K to 28K) than XOS (no time-sharing)
7240 Support	Require XDS effort to support under XOS	
System Residence	RAD or Disk	RAD (Disk planned)

CLOSING REMARKS

XOS -In selecting XOS, it must give us a clear advantage in some of the following areas:

- Availability (for benchmarking, demonstration, and installation)
- Performance (particularly in multi-programmed batch)
- Functional capabilities
- Market coverage (batch and transaction processing)
- Expandability to Sigma 9 (extended memory, multi-processor)
- Salability (IBM - like services)

to justify the cost of:

- Purchase price
- Logistics, (translation of documentation, publication, modification)
- Support, (education, duplicate libraries, etc.)
- Product Line Discontinuity

XMS - However, if we select XMS, will it meet:

- Functional Specifications, and
- Delivery Schedule (August, 1971), and
- Batch Performance (throughput)?

ALTERNATIVES

We have identified three reasonable alternatives available to XDS:

(a) Buy XOS, terminate the XMS development effort, and apply those resources to the large logistic problem associated with XOS.

(b) Develop XMS and use XOS resources to reinforce XMS development to assure delivery.

(c) Continue the development of XMS as planned and re-negotiate with CII to allow XDS to evaluate and monitor both XOS and XMS during 1Q71, then reconsider the XOS-XMS choice again. This would be the most expensive alternative. Considering the unknowns of XOS and XMS, this alternative reduces the risks associated with alternatives (a) and (b).

SECTION B

DETAILED ANALYSIS

This section described in further detail the analysis of XOS and XMS under the following headings:

1. AVAILABILITY

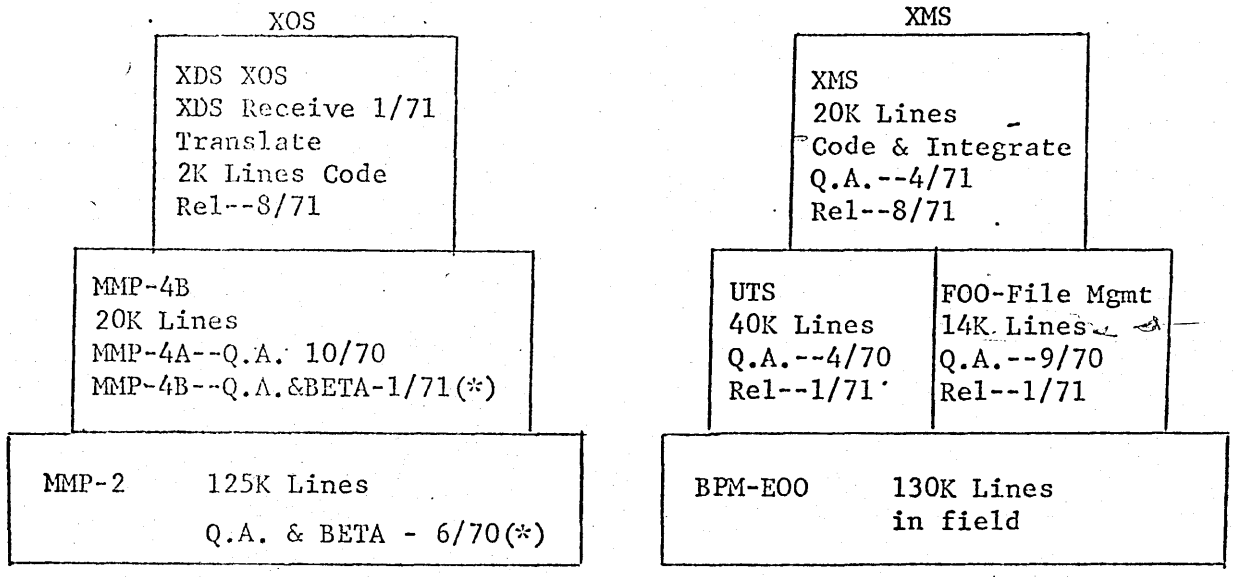
Both XOS and XMS are planned for release to XDS users in August, 1971. XOS represents a sizable translation and logistics problem. No work plan exists, which leads to the premise that this task is not very well understood. On the other hand, a work plan for XMS does exist, supported by detailed Functional and Implementation Specifications. One is reminded, however, that the Programming Division has grossly misjudged task sizes, complexities and schedules in the past.

It is apparent that both XOS and XMS schedules are speculative to a very large extent. The following charts are presented to allow a more rational analysis of stated schedules. These charts represent the amount of code involved, the size of the tasks involved, and the time scale in which these tasks are to be performed.

The identified risks are listed in the Summary of Analysis (Section A of this report).

AVAILABILITY (Continued)

This chart represents the tiers of code upon which XOS and XMS are built. The first tier (lowest) represents the size of existing code and its maturity. The second tier represents the amount of code in the process of creation. The third tier represents the work to be done prior to release in August, 1971.



Multi-programming; Telecommunications Access Method; some XDS program products; disk residence; no time-sharing; no checkpoint-restart; no real-time.

Multi-programming; UTS communications; time-sharing; all XDS program products; basic checkpoint-restart; no disk residence; no real-time.

*It is noted that CII release MMP to Beta test sites at the same time as to Q.A. Similarly, this could also be done for batch only XOS and XMS.

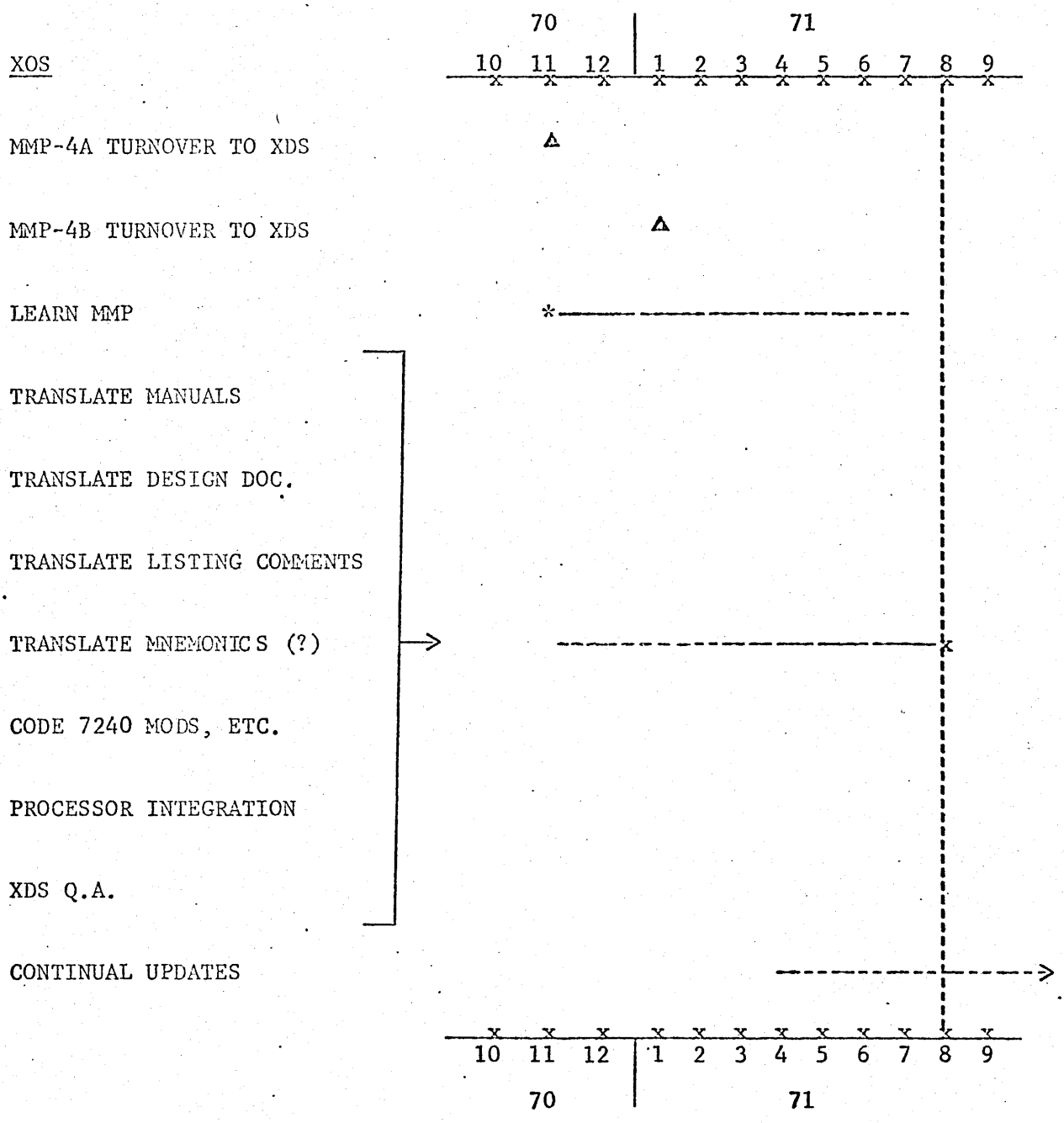
AVAILABILITY (Continued)

This chart represents a further breakdown of the work in process and work to be performed. This chart identifies the number of lines of code since these approximately represent the size of the task to be performed.

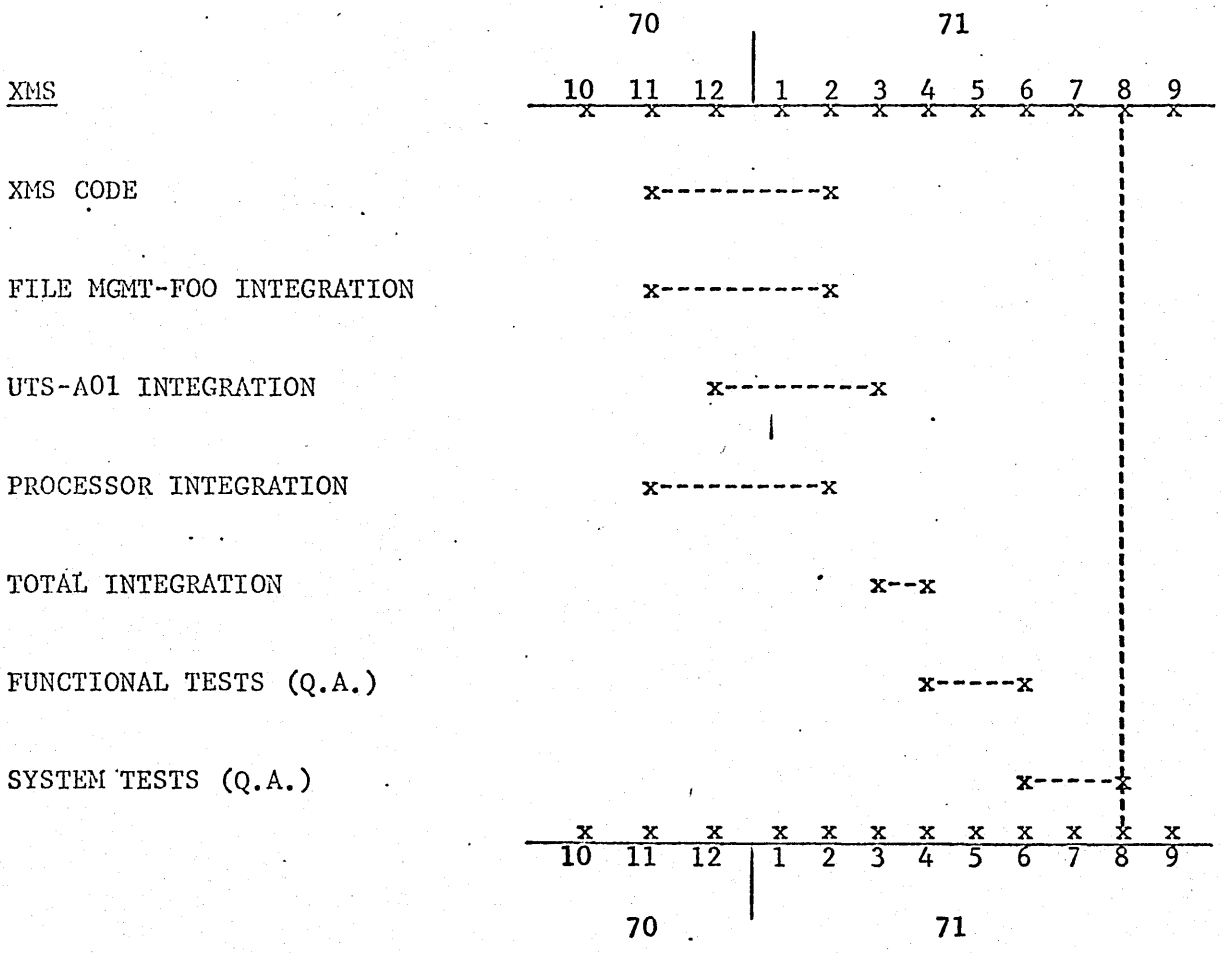
	XOS		XMS	
CODE	20K		20K	
INTEGRATE	20K	to	34K	to
TEST	MMP2 + 20K	1/71	UTS + 34K	4/71
Q.A. (CII) + BETA	MMP2 + 20K	(CII)	N/A	(XDS)
TRANSLATE	MMP2 + 20K		N/A	
LEARN	MMP2 + 20K		N/A	
CODE 7240, ETC.	2K	1/71	N/A	4/71
INTEGRATE PROCESSORS	?	to 8/71	?	to 8/71
XDS Q.A.	MMP2 + 22K	(XDS)	UTS + 34K	(XDS)

AVAILABILITY (Continued)

The following two charts identify the tasks to be performed between now and release on 8/71. In the case of XOS, no work schedule exists. The tasks are therefore grouped together and represented by a broken line with no specific start date.



AVAILABILITY (Continued)



2. PERFORMANCE

Every batch-oriented sales situation is highly competitive, and most involve some degree of benchmarking. The committee spent a large amount of time trying to determine which Operating System offered the greatest throughput advantage.

Available benchmark data could not be used because of accounting differences, and because the XOS benchmarks had not been run under UTS or BPM. Furthermore it is possible to benchmark UTS or BPM only in mono-programming mode. We could not assume, however, that multi-programming performance would be directly proportional to mono-programmed throughput.

In the absence of empirical data, an attempt was made at a deductive analysis of the systems' architecture.

XOS is designed to optimize the turnaround of the highest priority class, even though some available system resources may not be utilized. The performance capability of XOS File Management is considered to be better than XMS.

On the other hand, XMS is designed to optimize total throughput by optimizing the utilization of system resources. In accomplishing this, the turnaround of the highest priority class would be reduced.

It was generally agreed that we should not expect a performance difference greater than 10%. A difference of only 10% was not considered significant, and may be highly sensitive to the make-up of the benchmark.

3. FUNCTIONAL CHARACTERISITCS

I Multi-Programming

	XOS	XMS
(a) Job Scheduling	There are three job classes: parallel (highest priority), production (five sub-classes) and serial. Jobs in the parallel class are scheduled FIFO (until job queue or class resources exhausted), production sub-classes are scheduled FIFO (one job per sub-class), and serial jobs are scheduled in priority sequence. If resources are not available to schedule a parallel or production job, no further lower class or sub-class jobs will be scheduled until resources become available. If for some reason the active job in a sub-class is blocked, the rest of the jobs are not scheduled. If a serial job is not scheduled because of lack of resources, the lower priority serial jobs are looked at for scheduling purposes. Class and sub-class priorities are specified at system generation time and may be modified only at system start-up time.	Up to 16 job classes with up to 15 priorities in each class may be defined. Jobs are scheduled within each class in priority sequence. If the highest priority job within a class cannot be scheduled, the next highest priority job may be scheduled. The class specification may be dynamically changed by the operator Via key-in, the operator may change the priority and the attributes of the jobs under any class. Generally, XMS job scheduling is more flexible.

FUNCTIONAL CHARACTERISTICS

I. Multi-Programming (Continued)

	XOS	XMS
(b) CPU Dispatching	Available CPU time is allocated to classes and sub-classes in the priority sequence defined at system generation. When a job is in wait state, CPU time is then allocated to the next priority job. This scheme optimizes turnaround of the highest priority job and is preferred by BDP users. Control must be exercised, however, to prevent placing compute-bound jobs in the highest class where they can severely degrade turn-around of lower-priority I/O-bound jobs.	CPU time is allocated to classes by the quantum rotational method. Turnaround time of all jobs is approximately proportional to their execution time. This scheme is not acceptable to BDP users who want maximum turnaround for the highest priority jobs. According to Programming Development, the existing queuing and quantum structure can be easily modified to (optionally) optimize the turnaround of highest priority jobs.

(c) Resource Allocation

Although slight differences exist, both systems were judged equivalent in Resource Allocation. Both require that maximum resources for the entire job be available before a job may be scheduled. Both allow jobs to be started with less than the total required resources with a finite chance of "deadlock."

	XOS	XMS
(d) Memory Management	Batch jobs, once in core, are not swapped out even in long wait situations such as resource blockage or volume mounting/dismounting.	XMS memory management is superior in that long wait times are reduced by the swapping mechanism. Time quantum needs to be adjusted to avoid swapping during short I/O times (e.g., disk seek).

FUNCTIONAL CHARACTERISTICS

I Multi-Programming (Continued)

	XOS	XMS
(e) Conditional Execution	Job predicate relationships are determined by the physical order of appearance in the job stream. Conditional execution is implemented via the Job Switch Word.	Predicate job scheduling and job step forking are accomplished through SCHEDULE and BATCH commands. Conditional execution may be controlled by command language and logical expressions, a more flexible arrangement.
(f) Peripheral Utilization	System resources are released at job end.	Although roughly equivalent to XOS (see Resource Allocation), the FREE command enables users to release peripherals before job end if appropriate. Dynamic reassignment of system resources to privileged job classes also contribute to better peripheral utilization.
(g) Sharable In-Core Libraries	Not sharable, separate copies required.	User library routines and language processors (except COBOL) are sharable.

II File Management

	XOS	XMS
(a) Cataloging	The catalog structure, automatic updating of geneological catalogs, and non-automatic updating capabilities of XOS and XMS are approximately equivalent, however, XMS allows job-end independent cataloging.	

FUNCTIONAL CHARACTERISTICS

II File Management

	XOS	XMS
(b) File Generation	The file generation, version facilities and ability to reference files are similar in XOS and XMS.	
(c) Indexed Sequential Access Method (ISAM)	The ISAM file creation, update, access and index scheme are the same as IBM. There is only one key entry plus pointer in the index per data block resulting in smaller space utilization for indices. This indexing scheme enhances sequential access activity.	The XDS keyed random file is functionally equivalent to XOS ISAM. There is one key entry plus a 14 byte pointer per data record, resulting in a larger space utilization for indices. This indexing scheme enhances highly random access activity.
(d) Multi-Volume Disk Files	May be mounted sequentially (one at a time) or in parallel (all at the same time).	Must be mounted in parallel (all at the same time) even if processed sequentially.
(e) Tape Label Formats	Supports the ANS and IBM tape label formats.	Supports the ANS, IBM and BPM/UTS tape label formats.
(f) File Concatenation	(The ability to link several files to form a super file)	
	Only tape files with the same physical characteristics may be concatenated.	Both tape and disk files may be concatenated and may be catalogued as a file group in a sub-file catalog.

FUNCTIONAL CHARACTERISTICS

II File Management (Continued)

	XOS	XMS
(g) Partitioned Files (Useful for transaction processing)		
	Access to partitioned files provided.	None
(h) Locate Mode		
	(Locate mode allows faster I/O processing)	
	Available for tape and disk files.	Available for tape files only.
(i) Internal Secondary Storage Address		
	Information on file space allocation is maintained in a system separated directory. This gives each file a virtual file space independent of real file space, providing better device independence, reliability and modularity.	File space allocation uses real addresses by chaining portions of a file together through the actual contents of a file.
(j) Variable Block Lengths		
	User specified for tape or disk.	User specified for tape only.

(k) Record Contention Resolution

This facility, a requisite for transaction processing, is provided by neither system. Contention is resolved at the file level.

III Communications Management

This subject took more of our time than any other. There are basic differences between the XOS and XMS (UTS) communications services which tended confuse the analysis of their functional capabilities. This section will firstly compare the functional capabilities, and then contrast the differences between the two implementations. In some cases, these differences have no effect on functional capabilities. The services described for XMS presently exist in UTS.

FUNCTIONAL CHARACTERISTICS

III Communications Management (Continued)

	XOS	XMS (UTS)
(a) Bi-point	XOS (TAM) and XMS (UTS) are equivalent in their ability to allow a single terminal and a single program to communicate with each other.	
(b) Multi-Drop (Polling)	Support for polled terminals is available under TAM.	Programming Development estimated 6 man-months to support polled terminals on multi-drop lines.
(c) Queuing (QTAM provides the ability to queue messages from a communication network in core or disk, process these messages under program control and return the messages to a queue, then output these messages from the queue to the communication network with the same or different destinations.)	A QTAM capability is planned for XOS.	XMS (UTS) allows a message to be received from a terminal, processed and placed in a file (on disk) associated with another terminal. The program associated with the other terminal can retrieve the message from the file and output it to the destination terminal. Since XMS processors are sharable, the tw programs described above could actually be the same piece of code. Many terminals, constituting a network, can be handled in this fashion. The functional equivalence of QTAM is, therefore, inherent in UTS.

FUNCTIONAL CHARACTERISTICS

III Communications Management (Continued)

	XOS	XMS (UTS)
(d)	Automatic Log-On Terminals are "logged" into the system and checked/validated at the time the service program is invoked at the computer center.	XMS (UTS) provides the ability to automatically associate a specific service program with a specific terminal at the time the terminal logs-on. An enhancement is planned to recognize a terminal by line or poll location.
(e)	Multi Terminal Programs Available via TAM.	The ability of a single (sharable) processor to handle multiple terminals is inherent in UTS (See paragraph (c) above.)
(f)	Dial Out Not available in either system. Will require an estimated 6 man-months to implement.	
(g)	Network Definition TAM provides the ability to define a network which identifies lines and terminals on those lines, and their polling sequences. Changes to the network and polling lists are controlled centrally by program or operator control.	UTS recognizes the presence of terminals, hence a network, at log-on time. The network is implied and re-defined automatically as terminals log-on and sign-off. This capability is inherent to UTS.
(h)	Similarity to IBM Approach Since XOS TAM is similar to IBM BTAM, it represents a selling advantage.	The UTS service is different from IBM and represents a marketing difficulty.

FUNCTIONAL CHARACTERISTICS

III Communications Management (Continued)

(h) Differences in Implementation. (These differences do not imply a functional advantage or disadvantage unless specifically stated.)

	XOS	XMS (UTS)
(i)	Each terminal is dedicated to a specific libraried program called from the central site, although that program can call several associated sub-systems.	Any terminal (or device) can call and run any libraried program and its sub-system from a remote location.
(ii)	Buffers are in user area (fixed) -Additional user concern -Potentially use more core	Buffers are in system area (dynamic) -No user concern -Potentially save core space
(iii)	User's program continually resident even if no terminal activity. -Multiple on-line applications require more core space. -No significant difference from UTS for single application with high terminal activity.	User's program swapped out if no terminal activity. -Multiple on-line applications swapped into same core space.
(iv)	Terminal-to-terminal communications needs processor similar to UTS COMSYS.	COMSYS in development for UTS will provide terminal-to-terminal communications.

FUNCTIONAL CHARACTERISTICS

III Communications Management (Continued)

	XOS	XMS (UTS)
(v)	Use File Management System files with OPEN/CLOSE activated only at start/ end of job -Record contention is resolved within user's program.	File OPEN/CLOSE overhead is high, but is being avoided for UTS "internal communication" by using symbiont files. -Symbiont files not compatible with user files. -Record contention resolution by File Management would eliminate this problem.

IV Time-Sharing

	XOS	XMS
(a)	Availability Confidential CII plans are to implement time-sharing under XOS with a target date of Oct. 71 (XDS release no sooner than 1Q72).	Exists in UTS.
(b)	Compatibility If implemented by CII, the system would likely be incompatible with BTM or UTS.	Compatible with UTS.
(c)	Difficulty of Implementation Programming have estimated that it would take XDS at least 9 man-years to produce a BTM-like system. (BTM has taken 14 man-years to date.)	Exists in UTS.

FUNCTIONAL CHARACTERISTICS

IV Time-Sharing (Continued)

	XOS	XMS
(d) Service Level (Number of terminals, response, etc.)	Unknown, but estimated to be ~~similar~~ to BTM.	Like UTS.

V Real-Time

Not available in first release of either system. We estimated that expected functions and service levels would be similar.

VI Symbionts

XOS	XMS
Media conversion service programs, functionally equivalent to extended symbionts, are filed and scheduled as user jobs.	Extended symbionts are system services and are more convenient to use.

VII System Management

(a) Number of Operator Consoles

XOS	XMS
1	1

(b) Dynamic System Tuning

XMS (UTS) appears more complete.

(c) Crash Recovery

Not enough information is available to draw a meaningful comparison. Both systems, however, have a demonstrated recovery capability.

(d) Checkpoint/Restart

XOS	XMS
None in the first release, but a capability is planned (Specifications are unknown).	Program initiated checkpoint/restart capability will be provided in first release.

(e) On-line Diagnostics

XOS	XMS
None	Available (UTS)

FUNCTIONAL CHARACTERISTICS

VIII Job Control

	XOS	XMS
(a) JCL Edit	Equivalent in XOS and XMS.	
(b) Catalog Procedures	A catalog procedure cannot reference another catalog procedure (one level)	A catalog procedure can reference multiple catalog procedures (multi-level)

IX Processors

A separate set of FORTRAN, COBOL, Meta-Symbol, Sort, etc. processors would have to be maintained for XOS.	Use same processors as UTS/BPM.

4. HARDWARE CONSIDERATIONS

	XOS	XMS
Core Residence	XOS can reside in as little as 14K words, but performance is noticably slow. We estimated that a 22K residence area would assure acceptable throughput for a typical configuration.	XMS (no time-sharing) will require a 20K minimum residence area, and 28K for a typical configuration with acceptable performance. (6K larger than XOS)
Time-Sharing	Estimate additional 4K residence.	Additional 4K residence.
7240 Support	Require XDS effort to support under XOS.	
System Residence	RAD or Disk	RAD (Disk planned)

In summary, XOS with its smaller core residence requirements and disk residency offers a clear advantage for medium-scale batch systems such as the Sigma 6. Both systems offer larger core residence and RAD residence where these can be marketed as performance features on large-scale configurations such as Sigma 9.

5. EXPANDABILITY

	XOS	XMS
Time-Sharing	Confidential CII plans are to implement time-sharing under XOS with a target completion date of Oct. 71. (XDS release no sooner than 1Q72) If implemented by CII, the system would likely be incompatible with BTM or UTS. Programming have estimated that it would take XDS at least 9 man-years to produce BTM-like time-sharing under XOS. (BTM has taken 14 man-years to date). The service level (no. of users, response) of XOS time-sharing and impact on batch throughput are unknown at this time, but are estimated to be similar to BTM.	Exists in UTS.
Sigma 9 Extended Memory (beyond 128K)	Would require separate XDS effort.	Utilize UTS changes planned for release 9/71.
Sigma 9 Multi-Processor	XDS may be able to utilize the design of the CII multi-processor operating system, SIRUS 8 (Q.A. 6/71). However, there is enough difference between the Sigma 9 and IRIS 80 that it may require a significant XDS effort.	UTS multi-programming has not been designed. Although this may be difficult, XMS would utilize this effort.

EXPANDABILITY (Continued)

	XOS	XMS
Future Peripherals	Would require separate effort if CII did not support XDS devices or controllers (e.g., 7240). However, if XDS provided support for new peripherals, it would enhance the sale of these devices to CII.	Utilize UTS support.

Communications Management System

The existing communications capabilities of XOS and XMS are discussed under Functional Characteristics. Both of these capabilities are considered equally small when viewed within the much larger context of a Communications Management System. It appears, however, that the inherent differences between XOS and XMS would result in a different implementation of CMS. A further study would be required to probe the functional differences, their desirability, salability, and difficulty of implementation.

6. CONVERSION

Conversion is a significant consideration in most BDP user's decision to upgrade to a different computer system.

In our analysis, we considered the difficulty of converting files, programs and job control language. Several schemes were tried and all gave roughly equivalent results. The following table provides an approximation of the difficulty of conversion using a BPM to XMS conversion factor of 1.

	to XMS	to XOS
from BPM	1	6
from IBM	6	3

7. LOGISTICS

	XOS	XMS
Translation	Four levels of translation have been identified: (a) Reference and User's Manuals, required for Beta test sites and user release. These documents require a high level of editing. (b) Internal Design Documents, required for XDS modifications. and Q.A. (c) Listing comments, required for XDS modifications. (d) Mnemonics (symbolic labels), required for major XDS code changes, but may be computerized using a one-for-one replacement. The above levels of translation are required at initial turnover, and continually as updates are released to XDS.	None
Administration	The administrative problem includes coordination of XDS-CII working relations, translation of updates, and merging XDS modifications with CII updates to create new releases. We anticipate a delay of 4 to 6 months between CII release of XOS SIDIR's and updates and their release to the field by XDS.	Because it is an in-house effort, the administration of XMS is greatly simplified. The lack of a 4 to 6 month time delay as in XOS means that XMS could mature more quickly after initial release.

LOGISTICS (Continued)

	XOS	XMS
Maintenance	The maintenance problem is complicated by language differences, and potential differences in CII and XDS objectives for XOS.	Large part is common with UTS.
Customer and Field Support	Sales, Analyst and Customer training would have to be duplicated for XOS and UTS.	Large part is common with UTS.
Libraries	A duplicate set of program libraries would have to be maintained for XOS as well as UTS.	Common with UTS.
Other Project Impact	XOS, in requiring a higher level of XDS resources, would have a higher impact on other XDS projects than XMS.	