File Number 7090-36
Form C28-6248-1

IBM Systems Reference Library

IBM 7090/7094 IBSYS Operating System

System Monitor (IBSYS) Version 11

This publication describes the use and general function of the System Monitor of the IBM 7090/7094 IBSYS Operating System (7090-PR-130). The IBSYS Operating System consists of a number of commercial and scientific programming aids operating under over-all control and direction of the System Monitor (7090-SV-918). It is designed to process sequentially a variety of unrelated jobs with little or no operator intervention.

The System Monitor includes the System Supervisor, the System Core-Storage Dump Program, the System Editor, the System Nucleus, and the Input/Output Executor. In general, the "Introduction" of this publication and the sections describing the System Supervisor and the System Core-Storage Dump Program are directed primarily to the applications programmer. The remaining sections are directed to the systems programmer.

Preface

This publication describes the use and general function of the System Monitor of the IBM 7090/7094 IBSYS Operating System and provides information for maintaining the system. The IBSYS Operating System may be considered an integral part of the IBM 7090/7094 Data Processing System. It consists of a comprehensive set of programming aids operating as subsystems under a master System Monitor.

The System Monitor described in this publication encompasses the System Supervisor, the System Core-Storage Dump Program, the System Editor, the System Nucleus, and the Input/Output Executor. The subsystems operating under the System Monitor are described in separate publications. These publications are referred to in the "Introduction" section of this manual and are listed in Figure 2 in the "System Supervisor" section. Instructions for the operator of the system are provided in a separate publication entitled *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

This publication on the System Monitor is divided into seven major sections. The first three sections, "Introduction," "System Supervisor," and "System Core-Storage Dump Program," are directed primarily to the applications programmer. The remaining sections are directed primarily to the systems programmer. The systems programmer is an experienced programmer who is assigned to place the IBSYS Operating System into operation, modify it according to the special requirements of his installation, maintain it, and ensure adequate control over its content and use.

All of the System Monitor control cards that might be used by the applications programmer in programming a job are described in the section on the System Supervisor. However, very few of the cards are actually required for most jobs. Therefore, these cards are described first, for the benefit of the reader who is not a full-time programmer and who is only interested in the control cards required to run an average job, such as a FORTRAN compilation and execution. The control cards described in the remainder of the section are primarily of interest to the more experienced programmer, the systems programmer, and the operator.

The reader of this manual is assumed to be familiar with the contents of the IBM reference manual *IBM 7094 Data Processing System*, Form A22-6703.

MAJOR REVISION (February 1964)
This edition, Form C28-6248-1, is a major revision of the previous edition, Form C28-6248, and makes that publication obsolete. This edition also supersedes the publication *IBM 7090/7094 Operating System: Basic Monitor (IBSYS), Preliminary Specifications for 7340 Capability,* Form J28-6279.

Copies of this and other IBM publications can be obtained through IBM Branch Offices.
Address comments concerning the contents of this publication to:
IBM Corporation, Programming Systems Publications, Dept. D91, PO Box 390, Poughkeepsie, N. Y. 12602

Contents

Function

The 7090/7094 IBSYS Operating System consists of a comprehensive set of commercial and scientific programming aids operating as subsystems, under executive control and coordination of a System Monitor. The Monitor, by coordinating the operation of the subsystems, enables a series of unrelated jobs to be processed with little or no operator intervention. By reducing the degree of human participation in the mechanics of data processing, the Operating System ensures that jobs are processed faster, more efficiently, and are less subject to human error. As a result, turnaround time, the interval between the time a programmer submits a job for processing and the time he receives results, is significantly reduced.

Organization

System Monitor

The general organization of the System Monitor, as well as its general relation to the subsystems operating under it, is illustrated in Figure 1. The System Monitor consists of:

1. The System Supervisor, whose primary function is to control and coordinate the processing of jobs by passing control from one subsystem to another.
2. The System Core-Storage Dump Program, which may be used to facilitate the testing and analysis of any program executed under control of the Operating System.
3. The System Editor, which provides the systems programmer with facilities for modifying and maintaining the System Monitor and the subsystems operating under it.
4. The System Nucleus, which remains in core storage at all times and provides common facilities for intercommunication and control among the subsystems and between the System Monitor and the subsystems.
5. The Input/Output Executor, which normally remains in core storage to coordinate and control input/output and other trapping operations.

The System Monitor may also contain an installation accounting routine tailored to the specific requirements of the installation.

The subsystems operating under the System Monitor (Figure 1) provide the programmer with a variety of programming aids which he may use singly or in combination to process a particular job. Each of these subsystems is described briefly below.

IBJOB Processor

The IBJOB Processor is the major subsystem of the IBSYS Operating System. It is a highly integrated processor that can be used to compile, assemble, load, and execute programs written in FORTRAN IV or COBOL language. It can also be used to assemble, load, and execute programs written in the Macro Assembly Program (MAP) language or to load and execute previously assembled object programs. Facilities are provided for combining program segments written in different languages with previously assembled segments to form a single executable object program.

The IBJOB Processor contains a complete library of relocatable subroutines, including a complete Input/Output Control System (IOCS). When an object program is loaded preparatory to execution, only those portions of IOCS that are actually required are loaded.

The IBJOB Processor and its use are described in detail in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6275.

FORTRAN II Processor

The FORTRAN II Processor can be used in either of two modes, the FORTRAN mode or the IBSFAP mode.

In the FORTRAN mode, the FORTRAN II Processor can be used to compile, assemble, load, and execute source programs written in FORTRAN II language. It can also assemble, load, and execute programs written in FORTRAN II Assembly Program (FAP) language and load and execute previously assembled object programs. Facilities are provided for combining program segments written in FORTRAN II and FAP languages with previously assembled segments to form a single executable object program. Facilities are also provided for chaining core storage loads so that executed portions of a program may be overlaid with portions yet to be executed.

In the IBSFAP mode, the FORTRAN II Processor can be used to assemble, but not load and execute, programs written in FAP language. An assembled object program can be loaded and executed under control of the Input/Output Control System or the FORTRAN mode of the FORTRAN II Processor. The IBSFAP mode of the FORTRAN II Processor can also be used to up-

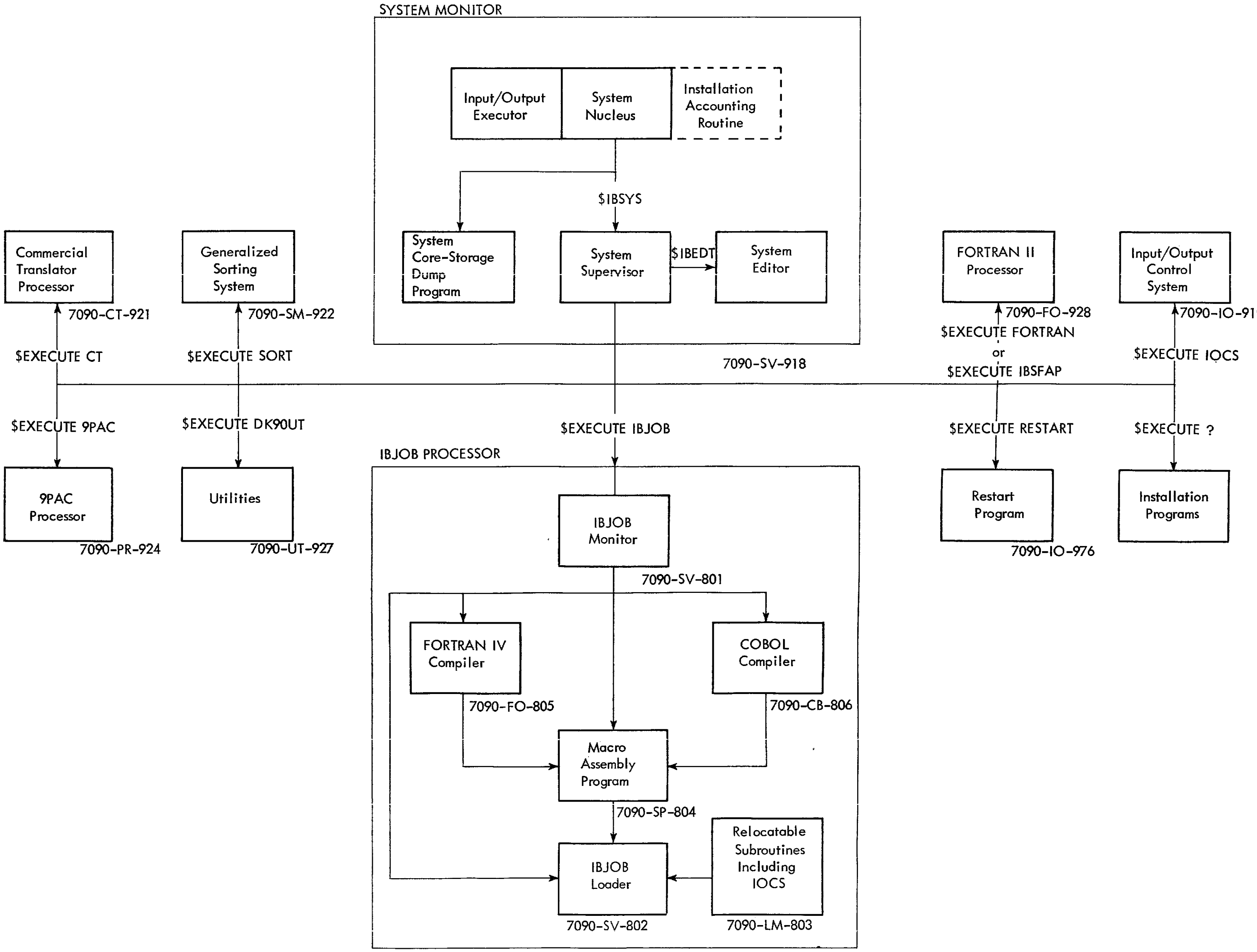


Figure 1. IBM 7090/7094 IBSYS Operating System

date symbolic tapes by changing, deleting, or adding instructions.

The FORTRAN II Processor and its use are described in detail in the publication *IBM 7090/7094 Programming Systems: FORTRAN II Operations*, Form C28-6066.

The FORTRAN II Assembly Program and its use are described in the publication *IBM 7090/7094 Programming Systems: FORTRAN II Assembly Program (FAP)*, Form C28-6235.

Input/Output Control System

The Input/Output Control System (IOCS) provides input/output control for programs assembled by the FORTRAN II Processor. It relieves the programmer of the task of writing complex input/output routines by automatically controlling the blocking and unblocking of data records, the overlapping of processing with input and output, and the preparation and checking of labels. Only those portions of IOCS actually required are loaded with the assembled object program.

IOCS and its use are described in detail in the publication *IBM 7090/7094 IBSYS Operating System: Input/Output Control System*, Form C28-6345.

Restart Program

Unlike the other subsystems operating under the System Monitor, the Restart Program is used exclusively by the operator of the system. It is designed to enable the operator to restart an interrupted program using a checkpoint record recorded by IOCS before the interruption occurred.

The use of the Restart Program is described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

Commercial Translator Processor

The Commercial Translator Processor may be used to compile, assemble, load, and execute programs written in the IBM Commercial Translator language.

The Commercial Translator Processor and its use are described in the publication *IBM 709/7090 Commercial Translator Processor*, Form J28-6169.

Generalized Sorting System

The Generalized Sorting System can be used to sort fixed-length or variable-length records, written in either binary or decimal mode. The control fields of the records may be signed or unsigned. The records can be sorted in ascending or descending order, using either the commercial or scientific collating sequence.

The Generalized Sorting System and its use are described in the publication *IBM 7090/7094 Generalized Sorting System: 7090/7094 Sort*, Form C28-6307.

9PAC Processor

The 9PAC Processor can be used to establish and maintain data files and to generate reports on the data in the files.

The 9PAC Processor and its use are described in the following publications:

IBM 7090 Programming Systems, Share 7090 9PAC
- *Part 1: Introduction and General Principles*, Form J28-6166
- *Part 2: The File Processor*, Form J28-6167
- *Part 3: The Reports Generator*, Form J28-6168

Utilities

The Utilities consist of a tape dump routine for 729 Magnetic Tape Units and 7340 Hypertape Drives and of the following for 1301 Disk Storage and 7320 Drum Storage: format track generation, home address and record address identification, load disk/drum, dump disk/drum, restore disk/drum, and clear disk/drum.

The utilities and their use are described in the following publications:

IBM 7090/7094 IBSYS Operating System: Utilities, Form C28-6364

IBM 7090/7094 Utility Routines for IBM 1301 Disk Storage, Form J28-6223

User Programs

In addition to the subsystems described above, the user of the IBSYS Operating System may design programs and incorporate them as subsystems operating under the System Monitor. Conversely, the user may remove subsystems or portions of subsystems that are not required at his installation.

Application

In programming a job, the programmer may employ any logical combination of the subsystems operating under the System Monitor. The programmer, in effect, controls and directs the processing of his job from his desk by inserting the proper control cards in his job deck. Before a particular job is processed, it may be combined with other jobs to form a single input file of unrelated jobs. The input file of jobs can then be processed by the Operating System without costly setup delays between jobs or job segments while the data processing system lies idle.

The operator of the system, for the most part, performs relatively routine functions, such as loading or unloading tape reels. Usually he is told what to do and when by means of an on-line printout from the Operating System. If the Operating System, owing to a programmer error, cannot complete a job or job segment, it automatically skips to the next job or job segment without intervention by the operator. However, the operator can, if he chooses, interrupt the Operating System at the end of any job. By means of control cards, he can then direct the Operating System to perform any one of several operations, for example, to restart at the beginning of another job on the input file.

If an error occurs during the execution of an object program, the operator, the Operating System, or the object program itself can call for a post-mortem dump of any part or all of the core storage (to facilitate analysis of the error) followed by an automatic skip to the next job segment. An object program can also, at any point in the program, call for a snap dump of any part or all of core storage. At the completion of the snap dump, the contents of core storage are restored and the execution of the object program is resumed. Any one of six formats can be specified for a core storage dump (Figure 6).

When an input file of jobs is completed, the Operating System stops after providing the operator with information on the status of the system input and output files. Then, by the use of control cards, the operator can direct the Operating System to perform any one of a number of operations. For example, he may direct it to restart at the beginning of a new input file or to rewind and unload the input and output files.

System Unit Functions

To ensure continuous job processing and proper coordination between subsystems, the System Monitor provides a logical framework for assigning input/output units to specific functions and for keeping track of the exact status of all units at all times. Some units are assigned system unit functions, that is, they are assigned specific functions required by the nature of the Operating System and may be used in that capacity by the System Monitor and each of the subsystems operating under it. For example, at least one unit is used as a system input file on which a series of jobs are stacked so that they can be processed continuously by the System Monitor and the various subsystems. Any units not assigned to system unit functions are available for use by the programmer, provided they are not logically detached from the Operating System. The following are the system unit functions to which units may be assigned. Some functions may not have units assigned to them, depending on the requirements of a particular installation.

Information as to which types of input/output devices can be assigned to specific system functions for a particular subsystem may be found in the *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355, under "Use of Input/Output Units."

System Input

A System Input Unit is required by every installation for use as a common job input file. Normally, the input file tape is prepared off-line on an auxiliary 1401 Data Processing System. The input file may contain System Monitor and subsystem control cards, symbolic source programs, binary object programs, and data.

System Output

A System Output Unit is required by every installation for use as a common output print file. The output file will contain messages from the Operating System and may contain source program listings, core-storage dump listings, and output data.

System Peripheral Punch

A unit must be assigned as a System Peripheral Punch Unit. The peripheral punch file may be processed off-line to produce object program card decks.

The same unit can be assigned as both the System Output Unit and the System Peripheral Punch Unit. However, when this dual assignment is used, FORTRAN II programs can not be executed. In addition, some output may be lost when the combined unit is backspaced to suppress punched output due to errors in relocatable FAP or IBSFAP assemblies.

System Library

At least one unit must be assigned as a System Library Unit on which the IBSYS Operating System itself is recorded. Up to four units may be assigned as library units. When magnetic tape units are used, the Operating System may be duplicated on two System Library Tapes which can be referred to alternately in order to reduce delays in processing while the library tape is being rewound. The Operating System can also be split between two or more library tapes in order to reduce the time required to gain access to particular parts of the Operating System.

System Utility

Four System Utility Units are required by every installation for general use by the System Monitor and the subsystems operating under it. The System Utility Units may also be used by object programs. However, files which are to be retained must not be assigned to System Utility Units that will be used by the Operating System during a job.

A unit assigned as a System Utility Unit may be simultaneously assigned as a System Checkpoint Unit.

System Checkpoint

A unit may be assigned as a System Checkpoint Unit on which checkpoint records are recorded under IOCS control. The same unit may also be assigned as a System Utility Unit.

System Printer

A 716 Printer is required by every installation. It is used by the Operating System to record messages to the operator.

System Card Reader

A 711 Card Reader is assigned as the System Card Reader. Normally, the card reader is used by the operator for inserting control cards that direct the Operating System. The card reader may also be used as a substitute input unit for processing small job files.

System Card Punch

A 721 Card Punch may be assigned as the System Card Punch. The punch may be used to punch control cards for use by the operator, or it may be used as a substitute output unit.

Alternate Units

A second unit may be assigned to the system input, output, peripheral punch, or checkpoint functions. If a second unit is assigned, it serves as an alternate unit to eliminate delays due to reel switching. When an end of tape is reached on one unit an automatic switch can be made to the alternate unit. In the distributed version of the Operating System, the same 729 Magnetic Tape Unit is assigned to a system utility function (SYSUT4) and to the alternate peripheral punch function (SYSPP2). Therefore, automatic reel switching should not be attempted unless different units are assigned to these functions.

Machine Requirements

The following minimum machine configuration is required for use of the 7090/7094 IBSYS Operating System:

1. An IBM 7090/7094 Data Processing System.
2. Three IBM 729 Magnetic Tape Units or IBM 7340 Hypertape Drives for assignment as System Input, Output, and Peripheral Punch Units. A single unit may be assigned as a combined System Output and Peripheral Punch Unit (but note restrictions stated under "System Peripheral Punch"). If this is done, the extra unit may be assigned as a second System Library Unit.
3. Five units; one for assignment as a System Library Unit and four for assignment as System Utility Units. These units may be any combination of IBM 729 Magnetic Tape Units, 7340 Hypertape Drives, or selected cylinders of random access storage units. (In the remainder of this publication, *random access storage*, when used, will refer to both IBM 1301 Disk Storage and IBM 7320 Drum Storage.)
4. One IBM 716 Printer for assignment as the System Printer.
5. One IBM 711 Card Reader for assignment as the System Card Reader.

The peripheral support program for the IBM 7090/7094 IBSYS Operating System requires an IBM 1401 Data Processing System with the following features:

1. 4000 locations of core storage
2. Advanced Programming feature
3. High-Low-Equal Compare feature
4. Sense Switch feature
5. One IBM 729 or 7330 Magnetic Tape Unit
6. One IBM 1402 Card Read Punch Unit with the Column Binary feature.
7. One 1403 Printer with 132 print positions and the Print Control feature

System Supervisor

Function

The primary function of the System Supervisor is to coordinate and supervise the processing of jobs by:

1. Passing control from one subsystem to another.
2. Restoring unit assignments between jobs.
3. Controlling interruption by the operator.
4. Skipping jobs or job segments when directed by a subsystem.

In addition, the System Supervisor can be directed by control cards to perform a variety of other functions, such as changing unit assignments, manipulating tape units, and passing control to the System Editor. The System Supervisor is directed to perform its functions mainly by means of control cards which it reads from a system input file, interprets, and acts upon.

Definition of a Job

Of the many control cards that are recognized by the System Supervisor, the key cards in controlling the continuous processing of jobs are the $JOB card, $EXECUTE card, $IBSYS card, and $STOP card. Each of these cards is recognized and acted upon by each of the subsystems operating under control of the System Supervisor, as well as by the System Supervisor.

Each job in a stack of jobs on an input file is considered to be entirely independent of any other job. The $JOB card is the first card in each job deck and may contain the name of the particular job it precedes. A job consists of all of the cards beginning with a $JOB card and ending with, but not including, the next $JOB card. A job may consist of any logical combination of job segments to be performed by the subsystems and the System Monitor. When a $JOB card is recognized by the System Supervisor or a subsystem, the System Supervisor will, if necessary, restore to its original assignment any unit that was reassigned or made unavailable during a previous job, with the exception of the following:

1. Any unit that was logically detached from control by the Operating System.

2. Any unit that was assigned to a system unit function in place of a detached unit.

3. Any unit that was assigned to a system input, system output, or system peripheral punch function.

A unit originally assigned to a system input, output, or peripheral punch function is not restored because it may have been validly replaced by its alternate unit when an end of tape was encountered.

A $JOB card may be followed by a $EXECUTE card, $IBSYS card, or $ID card. If a $ID card, which is an optional card used for intrajob accounting purposes, follows the $JOB card, it must itself be followed by either a $EXECUTE or $IBSYS card.

The $EXECUTE card contains the name of a subsystem and is used to define the beginning of a job segment to be performed by the specified subsystem. When a $EXECUTE card is recognized by the System Supervisor, it passes control to the subsystem specified on the card. When it is recognized by a subsystem, the subsystem retains control and reads the next card, provided it is the subsystem specified on the card. Otherwise, the subsystem returns control to the System Supervisor, which then passes control to the subsystem specified on the card.

The $IBSYS card is used to define the beginning of a job segment to be performed by the System Monitor. When it is recognized by a subsystem, the subsystem returns control to the System Supervisor, which then proceeds to read and process control cards.

A $STOP card defines the end of stack of jobs. When it is recognized by the System Supervisor or a subsystem, the System Supervisor causes an end-of-jobs sequence to occur.

A job segment to be performed by a subsystem consists of all cards beginning with a $EXECUTE card and ending with, but not including, the next $EXECUTE card, $IBSYS card, $JOB card, or $STOP card. The job segment may consist of one or more applications of the particular subsystem specified on the $EXECUTE card. Any cards in the job segment that follow the $EXECUTE card are read and interpreted by the subsystem specified on the $EXECUTE card. They, therefore, must conform to the requirements of the specific subsystem. The publications describing the requirements for each subsystem are listed in Figure 2.

A job segment to be performed by the System Monitor consists of all cards beginning with a $IBSYS card and ending with, but not including, the next $EXECUTE card, $JOB card, or $STOP card.

Control Card Format

The general format of the System Supervisor control cards follows:

Subsystem Name Specified on $EXECUTE Card	Full Name of Subsystem	Publications Describing Subsystems
IBJOB	IBJOB Processor	IBM 7090/7094 IBSYS Operating System: IBJOB Processor, Form C28-6275
FORTRAN	FORTRAN II Processor (FORTRAN Mode)	IBM 7090/7094 Programming Systems: FORTRAN II Operations, Form C28-6066
IBSFAP	FORTRAN II Processor (IBSFAP Mode)	IBM 7090/7094 Programming Systems: FORTRAN Assembly Program (FAP), Form C28-6235
IOCS	Input/Output Control System	IBM 7090/7094 IBSYS Operating System: Input/Output Control System, Form C28-6345
RESTART	Restart Program	IBM 7090/7094 IBSYS Operating System: Operator's Guide, Form C28-6355
CT	Commercial Translator Processor	IBM 7090/7094 Commercial Translator Processor, Form J28-6169
SORT	Generalized Sorting System	IBM 7090/7094 Generalized Sorting System: 7090/7094 Sort, Form C28-6307
9PAC	9PAC	IBM 7090 Programming Systems, Share 7090 9PAC Part 1: Introduction and General Principals, Form J28-6166 Part 2: The File Processor, Form J28-6167 Part 3: The Reports Generator, Form J28-6168
DK90UT	Utilities	IBM 7090/7094 IBSYS Operating System: Utilities, Form C28-6364 IBM 7090/7094 Utilities Routines for IBM 1301 Disk Storage, Form J28-6223

Figure 2. Subsystems Operating Under Control of the System Monitor

COLUMNS	CONTENTS
1	$
2-8	Control card name left-justified
16-72	Variable field information (argument 1, argument 2,... ,argument n)

Columns 7 and 8 are not examined by the System Supervisor.

Embedded blanks are not allowed within arguments, except for the $DATE card. A comma separates arguments and a blank separates the last argument from comments.

In this publication, the following conventions are used for variable field information:

1. Lower-case letters indicate that a substitution must be made.

2. Upper-case letters must be punched exactly as shown.

3. Brackets [ ] contain an option that may be omitted or included by the user.

4. Braces { } indicate that a choice of the contents is mandatory.

5. A number over the first character of a field indicates the first card column of the field.

Basic Control Cards

In a typical installation, many jobs are relatively small and require the use of only one or two subsystems. For these jobs, only the following System Supervisor control cards are normally required:

$JOB
$EXECUTE
$ID
$*
$PAUSE

Of these cards, the $ID, $*, and $PAUSE cards are optional. They may or may not be used, depending on the job and the regulations in force at a particular installation. The $JOB, $EXECUTE, and $ID cards are recognized by each of the subsystems and the System Editor as well as by the System Supervisor. Detailed descriptions of the $JOB, $EXECUTE, $ID, $*, and $PAUSE cards follow.

$JOB Card

Format:

1	16
$JOB	any text

This card defines the beginning of a job. It causes a transfer to the installation accounting routine (if one exists at the installation) and the restoration of any units that were reassigned or made unavailable during a previous job, with the exception of the following:

1. Any unit that was logically detached from control by the Operating System.

2. Any unit that was assigned to a system unit function in place of a detached unit.

3. Any suit that was assigned to a system input, system output, or system peripheral punch function.

When a $JOB card is read by a subsystem, the System Supervisor is called into core storage only if it is required either to restore the status of a unit or to control a manually initiated between-jobs interrupt condition.

The $JOB card is listed on both the System Printer and the System Output Unit. Columns 16 through 72 are normally used to identify a job and may contain any combination of alphameric characters and blanks.

$EXECUTE Card

Format:

1	16
$EXECUTE	subsystem name

This card defines the beginning of a job segment that is to be processed by the specified subsystem. If the $EXECUTE card is read by the System Supervisor, the System Supervisor positions the proper System Library Unit to the specified subsystem, reads in the first record of the subsystem, and relinquishes control to it. If the card is read by a subsystem other than the one specified, control, as well as the subsystem name, is passed to the System Supervisor, which in turn, reads in the first record of the specified subsystem and relinquishes control to it. If the card is read by the specified subsystem, the subsystem retains control and proceeds to process the job segment.

The subsystem name consists of six or fewer BCD characters corresponding to a name in the System Name Table of the System Supervisor. The System Name Table indicates the arrangement of the participating subsystems on the System Library Units and is used by the System Supervisor to locate a subsystem specified by a $EXECUTE card.

The names for the subsystems provided with the distributed versions of the IBSYS Operating Systems are listed in Figure 2, together with the full name of each subsystem and the publication or publications describing the subsystem and any control cards that are required in applying it.

$ID Card

Format:

1	7
$ID	any text

This card is used for intrajob accounting purposes at installations that employ an installation accounting routine. The card causes a transfer of control to the installation accounting routine if one exists. Upon completion of the accounting routine, the next card in the system input file is read and processed. Columns 7 through 72 of the card may contain any combination of alphameric characters and blanks.

The distributed version of the Operating System does not contain an installation accounting routine. Therefore, no action occurs when the card is read other than the listing of the card on the System Printer and the System Output Unit.

The exact use and placement of the $ID card will depend upon the accounting practices and regulations in force at a particular installation.

$* Card

Format:

1	3
$*	any text

This card is listed on the System Printer and System Output Unit. No further action occurs. Columns 3 through 72 of the card may contain any combination of alphameric characters and blanks.

$PAUSE Card

Format:

1	16
$PAUSE	instructions to operator

This card causes the machine to stop after the card and the following message is listed on the System Printer:

OPER. ACTION PAUSE

When START is pressed, the processing of cards on the system input file is resumed. Columns 16 through 72 of the card may contain any combination of alphameric characters and blanks.

This card provides the programmer with a means of temporarily interrupting processing to enable the operator to perform a specific task, such as file protecting a reel of tape. Therefore, when the card is used, it should contain an explicit message to the operator so that processing can continue without further delay.

Typical Input Deck

Figure 3 shows a composite system input deck containing several jobs.

Control Cards Used Primarily by the Experienced Programmer and the Operator

The control cards described in this section are mainly of interest to the experienced programmer and the

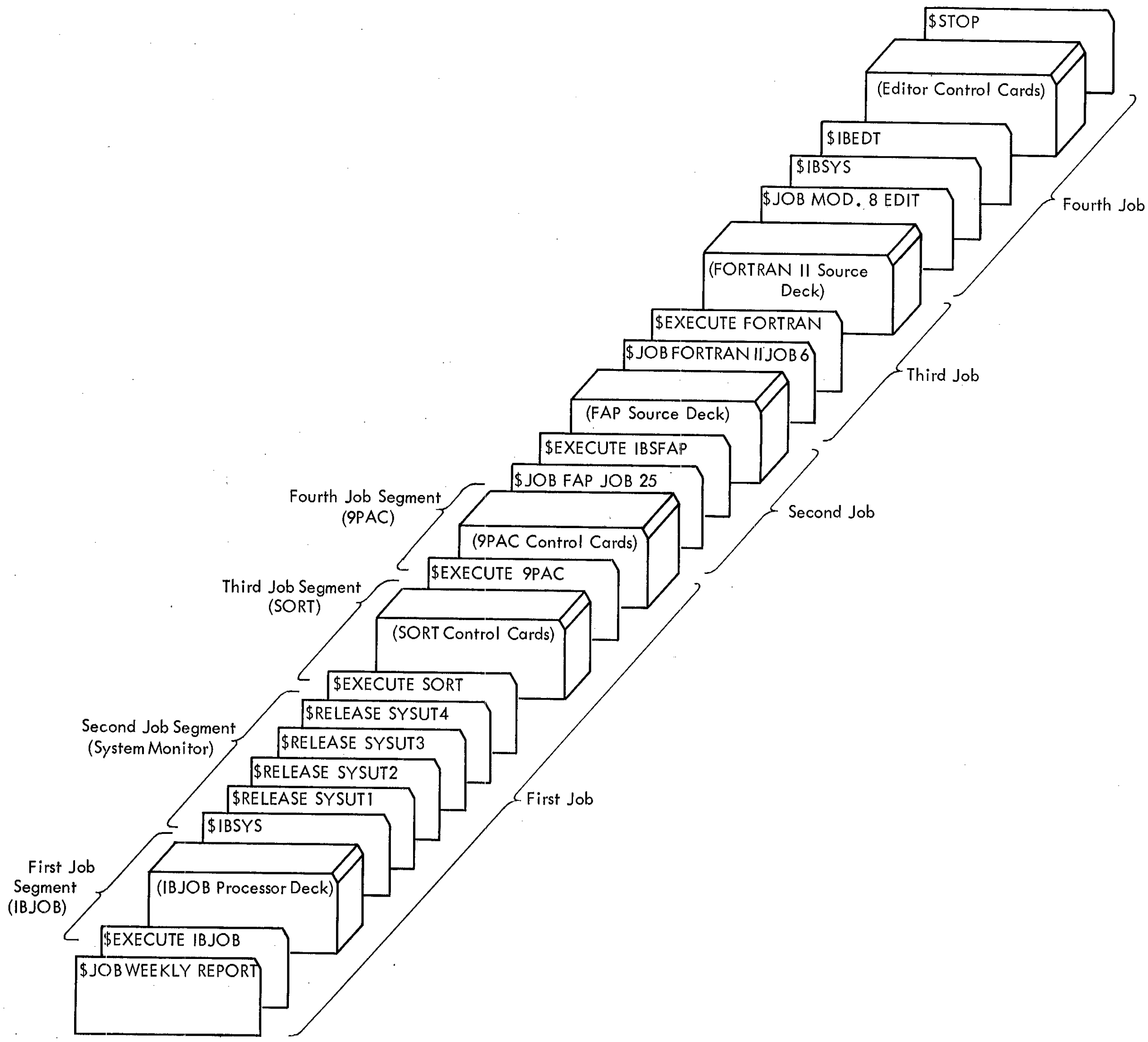


Figure 3. Composite System Input Deck

operator. Except for the $STOP card and the $IBSYS card, these control cards are recognized by the System Supervisor only and not by a subsystem. Therefore, they should be used only under the following conditions:

1. At the beginning of an input file following an initial start.

2. Following a $IBSYS card or other System Supervisor control cards, other than a $IBEDT card, that follow an $IBSYS card.

3. In the System Card Reader during a between-jobs or end-of-jobs interrupt.

Miscellaneous Control Cards

$STOP Card

Format:

```
1              16
$STOP
```

This card is used to define the end of a stack of jobs. It is normally placed, by the operator or a job setup man, at the end of a stack of jobs on the system input file. When the card is read by a subsystem, the System Supervisor is called into core storage and alerted to the fact that the card was read. Upon recognition of the $STOP card, the System Supervisor performs the following actions:

1. Prints, on the System Printer, the physical unit assignment and tape position (record and file count) of the System Output Unit, the System Peripheral Punch Unit, and the System Input Unit followed by the message:

END OF JOBS

2. Writes a trailer label on the System Peripheral Punch Unit, if it is not at load point, and backspaces over the trailer label.

3. If a $REWIND SYSOU1, $REMOVE SYSOU1, or another $STOP card was not read previously*, writes a trailer label on the System Output Unit and backspaces over the trailer label.

4. Stops the machine with the System Card Reader temporarily assigned to the system input function.

When START is pressed, the control cards, if any, in the card reader are read and processed by the System Supervisor until they are depleted. When the cards in the card reader are depleted, the System Supervisor proceeds to read control cards from the unit assigned as the System Input Unit. Therefore, when the machine stops at the end of a stack of jobs, the operator may terminate job processing by using any of the System Supervisor control cards, such as the $ENDFILE and $REMOVE cards, or he may continue processing a new stack of jobs, either on the card reader or on the tape unit assigned as the System Input Unit.

$IBSYS Card

Format:

```
1              16
$IBSYS
```

When this card is read by a subsystem or by the System editor, the System Supervisor is called into core storage and control is relinquished to it. The System Supervisor then reads and processes succeeding control cards until control is relinquished to a subsystem by means of a $EXECUTE card or to the System Editor by means of a $IBEDT card. The System Supervisor must be in control to process any of the control cards described in the remainder of this section.

$DATE Card

Format:

```
1              16
$DATE          mmddyy
```

*After a $REWIND SYSOU1, $REMOVE SYSOU1, or $STOP card is read and processed, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

This card is normally used by the operator at the beginning of each day. The card causes the six characters in columns 16 through 22 of the card to be stored in the SYSDAT word of the Communication Region of the System Nucleus. Although a subsystem may display or otherwise use the SYSDAT word, it should not be modified by the subsystem. If an installation has an interval timer that produces the current date, the date should be stored in the form specified for the $DATE card, where:

mm = Month (01 to 12)
dd = Day (01 to 31)
yy = Year (63 to 99)

$RESET Card

Format:

```
1              16
$RESET
```

This card causes the assignment of a unit to any system unit function, which currently has no unit assigned, but had a unit assigned at initial start. The original unit is reassigned to the function if the unit is not reserved or detached. If it is reserved or detached, a unit from a unit availability chain in the System Nucleus is assigned.

The $RESET card may be used following a $IBSYS card and preceding a $EXECUTE card to ensure that all system utility functions have units assigned before the start of a new job segment.

$RESTORE Card

Format:

```
1              16
$RESTORE
```

This card causes the restoration of the System Supervisor and the regeneration of the System Nucleus as specified by assembly parameters.

The $RESTORE card causes the System Monitor to be called into core storage from the System Library Tape or Disk, giving the same effect as an initial start, except that the tape positions and the SYSDAT word in the Nucleus are not disturbed. The effect of all previous control cards is canceled, except that the $RESTORE card does not effect the source of input as specified by $CARDS or $TAPE cards. However, the unit assigned as the System Card Reader or the System Input Unit may change as a result of the $RESTORE card because of a different unit having been assigned previously by $ATTACH, $AS, or $SWITCH cards.

$LIST Card

Format:

```
1              16
$LIST
```

This card causes all control cards to be listed on the System Printer as well as on the System Output Unit.

Normally, all cards are listed on the System Output Unit and only the $JOB, $ID, $*, $PAUSE, $RESTART, $STOP, $CARDS, $TAPE, $LIST, and $UNLIST cards are listed on the System Printer.

$UNLIST Card

Format:

```
1              16
$UNLIST
```

This card negates the effect of the $LIST card by causing only the $JOB, $ID, $*, $PAUSE, $RESTART, $STOP, $CARDS, $TAPE, $LIST, and $UNLIST cards to be listed on the System Printer. The normal mode is UNLIST, unless the control cards are read by the System Card Reader.

$RESTART Card

Format:

```
1              16
$RESTART       [{±n   }]
                {MATCH}
```

This card is used in the System Card Reader to restart at the beginning of a particular job on the System Input Tape. It may be used by the operator in performing an initial start procedure, a between-jobs interrupt procedure, an end-of-jobs procedure, or a procedure following the detection of an invalid System Monitor control card. The exact use of the $RESTART card in performing each of these procedures is described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

If the variable field of the $RESTART card is +n, the System Supervisor will restart at the beginning of the nth job following the last completed or last partially completed job. For example, if n is 1 and the card is used during an interruption between jobs on the System Input Tape, the System Supervisor restarts at the beginning of the next job on the input tape as though no $RESTART card was processed. If the variable field is −n, the System Supervisor will restart at the nth job preceding the last completed or partially completed job. The value n may range from 0 to 9999. If n is 0 or blank, the System Supervisor will restart at the beginning of the last completed or partially completed job.

When the word MATCH is specified in the variable field of the $RESTART card, the card should be followed by a $JOB card corresponding to a $JOB card on the System Input Tape. The System Supervisor will read the $JOB card following the $RESTART MATCH card, rewind the System Input Tape, search the input file for a job with a matching $JOB card and, if found, restart at the beginning of the job. If a $STOP card is encountered before a matching job card, the System Input Tape will be repositioned to the end of the last completed job and the restart request will be ignored.

NOTE: Only nonblank characters in the variable field will be compared when matching job cards.

$UNITS Card

Format:

```
1              16
$UNITS
```

This card causes all systems unit function names, physical unit assignments, and assigned densities to be listed on the System Output Unit. If a disk is assigned to a system unit function, the HA2 home address identifier and the cylinder limits for the function are included.

This information is also printed on the System Printer if the $UNITS card was read from the System Card Reader or if a $LIST card was read and a subsequent $UNLIST card was not read.

The printout caused by the $UNITS card may be used to verify all unit assignment operations.

$IBEDT Card

Format:

```
1              16
$IBEDT
```

Upon recognizing this card, the System Supervisor calls the System Editor into core storage from a System Library Unit and relinquishes control to it. The control cards that are recognized by the System Editor are described in the section "System Editor."

Unit Assignment Control Cards

The purpose of the unit assignment control cards is twofold. First, they provide a means whereby an installation may indicate changes in machine input/output capabilities to the System Monitor and the subsystems under its control. Second, they provide a means for changing input/output unit assignments within a job.

The unit assignment control cards fall into two categories: those which define the physical avaliability (attachment or detachment) of an input/output unit and those which reassign input/output units to logical system unit functions. Input/output units are initially assigned by assembly parameters when the IBSYS Operating System is assembled. The unit assignment control cards are normally used only for the temporary reassignment of units.

Unit Designation

Physical input/output units and logical system unit functions are designated on the unit assignment control cards as described below.

729 Magnetic Tape Units: A 729 tape unit is designated as xk, where x is the channel (A through H) and k is the tape unit number (0 through 9) on that channel.

Card and Printer Units: The card and printer units are designated as RDX, PUX, and PRX, where RD, PU, and PR are the card reader, card punch, and printer, respectively, and x is the channel (A through H).

Disk Storage Units: A disk storage unit is designated as xDam/s, where x is the channel (A through H), D designates disk, a is the access arm (0), m is the module (0 through 9), and s is the setting of the Data Channel Switch (0 for switch setting 1; 1 for switch setting 2). If the /s is missing from the unit designation, switch setting 1 is assumed.

DRUM

7340 Hypertape Drives: A 7340 Hypertape Drive is designated as xHk/s, where x is the channel (A through H), H designates Hypertape, k is the drive number (0 through 9), and s is the setting of the Data Channel Switch (0 for switch setting 1; 1 for switch setting 2). If the /s is missing from the unit designation, switch setting 1 is assumed.

System Unit Functions: A system unit function is designated as SYSxxx or SYSyyy, where SYSxxx or SYSyyy is the symbolic name for one of the system unit functions listed in Figure 4.

Symbolic Name	Function
SYSLB1	Library 1
SYSLB2	Library 2
SYSLB3	Library 3
SYSLB4	Library 4
SYSCRD	Card Reader
SYSPRT	Printer
SYSPCH	Card Punch
SYSOU1	Output
SYSOU2	Alternate Output
SYSIN1	Input
SYSIN2	Alternate Input
SYSPP1	Peripheral Punch
SYSPP2	Alternate Peripheral Punch
SYSCK1	Checkpoint
SYSCK2	Alternate Checkpoint
SYSUT1	Utility 1
SYSUT2	Utility 2
SYSUT3	Utility 3
SYSUT4	Utility 4

Figure 4. Symbolic Names of System Unit Functions

$DETACH Card

Format:

```
1           16
$DETACH     Unit
```

This card causes the specified unit to be detached. If the unit is assigned to a system unit function, the assignment is cancelled. The unit will remain unavailable and unassigned until it is made available by a $ATTACH card, or until its status is restored by a $RESTORE card or an initial start. The $DETACH card may be used to temporarily deactivate a unit for maintenance purpose.

$ATTACH Card

Format:

```
1           16
$ATTACH     unit [,II]
```

This card causes the specified unit to become attached. The unit attached by this card can be assigned to a system unit function by the next $AS card.

If the specified unit is a Model II or Model V 729 Magnetic Tape Unit, this must be specified by II in the variable field of the card. If II does not appear in the variable field when a 729 Magnetic Tape Unit is specified, the System Supervisor assumes that the unit is a Model IV or VI.

The $ATTACH and $DETACH cards may be used to alert the System Monitor to a physical change in the status of a unit. For example, if, at an installation, the sixth unit on channel D were physically disconnected and then reconnected as the fifth unit on channel C, this change in status might be indicated to the System Monitor by the following cards:

```
1           16
$ATTACH     C5
$DETACH     D6
```

$AS Card

Format:

```
1           16
$AS         SYSxxx
             ,H
             ,L
```

This card causes the unit specified on the last recognized $ATTACH card to be assigned to the specified system unit function. If the tape density is specified by H, HI, or HIGH, the density for the system unit function is set to high. It is set to low if the specifications is L, LO, or LOW. If the density specification is absent, the density will be set according to the assembly parameter HIGHLO. With the distributed System Library Tape, low density is assumed if the density is not specified on the card.

This card may apply to a unit already attached. In this case, there is no need to detach the unit before reattaching it. The $AS card releases the unit that was assigned to the system unit function before the $AS card was recognized. If there are two or more $AS cards in succession, the unit specified on the last recognized $ATTACH card is assigned to the function specified on each $AS card.

If the last attached unit was a disk storage unit, the $AS card is expanded as follows:

```
1           16
$AS         SYSxxx,nnn,ccc,hh
```

Here, nnn and ccc must be replaced by three-digit numbers denoting the number of cylinders (nnn) and the starting cylinder (ccc) defined for the System Unit Function Table entry. A two-character symbol, representing the HA2 home address identifier, should be placed in the next field (hh). Any blanks or zeros in this field will be converted to octal 12s.

For example, to assign 25 cylinders [starting with cylinder 125 (load point) on the access arm in module 0 on channel E, switch setting 2, with a HA2 of PQ] as System Utility Unit 3 (SYSUT3), the following control cards would be used:

```
1              16
$ATTACH        ED00/1
$AS            SYSUT3,025,125,PQ
```

$RELEASE Card

Format:

```
1              16
$RELEASE       SYSxxx
```

This card causes the unit assigned to the specified system unit function to be released from the function. If the unit was concurrently assigned to other system unit functions, it remains assigned to those functions.

$SWITCH Card

Format:

```
1              16
$SWITCH        SYSxxx,SYSyyy
```

This card causes the units assigned to the two specified system unit functions to be transposed; that is, the unit assigned to SYSXXX is assigned to SYSYYY, and the unit that was assigned to SYSYYY is assigned to SYSXXX. Physical density settings remain the same, and, after the transposition, the transposed units accept the established density settings.

$CARDS Card

Format:

```
1              16
$CARDS
```

This card causes the System Supervisor to read succeeding control cards from the unit assigned as the System Card Reader (SYSCRD).

$TAPE Card

Format:

```
1              16
$TAPE
```

This card causes the System Supervisor to read succeeding control cards from the unit assigned as the System Input Unit (SYSIN1).

Tape Manipulation Control Cards

The tape manipulation control cards provide the operator and programmer with facilities for the automatic manipulating of tape units assigned to system unit functions. If no unit is assigned to the system unit function specified on a tape manipulation control card, or if a card or printer unit is assigned to the function, the card has no effect. The $ENDFILE, $REWIND, and $REMOVE cards have the same effect on a 7340 Hypertape Drive and a 729 Magnetic Tape Unit.

$ENDFILE Card

Format:

```
1              16
$ENDFILE       SYSxxx
```

This card causes the tape unit assigned to the specified system unit functions to write, on the tape, an end-of-file gap followed by a tape mark. No test is made to determine if the operation is invalid, such as writing a tape mark on the System Input Unit.

$REWIND Card

Format:

```
1              16
$REWIND        SYSxxx
```

This card causes a tape unit assigned to the specified system unit function to be rewound. If the specified function is SYSOU1, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

$REMOVE Card

Format:

```
1              16
$REMOVE        SYSxxx
```

This card causes a tape unit assigned to the specified system unit function to be rewound and unloaded. If the specified function is SYSOU1, further use of the System Output Unit by the System Supervisor is suspended until a $JOB card is read or the System Supervisor is called into core storage again by a $IBSYS card or by a subsystem.

$UNLOAD Card

Format:

```
1              16
$UNLOAD        SYSxxx
```

This card causes the Hypertape assigned to the specified system unit function to be unloaded without rewinding. If a 729 tape unit is assigned to the specified system unit function, the $UNLOAD card will be interpreted as a $REMOVE card.

$PROTECT Card

Format:

```
1              16
$PROTECT       SYSxxx
```

This card causes the Hypertape assigned to the specified system unit function to be file-protected. The $PROTECT card is ignored if a Hypertape drive is not assigned to the specified system unit function.

System Core-Storage Dump Program

General Description

The System Core-Storage Dump Program is designed to facilitate the testing of programs under System Monitor control. To perform this function, snap dump and post-mortem dump options have been provided. The snap dump option of the Core-Storage Dump Program can dump and edit one or more sequential locations of core storage during the execution of an object program. After the dump is completed, core storage is returned to its original condition and control is returned to that point in the object program from which the dump was called. The post-mortem dump option of the Core-Storage Dump Program dumps core storage in the same manner as the snap dump option, but at its completion control is returned to the System Supervisor which then skips cards on the System Input File until a $STOP card or the next job or job segment is encountered. It then begins to process control cards. A job always begins with a $JOB card, and a job segment always begins with a $EXECUTE or $IBSYS card.

The format, the limits, and the output units for both the snap and post-mortem forms of the core storage dump may be either assembly defined or specified by a control word*. For the post-mortem dump, the additional option of defining these parameters with the console entry keys is provided.

When a dump is requested, the System Nucleus writes a portion of core storage onto the alternate System Peripheral Punch Unit (SYSPP2), reads in the Core-Storage Dump Program, and transfers control to it. If the console entry key option is requested, a halt occurs to allow the insertion of dump parameters. The Core-Storage Dump Program dumps the edited output onto the System Output Unit, onto the System Printer, or onto both.

Using the System Core-Storage Dump Program

Transfer to Dump Instructions

To obtain a dump of core storage during the execution of an object program, insert one of the following instructions in the body of the source program at the point at which the dump is required:

TRA SYSDMP: This instruction provides a post-mortem dump of all of core storage in an assembly defined format.

TRA SYSDMP with Sense Switch 4 Down: This instruction results in a post-mortem dump, in accordance with the parameters entered by means of the console entry keys. Before the dump is executed, a halt occurs and a message requesting insertion of dump parameters is printed on the System Printer.

TSX SYSDMP, 4, 1 Followed by a Parameter Control Word: This sequence results in a post-mortem dump, in accordance with the information from the parameter control word. The parameter control word is described in the section "Dump Parameters."

TSX SYSDMP, 4, 0 Followed by a Parameter Control Word: This sequence results in a snap dump, in accordance with the information from the parameter control word. For example, the following instruction and control word would result in an octal snap dump on the System Output Tape of storage locations beginning at STDMP and ending at ENDMP.

```
8         16
TSX       SYSDMP,4,0
PON       ENDMP,0,STDMP
```

Dump Parameters

The dump parameters are entered either by a parameter control word (Figure 5) for a snap dump or by a parameter control word or the console entry keys for a post-mortem dump. Any one of six dump formats (Figure 6) can be specified. In the distributed version of the Core-Storage Dump Program, output is single spaced. The various parts of the parameter control word are interpreted as follows:

Prefix	PON	1 FORMAT A – Octal, eight words per line.
	PTW	2 FORMAT B – BCD, sixteen words per line.
	PTH	3 FORMAT C – SQUEZY, Mnemonics with address and tag field. If the Core-Storage Dump Program cannot interpret the operation code, the octal representation is given.
	MZE	4 FORMAT D – Octal and SQUEZY. If the SQUEZY word would normally have appeared in octal form, it is not listed twice, but is suppressed. Otherwise, both the octal word and the SQUEZY word are listed.

*If no parameters are specified when a dump is requested, the dump will be accomplished using parameters within the Core-Storage Dump Program. These parameters are established when the Core-Storage Dump Program is assembled. (See the section "System Library Preparation and Maintenance.")

	MON	5 FORMAT E – Octal and mnemonics.
	MTW	6 FORMAT F – Octal, mnemonics, and BCD. A BCD interpretation of the word is listed to the right of the mnemonic.
Address	The ending location of the dump.	
Tag	0 = Dump onto System Output Unit. 1 = Dump onto System Printer. 2 = Dump onto both System Output Unit and System Printer.	
Decrement	The starting location of the dump.	

Prefix	Decrement	Tag	Address
Format Code	Starting Location	Output Code	Ending Location
S——2	3——17	18——20	21——35

Figure 5. Parameter Control Word Format

NOTES: The limits of the requested dump may be stated in any order, i.e., the starting parameter in the address field and the ending parameter in the decrement field, or vice versa.

In the distributed version of the Core-Storage Dump Program, the dump is always made onto the System Output Unit only, regardless of the contents of the tag position of the parameter control word.

A parameter control word of all zeros will provide a full core storage dump in the assembly defined format. In the distributed version, format 3(C) in Figure 6 is used.

Machine Status at the End of a Core Storage Dump

If any input/output operation had been in progress when the snap dump routine was called, the traps resulting from this operation are lost if they occurred on the channels used by the Core-Storage Dump Program. Since the System Loader disables all traps when loading the Core-Storage Dump Program, and the dump resets the work tape data channel when restoring core storage, it is recommended that all input/output operations be terminated before calling in the Core-Storage Dump Program.

At the completion of a snap dump, all of core storage, except locations SYSEND-18 through SYSEND, are restored. These last 19 locations are destroyed by the restore portion of the snap dump routine. Because of space restrictions, no error checking of the work tape is done while restoring core storage. The snap dump routine also repositions the System Library Tape to its position prior to the core storage dump as part of its restore process.

1 Format A -- Octal

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
000000000000  000000000000  001321000000  000000000000   00001    00006    74320
                                                         -77777   -77772   -03460
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000  100000000000  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
00030  000000000031  000060000030  000000000033  000060000032  001321000000  000060000034  001321000000  001321000000
```

2 Format B -- BCD

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000   00001    77323    71371
                                                         -77777   -00455   -06407
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000 DK9OUT 800000 4+003* 0+00** 0A00*F 000000 000000 0A00*H 0A00** 000000 000000 0A007R 000000 0A007R 000000 0A007R
00020 000000 0A007R 000000 0A007R 00000E 00 00D 00000G 00 00F 00000I 00 00H 00000. 00 00+ 801H4* 00 00) 801H4* 801H4*
00040 000000 000000 000000 80008U 000000 80108U 000000 80208U 000000 80308U 000000 80408U 000000 80508U 000000 80608U
```

3 Format C -- SQUEZY

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000700   00001    77323    71371
                                                         -77777   -00455   -06407
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000  244211006463  100000000000  HPR     377   TRA    5557    TTR    3626    HTR           HTR            TTR    3630
00010  TTR    3632   HTR           HTR           TTR     751    HTR            TTR    751    HTR            TTR     751
00020  HTR           TTR     751   HTR           TTR     751    HTR      25    HTR *   24    HTR     27     HTR *    26
```

4 Format D -- Octal and SQUEZY

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000   00001    77323    71371
                                                         -77777   -00455   -06407
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
                                   HPR       377  TRA     5557  TTR     3626  HTR           HTR           TTR     3630
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR    3632   HTR           HTR           TTR      751   HTR           TTR      751  HTR           TTR      751
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR           TTR      751  HTR           TTR      751   HTR       25  HTR *     24  HTR       27  HTR *     26
```

5 Format E -- Octal and Mnemonics

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000   00001    77323    71371
                                                         -77777   -00455   -06407
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
       TIX           TXI           HPR           TRA           TTR           HTR           HTR           TTR
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR           HTR           HTR           TTR           HTR           TTR           HTR           TTR
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR           TTR           HTR           TTR           HTR           HTR *         HTR           HTR *
00030  000000000031  000060000030  000000000033  000060000032  101201300475  000060000034  101201300475  101201300475
       HTR           HTR *         HTR           HTR *         TXI           HTR *         TXI           TXI
```

6 Format F -- Octal, Mnemonics, and BCD

```
       AC              MQ          SENSE IND       KEYS          XR1      XR2      XR4
-300000005176  000000000000  101201300475  000000000000   00001    77323    71371
                                                         -77777   -00455   -06407
         INDICATORS                          SENSE LIGHTS                            SENSE SWITCHES
Q-BIT P-BIT  TRAP   DCT    IOT    OFL          1      2      3      4              1     2      3      4      5      6
 OFF   OFF   OFF    OFF    OFF     ON         OFF    OFF    OFF    OFF            ON    OFF    OFF     ON    OFF     ON
00000  244211006463  100000000000  042000000377  002000005557  002100003626  000000000000  000000000000  002100003630
       TIX   DK9OUT  TXI   800000  HPR   4+003*  TRA   0+00**  TTR   0A00*F  HTR   000000  HTR   000000  TTR   0A00*H
00010  002100003632  000000000000  000000000000  002100000751  000000000000  002100000751  000000000000  002100000751
       TTR   0A00*+  HTR   000000  HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R
00020  000000000000  002100000751  000000000000  002100000751  000000000025  000060000024  000000000027  000060000026
       HTR   000000  TTR   0A007R  HTR   000000  TTR   0A007R  HTR   00000E  HTR * 00 00D  HTR   00000G  HTR * 00 00F
```

Figure 6. Core-Storage Dump Formats

Function and Organization

The System Nucleus is a portion of the System Monitor which remains in core storage at all times and provides common facilities for intercommunication and control between the System Monitor and the subsystems operating under it. The Input/Output Executor and the unit control blocks for the input/output units are described separately in the section "Input/Output Executor" although they may be considered part of the System Nucleus in that they normally remain in core storage at all times and provide common facilities for the System Monitor and the subsystems.

The System Nucleus consists of the following:

1. A communication region containing constants, control words, and transfer words that are required for intercommunication between the System Monitor and the subsystems.
2. The System Unit Function (SYSUNI) Table, which is used to keep account of the units assigned to system unit functions.
3. The Disk Limits Table, which, when disk storage is employed, supplements the function of the System Unit Function Table by defining the portion of disk storage assigned to each system unit function.
4. The Unit Control Block Table, which consists of a word for each channel containing the address of the first unit control block for the channel, the total number of input/output units assigned to the channel, and the number of card units assigned to the channel.
5. The Unit Availability Table, which consists of one word for each channel that serves as an entry point to a chain of available units on the channel.
6. The System Loader, which is used by the System Monitor and the subsystems for scatter-loading records from the System Library Unit.
7. An interrupt routine, which is used by the System Monitor and the subsystems to interrupt processing between jobs for the purpose of manual intervention.
8. A dump-calling routine, which is a bootstrap routine for loading the Core-Storage Dump Program whenever a dump is called for by the System Monitor, a subsystem, an object program, or manually by the operator.

Of the various portions of the System Nucleus, the most important, so far as the systems programmer is concerned, are the Communication Region, the System Unit Function Table, the Disk Limits Table, and the Unit Availability Table. Therefore, each of these is described below, together with job control communication requirements for subsystems operating under the System Monitor.

Communication Region

The Communication Region of the System Nucleus consists of 28 consecutive core storage locations containing various constants, control words, and transfer words that may be referred to by the System Monitor or by any of the subsystems operating under it. The function of each entry in the Communication Region, together with its octal absolute address and its symbolic address, is given in Figure 7. A more complete description of the function of each entry is given in Appendix A.

Octal Address	Symbolic Address	Function
100	SYSTRA	Transfer instruction to current subsystem
101	SYSDAT	Date Word
102	SYSCUR	Name of current subsystem
103	SYSRET	Location to which each subsystem returns
104	SYSKEY	Contents of entry keys at initial start
105	SYSSWS	Contents of sense switches at initial start
106	SYSPOS	Initial position and index of current subsystem
107	SYSUNI	Location and length of System Unit Function Table
110	SYSUBC	Location and length of table of unit control block locators by channel
111	SYSUAV	Location and length of the Unit Availability Table
112	SYSUCW	Location and length of all the unit control blocks
113	SYSRPT	Transfer to between-jobs interrupt routine
114	SYSCEM	Transfer to customer engineering diagnostic routine
115	SYSDMP	Transfer to bootstrap for core storage dump
116	SYSIOX	Location and length of IOEX communication Table
117	SYSIDR	Transfer to installation accounting routine, if any.
120	SYSCOR	Lower limit of usable core storage in decrement, upper limit in address
121	SYSLDR	Transfer to system scatter-load routine
122	SYSACC	Installation accounting routine Communication
123	SYSPID	Installation accounting routine Communication
124	SYSCYD	Channel commands for system to copy and disconnect
125	SYSCYD+1	
126	SYSSLD	Self-loading sequence
127	SYSTCH	Self-loading sequence
130	SYSTCH+1	
131	SYSTWT	System trap, wait, and transfer point
132	SYSGET	Subsystem communication with System Supervisor
133	SYSJOB	Job control word

Figure 7. Communication Region of System Nucleus

The entries in the Communication Region may be referred to by their absolute addresses, since their location will not change and is not dependent on the input/output configuration of the Operating System.

In addition, the entries in this region may be referred to by their symbolic addresses when using the FORTRAN II Assembly Program (FAP) if an SST (Save Symbol Table) pseudo-operation is included in the first card group of the FAP language source program. These entries may also be referred to by their symbolic addresses when using the Macro Assembly Program (MAP).

System Unit Function Table

The System Unit Function Table consists of 19 entries, one for each of the 19 possible system unit functions. The octal absolute address, the symbol, and the system unit function for each entry is given in Figure 8. When programming in FORTRAN II Assembly Program (FAP) language, and an SST (Save Symbol Table) pseudo-operation is included in the first card group of the FAP source program, the symbols listed in Figure 8 represent the symbolic addresses of the table entries. For example SYSLB1 represents 140_8 and SYSUT4 represents 162_8. However, when programming in Macro Assembly Program (MAP) language, the same symbols represent index numbers, beginning with 0 for SYSLB1 and ending with 22_8 for SYSUT4. To obtain the correct address of an entry in the System Unit Function Table when programming in MAP, the index number represented by the symbol must be added to the address portion of the SYSUNI entry in the Communication Region of the Nucleus. Since the address portion of the SYSUNI entry contains the address of the first word of the System Unit Function Table, the addition will result in the correct address. For example, adding the address portion of the SYSUNI entry (140_8) to SYSOU1 (defined as 7 by MAP), results in the address (147_8) of the entry in the System Unit Function Table for the System Output Unit.

Octal Address	Symbol	System Unit Function
140	SYSLB1	Library 1
141	SYSLB2	Library 2
142	SYSLB3	Library 3
143	SYSLB4	Library 4
144	SYSCRD	System Card Reader
145	SYSPRT	System Printer
146	SYSPCH	System Punch
147	SYSOU1	Output
150	SYSOU2	Alternate Output
151	SYSIN1	Input
152	SYSIN2	Alternate Input
153	SYSPP1	Peripheral Punch
154	SYSPP2	Alternate Peripheral Punch
155	SYSCK1	Checkpoint 1
156	SYSCK2	Checkpoint 2
157	SYSUT1	Utility 1
160	SYSUT2	Utility 2
161	SYSUT3	Utility 3
162	SYSUT4	Utility 4

Figure 8. System Unit Functions

The address portion of each entry in the System Unit Function Table contains the address of the first word of the unit control block for the unit assigned to the system unit function. If no unit is assigned to the system unit function, the address portion of the entry contains zeros. Normal assignments of units to system unit functions are specified when the System Monitor is assembled. However, normal unit assignments may be changed temporarily by unit assignment control cards.

Tape density is indicated by an entry in the sign bit position. Low density is specified when the sign is plus (PZE), and high density is specified when the sign is minus (MZE). The System Monitor will only set the densities of the tape units it uses. Each subsystem has the responsibility for setting the densities of the tape units it uses.

When disk storage is assigned to a system unit function, bits 6 through 17 of the entry for the system unit function contain the HA2 home address identifier that was specified on the $AS control card used to assign the disk module to the function. Bits 3 through 5 are zero.

Disk Limits Table

The Disk Limits Table is located, starting at location 163_8, immediately following the System Unit Function Table. It contains 19 entries corresponding to the 19 entries of the System Unit Function Table. When an entry in the System Unit Function Table contains the address of a unit control block for a disk storage module, the corresponding entry in the Disk Limits Table contains a parameter word, in binary form, which designates the first and last tracks of the consecutive tracks of the module that are assigned to the system unit function. The parameter word has the following format:

```
PZE        DORG,,DEND
```

Unit Availability Table

The Unit Availability Table consists of one entry per input/output channel, beginning with channel A. Each entry contains the address of the first word of the unit control block for the first unassigned (available) unit on the channel. The address portion of the first word of the unit control block for each unassigned unit on a channel contains the address of the first word of the unit control block for the next unassigned unit on the channel. In this way, a unit availability chain for each channel is formed, beginning with an entry in the Unit Availability Table. The end of the chain is indicated by zeros in the address portion of the first word of the unit control block for the last unassigned unit on the channel. Whenever a subsystem requires the use of an available unit, it interrogates a unit availability chain by way of a Unit Availability Table entry and removes the unit from the chain.

Job Control Communication with Subsystems

Each of the subsystems operating under the System Monitor must follow certain operation procedures involving communication with the System Monitor. These procedures are required to ensure proper job and job segment control, proper control of between-jobs and end-of-jobs interrupts, and restoration, if necessary, of input/output unit assignments at the beginning of each job.

Restoration of Unit Assignments Between Jobs

Restoration of input/output unit assignments is required at the beginning of a job if, during the previous job, a $IBSYS card was followed by a $AS, $SWITCH, or $RELEASE card or a subsystem (or an object program running under the subsystem) removed or replaced a unit in a unit availability chain. If either of these conditions occurred, the System Supervisor, at the beginning of the next job, begins restoring unit assignments by first chaining all attached units on each channel in ascending order in accordance with the unit or module number. Any unit assigned to a system unit function at initial start is then reassigned by the System Supervisor to the same function, provided the unit is not currently assigned to a system input, system output, or system peripheral punch function or is not currently assigned to a system unit function in place of a detached unit. Any unit assigned to a system unit function is removed from the unit availability chain for its channel unless it is a random access storage module. The restoration of unit assignments at the beginning of a job differs from that initiated by the $RESTORE card in that detached units remain detached and current system input, system output and system peripheral punch assignments are not changed.

Communication Region Locations SYSRET, SYSGET and SYSJOB

Communications between the subsystems and the System Supervisor is carried out chiefly by way of three locations in the Communication Region of the System Nucleus. These three locations are SYSRET, SYSGET and SYSJOB. The function of each of these locations is described below.

Communication Location SYSRET 1038

Whenever a subsystem is required to return control to the System Supervisor, it transfers to the location SYSRET. As a result, the System Supervisor is read into core storage and control is relinquished to it.

Communication Location SYSGET

Before returning control to the System Supervisor, a subsystem must ensure that a word is stored in the SYSGET location which indicates to the System Supervisor the reason why control was returned to it by the subsystem. In addition, whenever a post-mortem dump is performed by the System Core-Storage Dump Program it places a word in SYSGET (to indicate that a post-mortem dump was performed) before it returns control to the System Supervisor. When the System Supervisor obtains control from the dump program or from a subsystem, it examines the word in SYSGET and takes appropriate action.

If the word in the SYSGET location is a subsystem name, it indicates to the System Supervisor that a subsystem read a $EXECUTE card containing a subsystem name other than its own. Therefore, the System Supervisor loads into core storage the first record of the subsystem whose name was in the SYSGET location and relinquishes control to it.

If the word in the SYSGET location is "IBSYST," it indicates to the System Supervisor that a $IBSYS control card was read by a subsystem. Therefore, the System Supervisor begins processing control cards on the input file, beginning with the next card.

If the word in the SYSGET location is "IBSXEC," it indicates to the System Supervisor either of the following:

1. A post-mortem core-storage dump was taken by a subsystem, an object program, or the operator; therefore, a job segment was not completed.
2. A subsystem could not complete a job segment.

In either case, the System Supervisor skips on the system input file until a $IBSYS, $EXECUTE, $JOB, or $STOP control card is encountered and then processes cards, normally beginning with that card.

If the word in the SYSGET location is "IBSNXT," it indicates to the System Supervisor that a subsystem has determined that a job cannot be completed. Therefore, the System Supervisor skips on the system input file until a $JOB or $STOP control card is encountered and then processes cards normally, beginning with that card.

If the word in the SYSGET location is "$STOPb," where "b" is a blank, it indicates to the System Supervisor that a $STOP control card was read by a subsystem. Therefore, the System Supervisor initiates an end-of-jobs sequence as though it had read the $STOP card.

If the word in the SYSGET location is "IBSBSR," it indicates to the System Supervisor that a $JOB card was read by a subsystem and either a between-jobs interrupt condition exists or an input/output unit was reassigned or made unavailable during the previous job. In either case, the System Supervisor proceeds as though it had just read a $JOB card. Refer to the description of the $JOB card in the section "System Supervisor."

Communication Location SYSJOB

The communication location SYSJOB is used by the System Supervisor and the subsystems in controlling the processing of jobs.

The sign bit of the word indicates whether or not any input/output unit assignments need be restored at the beginning of the next job. A minus sign indicates that restoration is necessary and a plus sign indicates that restoration is not necessary. The System Supervisor sets the sign of SYSJOB to minus if it processes a $AS, $SWITCH, or $RELEASE control card. Similarly, each subsystem must set the sign to minus if it is about to change a unit availability chain. The sign should be set to minus before the change is made in the event the job "blows-up" and the subsystem does not regain control. The sign bit is interrogated at the beginning of each new job by the Nucleus routine SYSRPT to determine if restoration of input/output unit assignments is required. It is then set to plus by the System Supervisor before actual processing of the job begins.

Bit 17 of SYSJOB is used by the subsystems to indicate to one another whether or not a previous job segment could not be completed. Whenever a subsystem determines that it cannot complete a job segment, it sets bit 17 of SYSJOB to a 1 and stores the word "IBSXEC" in the SYSGET location. It then returns control to the System Supervisor, which skips to the next job segment.

When a subsystem gains control at the beginning of a job segment, it tests bit 17 of SYSJOB and proceeds normally if it is a 0. However, if it is a 1, it determines whether or not the present job segment should be discontinued as a result of a previous job segment not being completed. If the current job segment should be discontinued, the subsystem stores the word "IBSXEC" in the SYSGET location and returns control to the System Supervisor by way of SYSRET. At the completion of a post-mortem dump, the System Core-Storage Dump Program sets bit 17 to a 1 and stores the word "IBSXEC" in the SYSGET location before returning control to the System Supervisor. The Nucleus routine SYSRPT clears bit 17 to 0 at the beginning of each job.

The address portion of SYSJOB contains a count of the number of jobs processed on the current input file. This count is maintained by the System Supervisor and is used to locate a specified job when the $RESTART ±n control card is processed. The subsystems are in no way concerned with the job count.

Recognition of System Supervisor Control Cards by Subsystems

Each subsystem must recognize and act upon the $IBSYS, $EXECUTE, $STOP, $ID, and $JOB control cards. The action that must be taken by a subsystem when each card is recognized is as follows:

$IBSYS Card

When a $IBSYS card is recognized by a subsystem, it must return control to the System Supervisor. Whenever a subsystem gains control from the System Supervisor, the SYSGET location will contain the word "IBSYST." Therefore, it is not necessary for the subsystem to load the word "IBSYST" into the SYSGET location when a $IBSYS card is recognized.

$EXECUTE Card

When a $EXECUTE card is recognized by a subsystem and the subsystem name on the $EXECUTE card is the name of the subsystem, the subsystem retains control and continues normal processing. Otherwise, the subsystem stores the name specified on the card into the SYSGET location and returns control to the System Supervisor by way of SYSRET. If the subsystem had changed an availability chain, it would have previously set the sign of SYSJOB to minus.

$STOP Card

When a subsystem recognizes a $STOP card, whether or not it is located in the proper sequence on the input file, it loads the word "STOPb" in the SYSGET location and returns control to the System Supervisor by way of SYSRET.

$ID Card

When a subsystem recognizes a $ID card, it must TSX to SYSIDR as follows:

```
        TSX     SYSIDR,4
        PZE     L($ID)
            return
```

where L($ID) is the location of the first word of the buffer containing the $ID card in BCD form.

$JOB Card

When a subsystem recognizes a $JOB card, whether or not it is located in the proper sequence on the input file, it loads the word "IBSESR" into SYSGET and then executes a TSX SYSRPT, 4 instruction. The SYSRPT routine in the System Nucleus will then determine if the System Supervisor must restore unit assignments (the sign of SYSJOB is minus) or control a between-jobs interrupt (sense switch 1 is DOWN). If the System Supervisor must do either, control is passed to the System Supervisor by way of SYSRET. Otherwise, control is returned to the subsystem. When the subsystem regains control, it must restore the word "IBSYST" in the SYSGET location and TSX to SYSIDR as follows:

```
        TSX     SYSIDR
        PZE     L($JOB)
            return
```

where L($JOB) is the location of the first word of the buffer containing the $JOB card in BCD form.

Input/Output Executor

The Input/Output Executor (IOEX) consists of a trap supervisor and a number of utility routines that are used in common by the System Monitor and the subsystems operating under its control. Any subsystem that is incorporated under the System Monitor and employs 729 Magnetic Tape Units, card equipment, 1301 Disk Storage, or 7340 Hypertape Drives should use IOEX to ensure centralized control of input/output activity. The use of a single trap supervisor by the subsystems and the System Monitor not only minimizes input/output coding but also (1) ensures proper coordination of trapping, (2) enables a running log to be kept of tape positions, (3) enables error recovery procedures to be standardized, and (4) simplifies the diagnosis of input/output failures.

A subsystem communicates with IOEX and calls IOEX subroutines by way of a Communication Table located in storage just forward of IOEX. Location SYSIOX in the Communication Region of the System Nucleus (Appendix A) contains the address and length of this table. An entry in the IOEX Communication Table may be referred to by its symbolic address when using either the Macro Assembly Program (MAP) or the FORTRAN II Assembly Program (FAP). However, the symbols used to represent the addresses of the table entries for MAP are different from the symbols used to represent the same addresses for FAP. Both the MAP and FAP symbolic addresses, together with the function of each entry in the IOEX Communication Table, are listed in Figure 16 at the end of this section. When these entries are referred to symbolically using FAP, an SST (Save Symbol Table) pseudo-operation must be included in the first card group of the FAP source program.

The following description of IOEX, together with a symbolic listing of the IOEX portion of the System Monitor, should provide the system programmer with the information required to use IOEX. In both the description and the listing, the FAP symbols are used when referring to entries in the IOEX Communication Table. The equivalent MAP symbols may be obtained by reference to Figure 16.

Unit Control Blocks

Each input/output device that may be referred to by the IBSYS Operating System is represented in IOEX by a four-word unit control block. The unit control blocks are generated by the System Monitor at initial start in accordance with assembly parameters. The following describes the format and contents of the unit control blocks for 729 Magnetic Tape Units, card equipment, 1301 Disk Storage Modules, and 7340 Hypertape Drives.

Unit Control Block for 729 Tape Units and Card Equipment

Each 729 Magnetic Tape Unit and each card unit is represented in IOEX by a four-word unit control block whose format is shown in Figure 9.

	S	1	2	3	4	5——17	18	19	20	21——35
UCW 1	A	M	T	R	C	Unit Address	EOT	DO	DI	Chain Address
UCW 2	S	R				SEL				
UCW 3	N					File Count	Record Count			
UCW 4										

Figure 9. Unit Control Block for 729 Tape Units and Card Equipment

The contents of each unit control block is interpreted as follows:

Word 1

A: Availability Flag
M: Attachment Flag

A=0: The unit is assigned to a particular function, not necessarily a system unit function, and it is not in the availability chain.

- M=0: The assigned function of the unit is such that it should be repositioned when restarting.
- M=1: The unit should not be repositioned when restarting (for example, if the unit is assigned as SYSOU1, SYSOU2, SYSPP1, SYSPP2, SYSCK1, or SYSCK2).

A=1: The unit is not assigned to any particular function.

- M=0: The unit is attached to the channel and is in the availability chain. Card equipment is never in the availability chain.
- M=1: The unit is detached from the channel and cannot be used.

If the user desires to assign a unit in the availability chain to a function not included in the System Unit Function Table, he should set the availability flag to zero, remove the unit from the availability chain, and set the attachment flag to indicate whether or not the unit should be repositioned when restarting.

T: Unit Type (for 729 tapes only)
- T=0: Model II or V
- T=1: Model IV or VI

R: Reserve Status Flag (intersystem use only)
- R=0: The unit is not reserved.
- R=1: The unit is reserved. Address bits 24-35 of word 1 contain data for intersystem pickup. The unit should not be assigned a system unit function or be in the unit availability chain.

C: Channel Type
- C=0: 7607 channel
- C=1: 7909 channel

Unit Address: The address of the input/output unit is contained here. If the unit is a tape, the address is the BCD mode address; e.g., 1201 for 729 tape unit A1.

EOT: End-of-Tape Flag
- EOT=0: No end of tape has occurred on this unit.
- EOT=1: An end of tape has occurred while writing on this unit.

DO: Tape Density at Load Point
- DO=0: Low Density
- DO=1: High Density

DI: Density at the Current Tape Position
- DI=0: Low Density
- DI=1: High Density

Chain Address: This is the address of word 1 of the next unit control block in the availability chain. The chain address of the last unit is zero. This area is available to the user when the unit is not in the availability chain, or is not a reserve unit.

Word 2

S: Select Type
- S=0: Read
- S=1: Write

R: Permanent Redundancy Message (control)
- R=0: A message is printed if a permanent read redundancy occurs.
- R=1: No message is printed in the event of a permanent read redundancy.

SEL: Select Routine
SEL represents symbolically the location of a user's select routine, which initiates data transmission and the posting of completed input/output activity.

Word 3

N: Noise Record Flag (reading only)
- N=0: No noise records have been detected while reading.
- N=1: One or more noise records have been detected while reading.

File Count: The file count reflects the number of file marks written on or read from this tape.

Record Count: The record count reflects the number of records which have been written on or read from the current file.

Word 4

This word is provided for all-purpose systems usage. It is specifically used by IOCS labeling routines for storing the tape reel serial number in case of multifile reels.

Notes

The EOT flag and noise record flag are turned off only when the tape is returned to the rewound position.

The record count is complemented when backspacing from an end of file. A backspace which repositions in front of a file mark gives a record count (18-35) of 777777_8. For example, when writing occurs from such a position, the two low-order tag bits are cleared to prevent a spurious increase in the file count when the record count is increased.

The shaded area in Figure 9 is available to IOCS, or any subsystem using IOEX only.

Disk Unit Control Block

Each 1301 Disk Storage Module is represented in IOEX by a four-word unit control block whose format is shown in Figure 10.

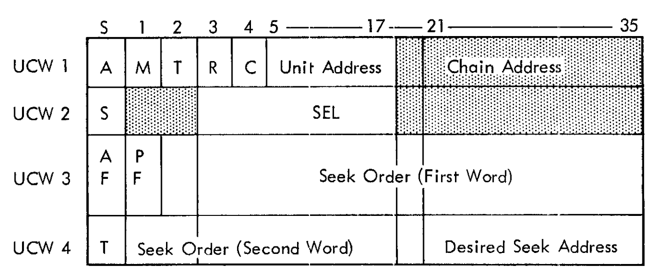


Figure 10. Disk Unit Control Block

The contents of each unit control block is interrupted as follows:

Word 1

This word has the same interpretation as for word 1 of the 729 tape unit control block except that when a disk module is attached to the channel, its unit control block remains in the availability chain even though it may be assigned a system unit function. Also, if the 7631 File Control Unit is a Model III or IV, bit position T will be one (T=1); otherwise, position T equals zero (T=0).

The disk unit address has the format shown in Figure 11.

S—8	9—11	12	13	14—17
Channel (1 through 8)	Device (1 for Disk)	Data Channel Switch (0 or 1)	Access (0)	Module (0 through 9)

Figure 11. Format of Disk Unit Address

Examples:

Channel C, Access, Module 1, Data Channel Switch setting $1 = (3101)_8$

Channel E, Access, Module 9, Data Channel Switch setting $2 = (5151)_8$

Word 2

The interpretation of this word is the same as for word 2 of the 729 tape unit control block except that redundancy message control is not provided. Any disk error will produce an on-line message if recovery is unsuccessful.

Word 3

AF: Seek Request Flag

AF=0: No seek is requested, or an ATTENTION signal was received on the previously requested seek.

AF=1: A seek is requested by the user, or a seek was issued and an ATTENTION signal is awaited. This flag is set to 1 by the user when a seek is requested. It is reset to 0 by IOEX when the ATTENTION signal is received.

PF: Seek Issued Flag (used by IOEX only)

PF=0: No seek was issued.

PF=1: A seek was issued and an ATTENTION signal is awaited.

Seek Order: The six bytes of word 3 and the first two bytes of word 4 are used to form the seek order for any seek requested of IOEX by the user. The last two bytes of word 3 and the first two bytes of word 4 will contain the track address (in BCD) for the desired seek.

Word 4

T: Track Flag

T=0: IOEX will set up the BCD track address for a requested seek, obtaining its information from the binary track address specified by the user in the address of word 4.

T=1: The user, on requesting a seek, has already set up the BCD track address in words 3 and 4. T is reset to zero by IOEX after the seek is issued.

Desired Seek Address: This is the track address (binary) for a seek requested by the user. The address of word 4 is never destroyed by IOEX.

Hypertape Unit Control Block

Each Hypertape drive is represented in IOEX by a four-word unit control block with the format shown in Figure 12.

	S	1	2	3	4	5—17	18	19	20	21—35
UCW 1	A	M		R	C	Unit Address	EWA			Chain Address
UCW 2	S	B		SEL						
UCW 3	AF	PF	I	File Count			Record Count			
UCW 4										

Figure 12. Hypertape Unit Control Block

The Hypertape unit control block is similar to the unit control block for 729 tapes. The differences between a Hypertape unit control block and a non-disk unit control block are as follows:

Word 1

Unit Address: The unit address field for Hypertape is similar to that for disk storage. It has the format shown in Figure 13.

5—8	9—11	12	13	14—17
Channel (1 through 8)	Device (0 for Hypertape)	Data Channel Switch (0 or 1)	0	Unit Number (0 through 9)

Figure 13. Format of Hypertape Unit Address

Examples:

Channel C, Hypertape Drive 1, Data Channel Switch setting $1 = (3001)_8$

Channel H, Hypertape Drive 9, Data Channel Switch setting $2 = (10051)_8$

EWA: End Warning Area Flag

EWA = 0: The unit is not in the EWA.

EWA = 1: The unit is or has been in the EWA.

The EWA flag is turned off only when the tape is returned to the BOT marker; or whenever HCCR, HCHC, or HUNL is executed.

Word 2

R: Select Mode Flag

R=0: Normal select

R=1: Select for backward reading

Both this bit and the S bit are set by the user.

Word 3

AF: AF Flag

PF: PF Flag

These flags are used only by IOEX to control the use of Attention servicing routines.

General Use of IOEX and Unit Control Blocks

The following steps describe the general procedure to be taken by the user when input/output activity for a unit is desired.

Before activity can be requested on an input/output unit, the user must determine whether or not the unit is free for use. This is accomplished by waiting for word 2 of the unit control block to become zero. Once it becomes zero, the user must assume control of the unit by placing the location of his select routine in the decrement of word 2. The user is also responsible for setting bit positions S and R in unit control block word 2. R is omitted for disk.

After the input/output unit is secured for use, the user may request channel activity for the unit by entering the IOEX routine (ACTIV. This routine, which is used to handle data select activity, permits all necessary entries into the user's specified select routine as soon as the channel is free to accept input/output activity from the unit. Non-data select activity is handled by the routine (NDATA.

On the first entry to the select routine, called the (SEL+) or select entry, the user must initiate execution of the input/output commands for the unit. Termination of the commands must always set up a trap condition for the CPU.

On the last entry to the select routine, called the (SEL−) or posting entry, input/output activity for the unit is complete. At this time the user may relinquish control of the unit by setting word 2 of the unit control block to zero, or he may choose a number of other options using information described under "Posting Entry."

The user distinguishes between (SEL+) and (SEL−) entries to his select routine by testing the sign of the accumulator at time of entry.

Normally, any direct reference to a unit control block at other than trap (SEL) time should be trap-protected by the sequence:

```
        ENB     L(0)
         .
         .
         .
        ENB*    (TRAPX
```

Data Transmission Via Select Routine

The initiation of data transmission operations for a unit, and the maintenance of any request queue for a unit, is the responsibility of the user of IOEX. These functions must be provided in a subroutine, labeled SEL for example, which is entered twice for each data transmission operation that results in a trap. The calling sequence from IOEX is:

	TSX	SEL,4	
Return 1			Normal return at either entry
Return 2			Returns used at (SEL−) time because of errors indicated. (See appropriate recovery section.)
Return 3			

Upon either entry to the select routine, IOEX disables channel trapping and provides:

C(IR1)	The 2's complement of channel index (0=A, 1=B, etc.)
S(AC)	Sign of accumulator: plus for (SEL+) and minus for (SEL−)
A(AC)	Location of the unit control block

Select Entry

If the sign of the AC is plus on entry to SEL, the routine must initiate a data select command sequence terminating in a trap. Traps must not be enabled by (SEL+). This is taken care of by IOEX. For 7909 channel operation, the command sequence should terminate with a

```
        TCH     SYSTWT
```

which will produce the necessary trap. In addition, an SMS command which disables ATTENTION interrupts and enables UNUSUAL END interrupts must initiate the command sequence.

For Hypertape, on each entry to the (SEL+) routine, the user must initiate a data select command which will read or write one record. This restriction, which is similar to the restriction on the use of 729 tape units, is necessary to ensure the correct record count in word 3 of the unit control block.

Posting Entry

If the sign of the accumulator is negative on entry to SEL, a trap has occurred as a result of the previous select for the unit. The following information is furnished upon entry:

Sense Indicators—7607 Channel

Bit S Noise record flag
- 1 The record was *not* an apparent noise record.
- 0 The record was an apparent noise record.

Bit 1 End of file (read) or end of tape (write)
- 0 No end of file or end of tape.
- 1 End of file or end of tape.

Bit 2 Permanent redundancy (read only)
- 0 No permanent redundancy.
- 1 Permanent redundancy.

Bit S is on in all cases except when a noise record is detected.

Sense Indicators—7909 Channel, Disk

Bits S, 1-5	7909 Control counter
Bit 6	Input/output check
Bit 7	Sequence check
Bit 8	UNUSUAL END
Bit 9	ATTENTION on Interface 0
Bit 10	ATTENTION on Interface 1
Bit 11	Interface check

Bits 12-35—First four bytes of the 7631 sense data. Bits 12-35 are supplied only if the trap was a result of an UNUSUAL END. In such a case, bit 8 will be a 1.

Sense Indicators—7909 Channel, Hypertape

Bits S, 1-5	7909 Control Counter
Bit 6	Input/output check
Bit 7	Sequence check
Bit 8	UNUSUAL END
Bit 9	ATTENTION on Interface 0
Bit 10	ATTENTION on Interface 1
Bit 11	Interface check
Bits 12-35	Bytes 1, 3, 4, 5, 6, and 7 (packed) of the 7640 sense data: supplied only if the trap was the result of an UNUSUAL END signal, i.e., if bit 8 is a 1.

Cell (COMMM

This word contains the results of a store channel instruction for the 7607/7909 channel.

URRX, 1 Table

Redundancy Counts:

PZE N1,,N2

N1:	Number of recovery entries to (SEL+).
N2=0:	No permanent redundancy (while reading) or no erase areas (writing).
N2=1:	Permanent redundancy (reading) or one or more erase areas (writing). If a permanent redundancy, N1=0. N2 is always zero for disk or Hypertape input/output.

The URRX,1 Table may be referenced by indirectly addressing the communication cell (URRXI. IR1 must contain the complement of the channel index.

If EOT (or EWA if Hypertape) is detected while writing, the EOT bit is set ON in the unit control block, and the EOT indication is given on each subsequent entry to (SEL−) for the unit until the tape is rewound. When an end-of-file indicator is given to (SEL−), the tape position has already been adjusted. The communication location (LTPOS contains the tape position prior to this adjustment. A redundancy indication cannot occur together with an end-of-file indication.

Design of Select Routines

Select routines must not destroy the contents of IR1. IR2 and the sense indicators need not be saved. They must not modify or change the unit address in word 1 of the unit control block or the four high-order bytes (except the sign bit) of word 3 of a disk unit control block.

In disk usage, the select routine must interpret word 1 of the unit control block if use of the compact access and module bits are needed for setting up Prepare to Verify orders, or it must use (FDAMT. (See "Form Disk Order" under "IOEX Utility Routines.")

Select routines should be designed to minimize processing time while the machine is trapped.

Unit Priority on a Channel

Channel Priority Location

When activity begins on a unit of a given channel, IOEX places the address of the unit control block in a channel priority location. This allows the unit to retain priority until all its waiting operations have been completed. IOEX clears this location when the user clears word 2 of the unit control block.

Upon a normal return from a (SEL−) routine, IOEX selects the next unit to be activated by examining the channel priority cell. The same unit will be reselected if control was not relinquished by the user by clearing word 2 of the unit control block.

Use of Channel Priority Location

If the channel priority location is not zero, its address portion is interpreted as the location of the unit control block for the unit to be activated next. Word 2 of the unit control block is then tested.

If word 2 is not zero, (SEL+) for the unit is entered. If it is a disk unit control block, the (SEL+) entry is held up (return is still made to the user) until any pending seek for the unit has been issued and attention received.

If word 2 of the unit control block is zero, the priority cell is cleared and the channel is scanned for a waiting unit, a unit that has a SEL routine specified in word 2 of its unit control block. In addition, if it is a disk unit control block, the AF flag in word 3 must be zero. If a waiting unit is found, the location of its unit control block is placed in the channel priority cell, and (SEL+) is entered for the unit. If none is found, the channel is allowed to become dormant. In the case of a disk, however, any pending seeks will be issued, and the channel will become dormant only if no seeks are waiting to be issued.

Activating a Channel and/or Assigning Priority

A channel which is dormant is activated, or a unit is given top priority on a channel, by means of the routine (ACTIV. The calling sequence is:

```
        TSX     (ACTIV,4
        P       A,T
            return
```

where the parameter A, T gives the address of a location (possibly a System Unit Function Table entry) which contains:

```
        OP      UCB,X,Y
```

UCB is the address of the unit control block of the unit desiring input/output activity. OP, X, and Y are ignored by (ACTIV. Note the double indirectness to the actual unit address.

When P is PZE in the calling sequence, controls are set up so that the user's select routine is entered whenever the channel is free to accept activity on the specified unit. From the discussion under "Unit Priority on a Channel," it is obvious that no action by (ACTIV is necessary if the channel is active upon entry. If such is the case, control is immediately returned to the user.

If the channel is dormant, (ACTIV will enter the user's (SEL+) routine directly before returning. In disk storage, the AF flag in word 3 of the unit control block must be zero for (SEL+) to be entered. If the AF flag is not zero, (ACTIV will first activate the channel by issuing a seek for the access and for any other accesses on the channel requesting the issuance of a seek. The select routines will be entered as the ATTENTION signals are received.

This entry to (ACTIV, requesting the issuance of seeks (P=PZE), is the only entry to the routine which does not require a SEL routine to be specified in word 2 of the unit control block. In such a case, (ACTIV will still issue all requested seeks. This allows the user, if he so desires, to use "seek time" with the knowledge that the seek ATTENTION signal will be recorded when received by IOEX but that no select routine for the unit will be entered. When the user is ready to have the select routine entered, he places its location in word 2 of the unit control block and, to ensure that the channel is activated if it became dormant, reenters (ACTIV with the AF flag equal to zero. To determine if a disk unit is free for use when this option is exercised, the user must test the AF flag as well as word 2 of the unit control block for zero.

When P is MZE in the calling sequence, the indicated unit is given top priority on the channel; that is, the channel priority cell is set for the unit, and (SEL+) is entered as soon as possible. If the entry to (ACTIV is made during non-trap time, control will not be returned to the user until the (SEL+) entry for the unit has occurred. In disk usage, if the AF flag in word 3 of the unit control block equals one (AF=1), the necessary seek will be issued prior to the (SEL+) entry.

If the entry to (ACTIV is at trap time, e.g., during (SEL−), the unit is given top priority by the setting of the channel priority cell. However, return is made immediately. Hence, a subsequent entry to (ACTIV, requesting priority (P=MZE), will cancel the priority effects of the previous entry if it occurs before (SEL+) is entered for the previous entry.

Any entry to (ACTIV at trap time must be made with MZE in the calling sequence, i.e., P=MZE. Also, the entry must be for a unit connected to the channel that caused the trap.

The (ACTIV routine will always enable traps on all channels upon return, unless entry was from a select routine.

(ACTIV may not be entered during (SEL+) time.

(ACTIV makes a validity test on the specified input/output unit. To be valid, the specified location of the unit control block must fall within the range of core storage provided for all unit control blocks. In addition, the A flag of word 1 of the unit control block must be zero. If a unit is judged to be invalid by these criteria, an automatic post-mortem dump is taken after the following message is printed on the System Printer:

```
    ILL UNIT REQ'ST AT XXXXX
```

After the dump is taken, the System Supervisor skips to the next job segment.

This above validity test may be bypassed by using:

```
        TSX     (ACTVX,4
```

in lieu of:

```
        TSX     (ACTIV,4
```

in the calling sequence.

The routine (ACTIV has the same purpose and use for Hypertape as it has for 729 tape units. Namely, upon entry to (ACTIV, a SEL routine must be specified in the appropriate unit control block. Entry is essentially a request by the user for input/output activity for the unit indicated in the calling sequence.

Non-Data Selects

Non-data selects are executed by the routine (NDATA. The calling sequence is:

```
TSX     (NDATA,4
PZE     A,T,NDS
    return
```

The parameter A,T has the same meaning as in the calling sequence for (ACTIV. NDS is interpreted for 729 tape units as follows:

NDS=0	NOP
NDS=1	SDNL
NDS=2	SDNH
NDS=3	REW
NDS=4	RUN
NDS=5	BSR
NDS=6	BSF
NDS=7	WEF

(NDATA gives the specified unit top priority by using (ACTVX with MZE in the calling sequence, i.e., P=MZE. Hence, the non-data select is executed as soon as the unit's present activity is complete. Return is made only after the non-data select has been executed. (NDATA may not be entered during (SEL+) time.

The (NDATA routine will always enable traps on all channels upon return unless entry was from a select routine. (NDATA may be entered at trap time only for a unit which is on the channel that has trapped. Non-data selects for card equipment and disk equipment are ignored.

(NDATA does a validity test on the specified unit control block in the same manner as (ACTIV. The test may be bypassed by using TSX (NDSLX rather than TSX (NDATA. Note that a NOP (NDS=0) entry to (NDATA will have the effect of performing a validity test and nothing else. Backspacing a record across the previous file mark complements the record count in bits 18-35.

WEF causes the following sequence to be executed:

```
WEFX
TCOX*
TRCX
ETTX
```

Recovery is attempted if a redundancy occurs while writing an end-of-file mark. The tape is backspaced, and the file mark is rewritten and checked as often as necessary. If the EOT condition is detected on 729 tape after a nonredundant end of file, return is made to 2, 4. *The normal return for WEF is 3, 4.*

Hypertape orders are also handled by the (NDATA routine. The calling sequence is:

```
TSX     (NDATA,4
PZE     A,T,NDS
    return
```

The parameter A, T is the same as for 729 tape units. NDS is interpreted as follows:

NDS=0	NOP((NDATA NOP)
NDS=28	HCCR—Change Cartridge and Rewind
NDS=30	HRWD—Rewind
NDS=31	HRUN—Rewind and Unload
NDS=32*	HERG—Erase Long Gap
NDS=33*	HWTM—Write Tape Mark
NDS=34*	HBSR—Backspace Record
NDS=35*	HBSF—Backspace File
NDS=36*	HSKR—Skip Record
NDS=37*	HSKF—Skip File
NDS=38	HCHC—Change Cartridge
NDS=39	HUNL—Unload
NDS=42	HFPN—File Protect On

Those orders marked with an asterisk are handled in the same way as non-data selects for 729 tape units. That is, the (NDATA routine uses (ACTVX with an MZE in the calling sequence. Hence, the order will be executed as soon as the unit's present activity is complete. Return is made from the (NDATA routine only after the order has been executed.

The remaining orders will cause an ATTENTION signal upon completion of the operation. They are treated in the following manner: After any present activity is finished, IOEX executes the order by specifying a select routine. Upon receiving the applicable ATTENTION signal, IOEX clears word 2 of the unit control block. The user must not ask for new activity on that unit until word 2 is cleared.

All orders must be given through the (NDATA routine. The return from the (NDATA routine is always to 2, 4. Entry to the (NDATA routine during trap time is permitted only for those Hypertape orders marked with an asterisk, and for those units on the same channel and Data Channel Switch setting as the unit which trapped.

Redundancy Recovery

Writing on 729 Tapes

The redundancy trapping mode is used for a write operation. If the first attempt to write a record produces a redundancy trap, the following procedure is followed:

1. The tape is backspaced one record.
2. An erase area is written. If this operation produces a redundancy check, an operator message is printed.
3. The record is rewritten (that is, (SEL+) is entered) and checked for redundancy.
4. Steps 1-3 are repeated until the record is written correctly or the EOT condition is sensed. After each group of 25 erase operations, an operator message is printed.

IOEX stops when the EOT condition occurs during redundancy recovery. However, there is always at least one erase-rewrite sequence attempted, and, if the rewrite is successful, the stop does not occur.

If, following the successful writing of a record, IOEX determines that the apparent record length was less than three words, an entry is made to (SEL−) with the noise record condition indicated in the sense indicators. Two exits are available for this condition:

Return 1	The record is accepted.
Return 2	The record is accepted, and an operator message is printed indicating that a short record has been written.

The record count is increased before entry to (SEL−). It is not increased before rewrite entries to (SEL+).

On each rewrite entry to (SEL+) during redundancy recovery, URRX,1 has the following configuration:

PZE N,,1

where N is the number of consecutive erase areas which have been written. Following a successful redundancy recovery, on entry to (SEL−), URRX,1 has the same configuration. N in this case is the total number of erase areas written on this recovery.

Reading from 729 Tapes

The redundancy trapping mode is not used during read operations so that a full word count of record size may be secured. If the redundancy occurs as the result of a reading operation, the following steps are taken:

1. If the record is an apparent noise record, (SEL−) is entered with appropriate indication in the sense indicators. If the record is not an apparent noise record, step 2 is taken.

2. The redundancy count (address of URRX,1) is increased by 1. It is initially zero.

3. If all of the following three conditions exist, a tape cleaner action is taken. If one or more conditions do not exist, step 4 is taken.
   a. The redundancy count is 1 or a multiple of 10 plus 1.
   b. The noise record bit in word 3 of the unit control block is off.
   c. There are at least two previous records in the current file.

The tape cleaner action consists of backspacing over the redundant record and the two previous records, giving two dummy reads, and then entering (SEL+) to reread the redundant record.

4. The tape is backspaced over the redundant record, and (SEL+) is entered to reread the redundant record.

5. Steps 1 through 4 are repeated until the record is read correctly or until the redundancy count reaches the value of the assembly parameter RDUNRT. If the value of RDUNRT is reached, a permanent read redundancy is assumed. If such is the case, the record count is increased by one and an operator message is printed if not suppressed by the control bit in word 2 of the unit control block. In addition, the instruction:

PZE 0,,1

is stored in URRX,1, and (SEL−) is entered with the permanent redundancy indication in the sense indicators.

Read redundancy checking and recovery may be suppressed by setting RCTX,1 to zero in (SEL +). The RCTX,1 Table may be referenced by indirectly addressing the communication location (RCTXI. IR1 must control the complement of the channel index. On every entry to (SEL+), RCTX,1 is set on (to nonzero).

Exits from (SEL−) for 729 Tapes

Three exits are available for (SEL−).

Return 1	The record is accepted.
Return 2	The record is considered noise and is discarded. An operator message is printed, and the noise record bit is set on in word 3 of the unit control block. The record count is reduced by 1; URRX,1 is cleared, and (SEL+) is re-entered to read the next record.
Return 3	The redundancy recovery procedure is entered at step 2.

After each successfully completed write operation, and after a redundant read operation, the following test is made for an apparent short (noise) record. The address of the last input/output command is subtracted from the address of the Store Channel word to obtain an apparent word count. If this word count is less than three, IOEX gives the noise record indication to (SEL−).

Obviously, there are sequences of input/output commands which will produce this noise record condition, even when the true record length is greater than two words.

No test is made for use of the indirect addressing feature of input/output commands. If this feature is used, a short record will normally be detected.

Reading and Writing Disk

Figure 14 is a table showing the bit assignments for the first four bytes of the 7631 sense data. When an UNUSUAL END signal occurs for a condition indicated in bytes 3, 4, or 5 of the sense data, IOEX takes the following recovery action.

Recovery Action

(SEL+) is re-entered up to four times. If the UNUSUAL END persists, two seeks are given to recalibrate the arm. This is followed by the original seek requested by the user. (SEL+) is then entered again. If the UNUSUAL END still persists, (SEL+) will be re-entered up to three more times. If success is not achieved, (SEL−) is entered with the proper indication.

Byte	Bit	Assignment	Abbre-viation	Comment
1	2	Reserved	--	Summary Bits
	4	Program Check	PRGCK	
	5	Data Check	DATCK	
	6	Exceptional Condition	EXCND	
2	2	Invalid Sequence	I'SEQ	Program Check
	4	Invalid Code	I'CDE	
	5	Format Check	FRMCK	
	6	No Record Found	NOREC	
3	2	Invalid Address	I'ADR	Data Check
	4	Response Check	RSPCK	
	5	Data Compare Check	DCOMP	
	6	Parity or Cyclic Check	P/CCK	
4	2	Access Inoperative	INOPR	Exceptional Condition
	4	Access Not Ready	NTRDY	
	5	File Frame Circuit Check	DKCRK	
	6	File Adapter Circuit Check	CUCRK	

Figure 14. Bit Assignments for the First Four Bytes of the 7631 Sense Data

Select Exits

Whenever (SEL−) is entered because of an unsuccessful recovery attempt, an operator message will be printed indicating the error. The message will have the following form:

UNIT xDam/s WRITE ERROR–TRCK 00tttt
(READ)
x1,x2,x3, . . . ,x24

where tttt is the last track "seeked" by IOEX and x1, x2,x3,...,x24 are octal numbers representing all 72 bits of the two 7631 sense data words (Appendix C).

In addition, the address portion of URRX,1 will contain the total number of re-entries to (SEL+) made in attempting to recover. On each entry to (SEL+), URRX,1 will be zero or will contain the number of times (SEL+) has been entered previously in attempting to recover from an UNUSUAL END.

The two possible exits from (SEL−) are:

Return 1	Normal return.
Return 2	Not to be used.
Return 3	Re-enter (SEL+) and repeat the recovery actions if necessary.

On any entry to (SEL+), 7631 checking may be qualified by an exclusion flag corresponding to the 7631 sense data listed in Figure 14.

Procedure to Qualify 7631 Checking

Into cell RCTX,1 place a word of the following format:

```
        PZE        FLAGWD
```

where FLAGWD is the location of a word which is an exclusion mask to be tested by IOEX prior to any error procedures.

For example, set exclusion to ignore NOREC, RSPCK, DCOMP, and P/CCK.

```
              CLA      FLBITS
              STO*     (RCTXI
               .
               .
               .
FLBITS        PZE      *+1
              OCT      060107000000
```

The proper summary bit is set on (to nonzero) as long as it is desired to exclude at least one of the errors to which it refers. In the above example, exclusion of the NOREC error also necessitated the exclusion of its summary (PRGCK) bit. On every entry to (SEL+), RCTX,1 will be initialized to zero.

Reading and Writing Hypertape

Figure 15 is a table showing bytes 1, 2, 3, 4, 5, 6, and 7 of the 7640 sense data which have been packed into four bytes as shown in the left-hand columns. The table indicates the associated recovery action taken by IOEX when an UNUSUAL END signal occurs other than on an order issued by the (NDATA routine.

Action 1

The (SEL+) routine is re-entered. If the error persists, the (SEL−) routine will be entered and the proper sense data will be in the sense indicators.

Action 2

The (SEL−) routine is not entered with the error indication until the sequence HBSR (or HSKR) – Enter (SEL+) is repeated ten times without success.

Action 3

An HBSR order followed by an HERG order is executed. The (SEL+) routine is then re-entered. If the same UNUSUAL END signal persists after ten re-entries to (SEL+), the (SEL−) routine is entered with the proper indication.

Action 4

An operator message is printed indicating the corrective action to be taken by the operator. Upon completion of the necessary operator action, the (SEL+) routine is re-entered.

Exits

The three possible exits from the (SEL−) routine are:

Return 1	Normal return.
Return 2	Not to be used.
Return 3	Re-enter (SEL+) and repeat the recovery actions if necessary.

Byte	Bit	Assignment	Abbreviation	Recovery Action		Comment
				RD	WR	
1	1	Operator Required	OPREQ			Summary Bits
	2	Program Check	PRGCK			
	3	Data Check	DATCK			
	4	Exceptional Condition	EXCND			
	5	Selected Drive Not Ready	NTRDY	4	4	Operator Required
	6	Selected Drive Not Loaded	NTLOD	4	4	Program Check
2	1	Selected Drive File Protected	FILPR	-	4	
	2	Not Used	-			
	3	Invalid Order Operation Code	I'ORD	1	1	
	4	Selected Drive Busy	DRBSY	1	1	
	5	Selected Drive at BOT	ATBOT	1	1	
	6	Selected Drive at EOT	ATEOT	1	1	
3	1	Correction Occurred	-			Data Check
	2	Channel Parity Check	PARCK	2	3	
	3	Code Check	CDECK	2	3	
	4	Envelope Check	ENVCK	2	3	
	5	Overrun Check	OVRCK	2	3	
	6	Excessive Skew Check	ESKCK	2	-	
4	1	Track Start Check	TRSCK	2	3	
	2	Not Used	-			Exceptional Condition
	3	Selected DR Read a Tape Mark	-	*		
	4	Selected DR in EWA	-		**	
	5	No Data Transmitted	NDTRN	2	1	
	6	Not Used	-			

* When a tape mark is read, return will be made to the (SEL-) routine with the proper sense data in the sense indicators. The applicable unit control block will contain a zero record count, and the file count will be increased by one. In addition, location (LTPOS will contain word 3 of the unit control block as it was before the tape mark trap occurred.

** When writing continues into the EWA, the (SEL-) routine will be entered with the proper sense data in the sense indicators. In addition, the appropriate bit in word 1 of the unit control block will be set on.

Figure 15. Recovery Action Taken on an Unusual End Signal

Whenever the (SEL−) routine is entered with an UNUSUAL END signal, an operator message is printed indicating the error. The message will have the following form:

```
UNIT xHk/s WRITE ERROR
           (READ)
x1,x2,x3, . . . ,x24
```

where x1,x2,x3,...,x24 are octal numbers representing all 72 bits of the two 7640 sense data words (Appendix D).

As with disk storage, 7640 checking may be qualified upon entry to the (SEL+) routine by means of an exclusion flag corresponding to the packed 7640 sense data shown in Figure 15. Thus, the user can specify with flag bits any unusual conditions which he may wish to ignore. For example:

```
        CLA     FLBITS
        STO*    (RCTXI
        .
        .
        .
FLBITS  PZE     *+1
        OCT     301720000000
```

will cause IOEX to bypass any recovery attempts if any one of the following conditions causes an UNUSUAL END signal:

I'ORD
DRBSY
ATBOT
ATEOT
PARCK

These exclusion bits will also prevent the printing of any on-line message which usually would follow one of the above errors.

Channel Control Tables

The (SEL+) routine can make use of the IOEX Tables of Reset and Load Channel instructions and Start Channel (7909) instructions. This should be done indirectly through the communication locations (RCHXI and (STCXI. IR1 must contain the complement of the channel index.

The form of the tables themselves is:

```
RCHX  RCHA   **  STCX  STCA
      RCHB  .**        STCB
(RCHXI contains PZE   RCHX,1.
(STCXI contains PZE   STCX,1.
```

If (RCHXI is used, the applicable address in the RCHX Table must be set upon every entry to (SEL+).

IOEX Utility Routines

The following are IOEX utility routines that may be used by subsystems operating under control of the System Monitor. The routines (PROUT, (PUNCH, (CVPRT, (PAUSE, (PAWSX, (BCD5R, and BCD5X all enable trapping (except at trap time) upon exit. This trapping can be suppressed by the user by setting the address portion of the IOEX Communication Table entry (ENBSW to nonzero. The address must be reset to zero by the user.

Message Writer

Messages can be printed both on-line and off-line, using the subroutine (PROUT. Off-line messages are actually printed by a separate subroutine, SPOUT, which is called by (PROUT when an off-line message is specified in the (PROUT calling sequence. SPOUT is stored in core storage locations SYSEND-199 through SYSEND. Therefore, if any subsystem uses SPOUT, it must consider SYSEND-200 as the end of usable core storage, rather than SYSEND.

The calling sequence for (PROUT is:

```
TSX     (PROUT,4
PFX     N
P       L1,T1,M1+512*SPR1
P       L2,T2,M2+512*SPR2
.
.
.
P       LN,TN,MN+512*SPRN
```

If PFX=PZE, the message is printed on-line and no reference to SPOUT is made. A subsystem that destroys SPOUT must specify PZE whenever (PROUT is used. PFX is interpreted as PZE for any call to (PROUT during (SEL+) or (SEL−) time.

If PFX=MZE, the message is recorded both off-line and on-line.

If PFX=MON, the message is recorded off-line only.

N is the number of entries following the calling sequence. M words (six characters each) beginning in location L, T are converted and placed in the line image for printing. If P=PZE, the image is taken to be complete and the line is printed. If P=MZE, this line is considered incomplete, and the L,T,M of the next calling sequence entry are used to continue building the image, beginning with the next print position to the right.

If P=PZE, the sense exit SPR is activated after the line is printed. (An SPR appearing in a word with P=MZE is ignored.) For SPOUT, SPR must be 0 (single space), 1 (eject), or 4 (double space). To activate an exit before printing the first line, an entry of the form:

```
PZE     **,,512*SPR
```

should be made in the calling sequence. This will print a blank line, followed by activation of the hub SPR.

(PROUT can be used either at trap time or at non-trap time. Printing is immediate; that is, the printing operation is started before the return from (PROUT.

Alphameric Punch

The entry

```
TSX     (PUNCH,4
```

with sequence similar to (PROUT, excepting sense exits, provides for punching BCD cards on-line for accounting, etc.

Error Pause

The instruction

```
TSX     (PAWSX,4
```

causes a machine stop (HPR −1) after on-line printing of the message

PRES STRT TO GO ON

Pressing the Start key causes on-line printing of the message

. . . CONTINUING

and returns control to 1, 4.

Operator Action Pause

The instruction

```
TSX     (PAUSE,4
```

causes a machine stop (HPR −1) after on-line printing of the message

OPER. ACTION PAUSE. . .

Pressing the Start key causes on-line printing of the message

. . . CONTINUING

and returns control to 1, 4.

BCD Zero Conversion

The instruction

```
XEC     (IBCDZ
```

replaces decimal zeros in the BCD number located in the MQ with BCD zeros, i.e., 12_8. A CRQ is peformed. This routine is used for making disk/drum orders only.

Form Disk/Drum Order

The following routine places AMTTTT disk or drum data into the location of a specified order. It also sets interface bits for SMS and places RR (HA2) into the second word of the order.

The calling sequence is:

```
TSX     (FDAMT,2
BCI     1,00RR0B
PZE     DORDER,T
  return
```

IR4 equals −L (UCB), and bytes 3 through 6 of the MQ contain TTTT head and track in BCD. DORDER, T is the location of the disk/drum order. RR is HA2 identifier, and B is the mask for the 7909 SMS command. Bit 35, i.e., hte low-order bit of B, is set for interface selection by (FDAMT. T may be zero or IR1.

Post-Mortem Dump

The instruction

```
TSX     (STOPX,4
```

causes a transfer to the location SYSDMP in the System Nucleus. A transfer to SYSDMP causes a post-mortem dump of core storage followed by the skipping of cards on the system input file until a $IBSYS, $EXECUTE, $JOB, or $STOP card is encountered.

Symbolic Unit Conversion

The instruction

```
TSX     (SYMUN,4
```

converts the unit address located in the decrement of the MQ, i.e., LDQ with the address of word 1 of the unit control block, to its BCD equivalent as it appears on a $ATTACH card. The results, straddled by any necessary BCD blanks, are located in the AC upon return to 1, 4.

Binary to Decimal Conversion — AC Decrement

The instruction

```
TSX     (DECVD,4
```

converts the binary number in the decrement of AC to its BCD equivalent. The results are located in the low-order end of the MQ. The high-order MQ character is a BCD blank. Control is returned to 1,4.

Binary to Decimal Conversion — AC Address

The instruction

```
TSX     (DECVA,4
```

has the same function as (DECVD, except that the address portion of AC is converted.

Binary to BCD Octal Conversion — MQ Decrement

The instruction

```
TSX     (BCD5R,4
```

converts the binary number in the decrement of MQ to its octal equivalent in BCD code. The results are located in the low-order end of the AC. The high-order AC character is a BCD blank. Control is returned to 1,4.

Binary to BCD Octal Conversion — S, 1-14 of MQ

The instruction

```
TSX     (BCD5X,4
```

has the same functions as (BCD5R above, except that bit positions S, 1-14 of the MQ are converted.

Convert and Append Unit Designation to Message

The words "UNITXXXXXX," where XXXXXX is converted from the unit address in D(MQ), i.e., LDQ with the address of word 1 of the unit control block, can be appended to a message by the calling sequence

```
TSX     (CVPRT,4
PFX     L,T,M+512*SPR
  return
```

where PFX is interpreted the same way as the PFX directly following the TSX (PROUT,4. The remainder of the control word, that is, L,T,M+512* SPR, is interpreted the same as in the control words in (PROUT.

Freeing a Channel

The user tests channel activity with the sequence

```
ZET*    (CHXAC
TRA     *-1
```

IR1 must contain the 2's complement of the channel index (0 = channel A, 1 = channel B, etc.).

When the ZET falls through, all activity on the channel is complete, including any disk-seek, drum-seek, or Hypertape free-running orders.

FAP Symbolic Address	MAP Symbolic Address	IOEX Entry		Function
(ACTIV	.ACTV	TTR	TEST	Activate Routine and Test
(ACTVX		TTR	ACTIV	Activate Routine Without Test
(NDATA	.NDSEL	TTR	TEST	Non-Data Select and Test
(NDSLX		TTR	NDATA	Non-Data Select Without Test
(PROUT	.MWR	TTR	PROUT0	Message Writer
(PUNCH	.PUNCH	TTR	PUNCH0	Alphameric Punch
(ENBSW	.ENBSW	HTR	**	Enable Switch
(PAWSX	.PAWS	TTR	PAWS	Error Pause
(PAUSE	.PAUSE	TTR	PAUSE	Operator Action Pause
(STOPX	.STOP	TTR	SYSDMP	Post-Mortem Dump
(SYMUN	.SYMUN	TTR	SYUNCV	Symbolic Unit Conversion
(DECVD	.DECVD	TTR	BCVDEC-1	Binary to Decimal - AC Decrement
(DECVA	.DECVA	TTR	BCVDEC	Binary to Decimal - AC Address
(CKWAT	.CKWAT	TTR	CKWAIT	Checkpoint Wait
(BCD5R	.BCD5R	TTR	BCD5-1	Binary to BCD Octal, Bits 3-17 of MQ
(BCD5X	.BCD5X	TTR	BCD5	Binary to BCD Octal, Bits 1-14 and S of MQ
(CVPRT	.CVPRT	TTR	CVPRT	Convert and Append Unit Designation to Message
(STOPD	.STOPD	TTR	SYSDMP	Post-Mortem Dump
(CHXAC	.CHXAC	PZE	CHXAC,1	Channel Activity (Indirect Reference)
(URRXI	.URRX	PZE	URRX,1	Redundancy Count (Indirect Reference)
(RCTXI	.RCTX	PZE	RCTX,1	Redundancy Control (Indirect Reference)
(RCHXI	.RCHX	PZE	RCHX,1	Reset Load Channel (Indirect Reference)
(TCOXI	.TCOX	PZE	TCOX,1	Channel Delay (Indirect Reference)
(TRCXI	.TRCX	PZE	TRCX,1	Tape Redundancy Test (Indirect Reference)
(ETTXI	.ETTX	PZE	ETTX,1	End Tape Test (Indirect Reference)
(TEFXI	.TEFX	PZE	TEFX,1	End File Test (Indirect Reference)
(TRAPX	.TRAPX	PZE	(TRAPS	Current Traps Enabled (Indirect Reference)
(TRAPS	.TRAPS	OCT	377	Current Traps Enabled
(COMMM	.COMM	PZE	**,,**	Store Channel Results at Trap
(LTPOS	.LTPOS	PZE	**,,**	Tape Position Before Last Trap
(IOXSI	.IOXSI	BSS	1	Sense Indicators at Trap
(CHPSW	.CHPSW	PZE	**	Checkpoint Switch
(TRPSW	.TRPSW	PZE	**	Trap Switch
(FDAMT	.FDAMT	TTR	FDAMT	Form Disk Order
(SDCXI	.SDCXI	PZE	SDCX,1	SDCX Table (Indirect Reference)
(STCXI	.STCXI	PZE	STCX,1	STCX Table (Indirect Reference)
(COMMD	.COMMD	PZE	**	Store Channel Diagnostic Results
(IBCDZ	.IBCDZ	CRQ	DECRQ,,6	BCD Zero Conversion
(CHXSP	.CHXSP	PZE	CHXSP,1	Priority Switch Table

Figure 16. IOEX Communication Table

System Editor

Function

The function of the System Editor is to modify, add, delete, replace, or rearrange records of the IBSYS Operating System in order to meet the requirements of a particular installation. The IBSYS Operating System is located on one or more System Library Units that may be 729 Magnetic Tape Units, 1301 Disk Storage Cylinders, or 7340 Hypertape Drives. Editing may proceed from any one type of unit (729, 1301, or 7340) to the same or any other type of unit (729, 1301, or 7340).

The System Editor is normally the last file in SYSLB1. When a $IBEDT card is read by the System Supervisor, the System Editor is called into core storage and control is relinquished to it. The System Editor then proceeds to read and process control cards on the system input file.

When a new System Library is produced by the Editor, it is recorded on SYSUT1. It may be formed from a combination of input from the old System Library on SYSLB1 or SYSLB2, from alteration cards on SYSIN1 and/or SYSUT2, or from records on any other specified System Unit.

**EDIT Card*

Control information is transmitted to the System Editor by an *EDIT card. This control card is required for every edit run and must immediately follow the $IBEDT card, whether or not any options are specified on the control card. The format of the *EDIT card is as follows:

```
7          16
*EDIT      [SYSLB2][,HIGH][,MAP][,MODS][,xDam/s]
                   [,LOW ]             [,xHk/s ]
```

The options on this control card are interpreted as follows:

SYSLB2	This option specifies that SYSLB2 is to be edited. If no option is indicated, SYSLB1 is edited.
HIGH LOW	This option specifies the density setting of the new System Library Tape (on SYSUT1). If no density is specified, the density is the same as the old System Library Tape.
MAP	This option specifies that the names of the records and files on the new System Library be listed on the System Output Unit.
MODS	This option specifies that the maintenance control cards (*MODIFY, *REPLACE, etc.) or OCTAL alteration cards that affected a record be listed before the record name on the list specified by the MAP option. If the MAP option is not specified, this option is nullified.
xDam/s xHk/s	If the IBSYS record is being edited to a new System Library on disk or Hypertape, a load card is required; that is, a card is required which loads in the IBSYS record at initial start when it is placed in the card reader and the LOAD CARDS push button on the Operator's Console is pressed. When this option is present, a load card is punched out on the System Punch Unit at the end of the edit. The card loads the IBSYS record from the unit specified by xDam/s (disk) or xHk/s (Hypertape). No card is punched if no unit is specified.

Any text in columns 55 through 72 of the *EDIT card is used as the heading of all printed output from the System Editor.

Arrangement of Subsystems

The arrangement of subsystems in the System Library is indicated in the System Name Table (SYSNAM) of the System Supervisor. The table is used by the System Supervisor to locate a subsystem when a $EXECUTE card is read from the input file. There are two entries in the table for each subsystem, as follows:

```
BCI     1, sysnam
PZE     tfiles, index, nfiles
```

In the first entry, sysnam is the name of the subsystem. It corresponds to the name on the $EXECUTE card that calls the subsystem. In the second entry, tfiles is the number of consecutive files that make up the subsystem; index is the number 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, respectively; and nfiles is the number of files the System Supervisor must skip before scatter-loading the first record of the subsystem. The System Monitor (IBSYS) file is always the first file on SYSLB1, and the System Editor is normally the last file on SYSLB1.

**PLACE Card*

The *PLACE card is used to modify the System Name Table. It is the only control card that can cause a change in the System Name Table.

NOTE: All *PLACE cards must immediately follow the *EDIT card.

The format of the *PLACE card is:

```
7          16
*PLACE     sysnam [, tfiles, index, order]
```

The *PLACE card causes a subsystem name and data concerning the location of the subsystem to be inserted into or deleted from the System Name Table.

If the three arguments tfiles, index, and order are specified on the *PLACE card, an entry is posted in the System Name Table for the subsystem specified by sysnam. The argument tfiles is the number of consecutive files that make up the subsystem; index is the number 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, respectively; and order is the order of the specified subsystem in the System Library Unit with respect to other subsystems. In specifying the order of a subsystem, the IBSYS file should be considered. For example, if index is 1, the first subsystem following the IBSYS file on SYSLB1 is specified by an order of 1; the second subsystem following the IBSYS file is specified by an order of 2; and so forth. However, if index is 2, 3, or 4, the first subsystem on SYSLB2, 3, or 4, respectively, is specified by an order of 0; the second subsystem by an order of 1; and so forth.

If the argument order is omitted and the arguments tfiles and index are included on the control card, the entry in the System Name Table will indicate that the specified subsystem is the last subsystem on the new System Library Unit, followed only by the System Editor.

If the three arguments tfiles, index, and order are omitted from the *PLACE card, the entry in the System Name Table for the specified subsystem is deleted.

When editing to disk storage, a System Loader Table (SLTABL), as well as the System Name Table, is maintained by the System Editor. When the System Library is on disk, the System Supervisor refers to the System Loader Table, instead of the System Name Table, to locate a subsystem when a $EXECUTE card is encountered. However, when editing to disk storage, the System Name Table must still be kept up to date, using the *PLACE card. When editing from disk to tape, the new System Library is generated in accordance with the System Name Table on disk and the System Loader Table is omitted.

When the System Library is on disk, the *PLACE card may also be used to insert or delete home address (HA2) identifiers. This use of the *PLACE card is described in the section "Disk Editing."

Maintenance of the System Name and Loader Tables

The System Name Table is used by the System Supervisor and is, therefore, part of the IBSYS record. When the System Supervisor is loaded into core storage, the System Name Table is loaded into locations SYSORG through SYSORG+99. When the System Editor is subsequently loaded into core storage, it is loaded so that it does not overlay the System Name Table. Therefore, the table is available for updating by the System Editor if any *PLACE cards are read. When the IBSYS record, which is the first record on tape, is edited onto SYSUT1, the System Editor performs the following steps:

1. Examines the location SYSCOR in the Communication Region of the System Nucleus (refer to Appendix A) to determine the current location of the System Name Table.

2. Changes the table as specified by any *PLACE cards.

3. Transfers the updated table to the IBSYS record that is going to be written on SYSUT1. The table is placed in the new IBSYS record, so that it will be located beginning at SYSORG whenever the System Supervisor is loaded into core storage.

When editing to disk, the System Loader Table is automatically maintained by the System Editor. The System Loader Table is located at SYSORG+107 through SYSORG+946. The System Name Table is always maintained by the System Editor from information supplied by *PLACE cards, regardless of whether the System Library is on disk or tape.

Alteration Cards

Standard System Library records are derived from two types of alteration cards: absolute column-binary and octal.

Absolute Column-Binary Cards

The absolute column-binary cards are the standard 22-word-entry type, illustrated in Figure 17. The data or instruction words, beginning with word 3 on the control card, are written on the Sysem Library Unit in a standard System Library record format that enables them to be loaded in sequential core-storage locations, beginning with the location specified in bit positions 21 . . . 30 of word 1.

Octal Cards

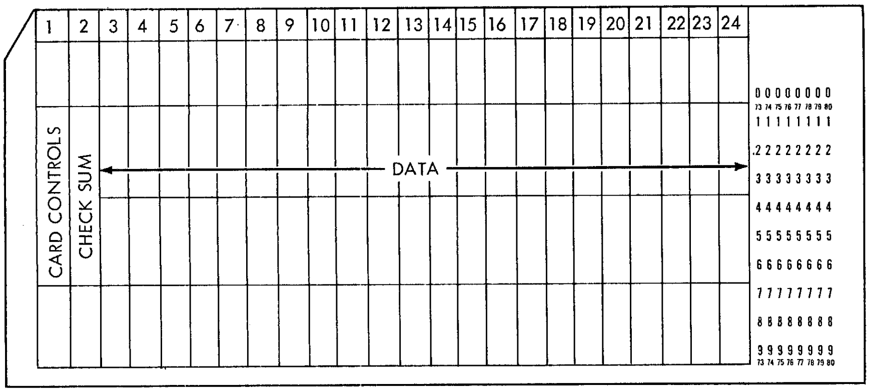
The octal card format is as follows:

1	7	16	72
octloc	{*OCT / OCTAL}	word 1, word 2, . . . ,word n	

The use of * in column 7 is optional. Each octal word must consist of 12 or fewer unsigned digits. When fewer than 12 digits are entered in a word, they are right-justified in the word. The octal words are separated by commas on the control card. The words will be written on the System Library Unit in a standard System Library record format that enables them to be loaded into sequential core-storage locations, beginning with the location specified by the octal address octloc.

Standard System Library Record Formats

Before records are edited onto 729 tape, they are converted by the System Editor into a standard self-loading scatter-load format (shown in Figure 18) that



Word	Bit Positions	Contents
1	S	Must be blank.
	1	Must be blank.
	2	If punched, the System Editor does not compute a check sum from card data for comparison with card's prepunched check sum.
	9	Must be punched.
	11	Must be punched.
	12-17	Count of words in the card, excluding words 1 and 2.
	21-35	Absolute loading address -- the actual address of the core storage location where the first data or instruction word on this card is to be stored.
2	S-35	Check sum -- the logical sum (Add and Carry Logical Word) of all words in this card except word 2.
	3-24	Data or instruction words.

Figure 17. Absolute Column-Binary Card Format

enables them to be loaded by the System Loader in the Nucleus. The records may be converted from similar standard System Library records on disk or Hypertape (shown in Figure 19), from column-binary alteration cards or card images, from octal alteration cards or card images, or from a combination of these.

The System Loader uses the following input/output command sequence to load standard System Library records from 729 tape:

```
IOCP    *+1,,1
xxxx    **,,**
TCH     *-2
```

where xxxx is either an IOCP or IOCT command.

Since the System Supervisor employs the System Loader to load the first record of a subsystem, it is mandatory that the first record be in the standard System Library format shown in Figure 18. A subsystem may also use the System Loader to load succeeding records, provided the records are also in the standard System Library format. No more than one record at a time may be loaded using the System Loader, since redundancy checking is performed and the instruction sequence BSR, RDS is attempted 10 times, if necessary.

Each subsystem consists of one or more files, each file consisting of one or more records. The first record of each subsystem must load the location SYSTRA, in the Communication Region of the Nucleus, with a transfer to the first instruction in the subsystem that is to be executed. This is necessary because the System Supervisor transfers control to SYSTRA after loading the first record of a subsystem specified on the $EXECUTE card.

The first word in a standard record, following the initial input/output command (location LOC1 in Figure 18), must be the name of the record in BCD form, without leading blanks. If a record is the first record of a file, the name of the record is also the name of the file. If the file is the first file of a subsystem, the name of the first record of the file is the name of the subsystem (as specified in the System Name Table), as well as the name of the first file of the subsystem. When the System Library is being edited, the name of the first record of each subsystem in the new System Library is checked against the entry for the subsystem in the System Name

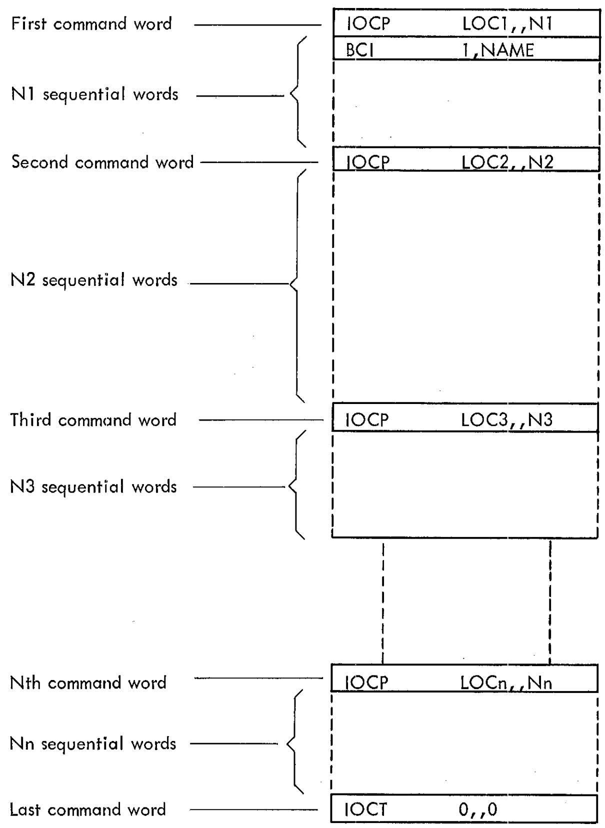


Figure 18. Standard System Library Record Format for 729 Tape

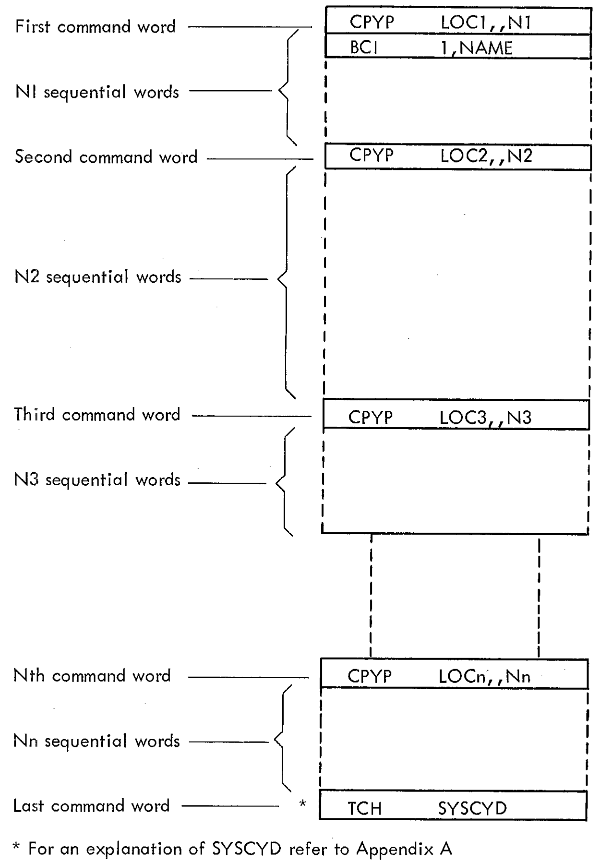


Figure 19. Standard System Library Record Format for Hypertape and Disk

Table. If a discrepancy exists, an error message is printed.

The record name is the name printed when the MAP option on the *EDIT card is exercised. It is also used by the System Editor to identify a record, a file, or a subsystem.

Note, however, that the System Supervisor locates a subsystem specified on a $EXECUTE card by referring to the System Name Table or, if the System Library is on disk, by referring to the System Loader Table. The last record on a System Library Tape has the identification *EOT and is used to inform the System Editor of the logical end of tape.

When a subsystem or part of a subsystem is loaded into core storage, it must not overlay the System Nucleus.

If the subsystem is not loaded into the area occupied by the Input/Output Executor (IOEX) or the off-line writing routine SPOUT, IOEX and SPOUT are immediately available for use by the subsystem. The off-line writing routine SPOUT cannot be used if IOEX is overlaid.

Before records are edited onto Hypertape or disk, they are converted by the System Editor into a standard System Library format (shown in Figure 19) similar to the format for 729 tape, that enables them to be loaded by the System Loader. The records may be converted from standard System Library records on 729 tape (shown in Figure 18), from column binary alteration cards or card images, from octal alteration cards or card images, or from a combination of these. The System Loader uses the following input/output command sequence to load standard System Library records from Hypertape or disk:

```
CPYP    *+1,,1
xxxx    **,,**
TCH     *-2
```

where xxxx is either a CPYP or TCH command.

Since the System Supervisor employs the System Loader to load the first record of a subsystem, it is mandatory that the first record be in the standard System Library format as shown in Figure 19.

When loading standard System Library records from disk, the cylinder mode of operation is used. A sample scatter-read program for reading records from disk is shown in Appendix E.

When a disk record extends across two cylinders, the System Editor sets the last word on the first of the two cylinders to the following:

```
TCH         SYSCYD,,yyy
```

where yyy is the location of the first track of the second of the two cylinders. The remainder of the record on the second cylinder begins with an appropriate CPYP command. If editing proceeds from disk to magnetic tape, any TCH command within a record that was written because the record straddled two cylinders is reset by the System Editor to a CPYP command with a 0 word count.

In a disk record, the low-order bits of the decrement in the final TCH SYSCYD command points to the track origin of the next sequential record, and the high-order bit of the decrement is set to 1. The 1-bit indicates to the System Loader that the TCH is the true end of the record and not just the end of a cylinder.

Maintenance Control Cards

The actual addition, deletion, rearrangement, or modification of subsystems or records and files within a subsystem is performed by the use of the maintenance control cards described in this section. A *PLACE card does no more than post, in the System Name Table, the postition of a subsystem. The maintenance control cards must be in the same order as the records to which they refer. If a maintenance control card refers to a record out of sequence in the edit deck, the effect of the control card is nullified and a message is printed.

*MODIFY Card

The format of the *MODIFY card is as follows:

```
1           7          16
[TAPE]      *MODIFY    recnam
```

where recnam is the name of a record in the old System Library (SYSLB1 or SYSLB2) that is to be modified.

If TAPE does not appear in columns 1-4, this control card causes the specified record to be consolidated with alteration cards that follow this control card on SYSIN1. The alteration cards on SYSIN1 may be any combination of octal and column-binary cards.

If TAPE appears in columns 1-4, the *MODIFY card causes the specified record to be consolidated with alteration cards on SYSUT2. The alteration cards on SYSUT2 must be in the form of column-binary card images.

The System Editor performs the following steps after reading a *MODIFY card:

1. Transfers to SYSUT1 all files, file marks, and records on SYSLB1 (or SYSLB2, if specified on the *EDIT card) up to, but not including, the record specified on the *MODIFY card. If necessary, records are converted to the standard System Library record format shown in Figure 18, when SYSUT1 is 729 tape, or to the standard format shown in Figure 19, if SYSUT1 is Hypertape or disk.

2. Reads the specified record into core storage.

3. Deletes the IOCT or TCH command at the end of the record.

4. Appends an IOCP (or CPYP) command of the following form to the record:

```
IOCP        LOC,,N
```

where LOC is the storage location specified in the first alteration card and N is the number of words, contained on the first alteration card and the alteration cards immediately following it, that are to be loaded into contiguous core-storage locations.

5. Places the N words from the alteration cards into the record following the newly appended command word.

6. Repeat steps 4 and 5, with appropriate LOC and N values, until, if TAPE is specified, a transfer card is encountered on SYSUT2 or, if TAPE is not specified, until a new System Editor control card is encountered on SYSIN1.

7. Appends, to the end of the record, an IOCT command with a word count of zero or, if SYSUT1 is Hypertape or disk, a TCH SYSCYD command.

8. Writes the expanded record onto SYSUT1.

After all actions required by the *MODIFY card are completed, SYSLB1 (SYSLB2) will be positioned just after the specified record and before the next record (or file mark, if not a record).

*REPLACE Card

The format of the *REPLACE card is as follows:

```
1           7          16
┌FILE┐      *REPLACE   {recnam}, SYSxxx
└TAPE┘                 {sysnam}
```

If neither TAPE nor FILE appears in columns 1-4, the *REPLACE card causes an entire record (specified by recnam) in the old System Library on SYSLB1 (SYSLB2) to be replaced on SYSUT1, with a new record formed entirely from the alteration cards that follow the *REPLACE card on SYSIN1. If TAPE appears in columns 1-4, the specified record is replaced by a new record formed from the alteration cards on SYSUT2, rather than SYSIN1.

Before the specified record is replaced, all of the files, file marks, and records that precede the record on SYSLB1 (SYSLB2) are transferred to SYSUT1. The replacement record represented by the alteration cards and, if necessary, the records transferred from SYSLB1 (SYSLB2) are converted to the standard System Library record format shown in Figure 18 if SYSUT1 is 729 tape, or to the standard format shown in Figure 19 if SYSUT1 is Hypertape or disk.

The alteration cards on SYSUT2 must be column-binary card images when TAPE is specified. When TAPE and FILE are not specified, the alteration cards that follow the *REPLACE card on SYSIN1 may be any combination of octal and column-binary cards. As with the *MODIFY card, the end of the alteration cards is signaled by a transfer card when TAPE is specified, or by a System Editor control card when TAPE and FILE are not specified.

When all of the operations called for by the *REPLACE or TAPE *REPLACE card are completed, SYSLB1 (SYSLB2) is positioned just after the record specified by recnam, but before the next record or file mark.

If FILE appears in columns 1-4 of the *REPLACE card and the name appearing in columns 16-21 matches a subsystem name in the System Name Table, all records, files, and file marks are read in from SYSLB1 (SYSLB2) and written out on SYSUT1, until the specified subsystem is located in accordance with the entry in the System Name Table. When the subsystem is located, it is replaced on SYSUT1 by the files on SYSXXX. If the number of files in the replacement differs from the original, a *PLACE card reflecting this must be inserted into the edit deck following the *EDIT card, but preceding the maintenance control cards.

The first word of each record read from SYSXXX is checked for an IOCP (or CPYP) command. If it is not an IOCP (or CPYP) command, the record is considered nonstandard and is duplicated without change onto SYSUT1. If it is an IOCP (or CPYP) command, the record is considered a standard System Library record. In this case, the record is duplicated onto SYSUT1, except for the last word. The last word is changed to an IOCT 0,,0 command if SYSUT1 is 729 tape, or to a TCH SYSCYD command if SYSUT1 is Hypertape or disk. If SYSUT2 is disk, the required TCH command is inserted whenever a cylinder boundary is crossed while a record is being written.

If FILE appears in columns 1-4 of the *REPLACE card and the name appearing in columns 16-21 does not match a subsystem name in the System Name Table, all records, files, and file marks are read in from SYSLB1 (SYSLB2) and written out on SYSUT1, until a record with the specified name is located. When the specified record is located, it, and all succeeding records (if any) up to and including the next file mark, are replaced on SYSUT1 by the next file and file mark on SYSXXX. The replacement records are processed by the System Editor as described in the preceding paragraph. To replace an entire file, the record name specified on the FILE *REPLACE card must be the name of the first record in the file. However, if the name of the first record in the file matches a subsystem name in the System Name Table, all of the files in the subsystem will be replaced, as previously described.

When all actions required by the FILE *REPLACE card are completed, SYSLB1 (SYSLB2) is positioned after the file mark that follows the last file (or part of a file) replaced.

*INSERT Card

The format of the *INSERT card is as follows:

```
1          7         16
[TAPE]     *INSERT   [FILEMK]
```

If TAPE and FILEMK are not specified, the *INSERT card causes a new record, formed from octal and/or column-binary alteration cards following the *INSERT card on SYSIN1, to be written on SYSUT1 at its current position. The end of the alteration cards on SYSIN1 is indicated by a new System Editor control card.

If TAPE appears in columns 1-4 and FILEMK is not specified, the new record is formed from alteration cards in the form of column-binary card images on SYSUT2. The end of the alteration cards on SYSUT2 is indicated by a transfer card.

The new record, represented by the alteration cards on SYSIN1 or SYSUT2, is converted to the standard System Library record format shown in Figure 18, if SYSUT1 is 729 tape, or to the standard format shown in Figure 19, if SYSUT1 is Hypertape or disk.

If FILEMK is specified, a file mark is inserted on SYSUT1 at its current position. When SYSUT1 is disk, a flag, analogous to a tape file mark, is placed into a disk record address area (shown in Figure 20). A single *INSERT card cannot be used both to insert a new record and to write a file mark.

The position of SYSLB1 (SYSLB2) is not changed by an *INSERT card.

*REMOVE Card

The format of the *REMOVE card is as follows:

```
1          7         16
[FILE]     *REMOVE   {recnam}
                     {sysnam}
                     {FILEMK}
```

If neither FILE nor FILEMK is specified, the *REMOVE card causes the record specified in columns 16-21 to be spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1. Before the specified record is removed, all of the files, file marks, and records that precede the record on SYSLB1 (SYSLB2) are transferred to SYSUT1. After the *REMOVE card is processed, SYSLB1 (SYSLB2) is positioned immediately after the specified record.

If FILEMK is specified in columns 16-21, the next end-of-file mark encountered on SYSLB1 (SYSLB2) is spaced over and omitted from SYSUT1. All records that precede the file mark on SYSLB1 (SYSLB2) are transferred to SYSUT1. SYSLB1 (SYSLB2) ends up positioned immediately after the file mark that was omitted from SYSUT1.

If FILE is specified and the name in columns 16-21 is a subsystem name in the System Name Table, all the files and accompanying file marks associated with the subsystem (as defined by the System Name Table entry) are spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1.

If FILE is specified and the name in columns 16-21 is the name of a record, but not the name of a subsystem posted in the System Name Table, the specified record and all succeeding records (if any), up to and including the next file mark, are spaced over on SYSLB1 (SYSLB2) and omitted from SYSUT1. If an entire file is to be removed, the name specified on the FILE *REMOVE card must be the name of the first record in the file. However, if the name of the first record in the file matches a subsystem name in the System Name Table, all of the files in the subsystem will be removed, as previously described.

Before a specified subsystem or record is removed, all of the records, files, and file marks that precede it on SYSLB1 (SYSLB2) are transferred to SYSUT1.

*AFTER Card

The format of the *AFTER card is as follows:

```
1          7          16
[FILE]     *AFTER     {recnam }
                      {sysnam }
                      {FILEMK }
```

If neither FILE nor FILEMK is specified, the *AFTER card causes the reading of control cards to be suspended until all files, file marks, and records on SYSLB1 (SYSLB2), up to and including the record specified in column 16-21, have been transferred to SYSUT1.

If FILEMK is specified in columns 16-21, the next file mark on SYSLB1 (SYSLB2) and all records preceding it are transferred to SYSUT1.

If FILE is specified and the name in columns 16-21 is a subsystem name in the System Name Table, all of the files, file marks, and records on SYSLB1 (SYSLB2), up to and including the files and accompanying file marks associated with the specified subsystem (as defined by the System Name Table entry), are transferred to SYSUT1.

If FILE is specified and the name in columns 16-21 is the name of a record, but not the name of a subsystem posted in the System Name Table, all of the files, file marks, and records on SYSLB1 (SYSLB2), up to and including the next file mark following the specified record, are transferred to SYSUT1.

*DUP Card

The format of the *DUP card is as follows:

```
7          16
*DUP       SYSxxx, SYSyyy, n
```

This control card transfers n files from SYSxxx to SYSyyy. The transfer proceeds up to and through the nth file mark read on SYSxxx. If n is blank, 1, or 0, one file and the file mark following it are transferred to SYSyyy.

If SYSyyy is SYSUT1, the transfer of files from SYSxxx is performed as described for the FILE *REPLACE card, as follows. The first word of each record of the files read from SYSxxx is checked for an IOCP (or CPYP) command. If it is not an IOCP (or CPYP) command, the record is considered nonstandard and is duplicated without change onto SYSUT1. If it is an IOCP (or CPYP) command, the record is considered a standard System Library record. In this case, the record is duplicated onto SYSUT1, except for the last word. The last word is changed to an IOCT 0,,0 command if SYSUT1 is 729 magnetic tape, or to a TCH SYSCYD command if SYSUT1 is Hypertape or disk. If SYSUT1 is disk, the required TCH command is inserted whenever a cylinder boundary is crossed while a record is being written.

*REWIND Card

The format of the *REWIND card is as follows:

```
7          16
*REWIND    SYSxxx
```

This control card rewinds the unit assigned to the specified system unit function. When the unit assigned to the specified system unit function is disk, the *REWIND card causes the System Editor to begin its next read or write operation on the unit at the load point defined by the parameters on the $AS card when the unit was assigned to the specified system unit function.

The *REWIND card may be used to define the starting tape position or disk load point before using a *DUP card. The *REWIND and *DUP cards may be used to rearrange subsystems in the System Library or to incorporate, into the System Library, special-format records that have been previously edited by a subsystem.

*CHECK Card

The format of the *CHECK card is as follows:

```
7          16
*CHECK     count, oldnam, newnam
```

This control card causes a test to be made to ensure that the correct number of editing cards were read and that the correct System Library tape was processed. The argument count is the number of alteration cards and maintenance control cards, including the *CHECK card, but not the *EDIT card. The argument oldnam is a name which is compared with the System Library name in the fourth word of the *EOT record on SYSLB1 (SYSLB2). If the two names are not the same, an error message is printed. The argument newnam is the name assigned to the new System Library (tape or disk) on SYSUT1. The arguments must appear in the order given. If an argument is blank, the corresponding operation is ignored.

If more than one *CHECK card is read during an edit run, only the last one is processed.

*REMARK Card

The format of the *REMARK card is as follows:

```
7           16
*REMARK     any remark
```

This control card causes the characters in columns 16-72 of the control card to be listed on the System Printer and the System Output Unit.

Termination of Editing

The System Editor completes the editing process when an end of file or a control card with a 7 and 8 punch in column 1 is encountered on SYSIN1. The System Editor performs the following steps to terminate the editing process:

1. Transfers from SYSLB1 (SYSLB2) to SYSUT1 all remaining files, file marks, and records up to, but not including, the *EOT record on SYSLB1 (SYSLB2).
2. Reads the *EOT record from SYSLB1 (SYSLB2) and performs the tests and changes specified on the last *CHECK card read.
3. Writes the new *EOT record and a file mark on SYSUT1.
4. Rewinds SYSLB1 (SYSLB2) and SYSUT1.

The System Editor then reads the next card on SYSIN1, expecting to find a $IBSYS, $JOB, $EXECUTE, or $STOP card.

Disk Editing

When editing to disk storage, a System Loader Table, as well as the System Name Table, is maintained by the System Editor. When the System Library is on disk, the System Supervisor refers to the System Loader Table, instead of the System Name Table, to locate a subsystem when a $EXECUTE card is encountered. However, when editing to disk storage the System Name Table must still be kept up to date, using the *PLACE card. The System Loader Table is part of the IBSYS record which must be the first record edited to disk.

Records which do not have an IOCP or CPYP command as their first word, containing the number of words following, are not included in the System Loader Table. A record is not considered to be a standard System Library record if it begins with any of the following commands: IORT, IORP, IOCT, IOSP, or IOST.

An end of record or an end of file is indicated on disk by a flag in the record address area. The format of a record address area is shown in Figure 20. An end of record is indicated by a 1-bit in bit position S and an end of file is indicated by a 1-bit in bit position 1.

E O R	E O F		Seek Chain Address		"Pseudo" Record Count to simulate SCHx Instruction
S	1	2	3 — 17	—	21 — 35

Figure 20. Format of Disk Record Address Area

The System Editor writes using the home address operation. Writing is in the 6-bit mode using a 465-word track format. ~~The 1301 Disk Storage used~~ by the Disk ~~System Editor~~ ~~must~~ have ~~the~~ Verify Cylinder feature.

If an error occurs while editing to disk, the format track, which was assumed good at the start, is rewritten to ensure its validity, and editing continues. If a second error occurs on the same select request, editing continues, if possible, and the condition is ignored.

When editing involves disk, the HA0 switch on the IBM 7631 File Control must be ON (up). To protect the System Library, a home address (HA2) identifier containing BM0000 is written on each track containing records of the System Library. The format key should be unlocked to enable the System Editor to rewrite format tracks. It should be locked again when the edit run is completed.

The *PLACE card may be used to insert or delete home address (HA2) identifiers. This use of the *PLACE card is illustrated by the following examples.

Example 1

Starting with cylinder 10, write home address (HA2) identifiers of PQ on 15 cylinders assigned to the specified system unit function.

```
1          7          16
SYSxxx     *PLACE     PQ, 15, 10
```

Example 2

Reset to general use code 00 the home address (HA2) identifiers on 250 cylinders assigned to the specified system unit function beginning with cylinder 0.

```
1          7          16
SYSxxx     *PLACE     00, 250, 0
```

Care should be taken to ensure that the cylinder limits are defined correctly, since incorrectly defined limits may result in the destruction of valid data.

Example 3

If it is desired to write home address (HA2) identifiers on the cylinders assigned to a specified system unit function and to bypass subsequent editing, the following *PLACE card should be inserted in the edit deck, after all other *PLACE cards and preceding the end-of-file card.

```
1         7
SYSxxx    *PLACE
```

Editing Relocatable Records

If a standard relocatable record is located on SYSUT2 when one of the following maintenance control cards is read, the System Editor converts it to a format similar to the standard System Library format.

```
1         7         16
TAPE      *REPLACE  recnam
TAPE      *INSERT   recnam
```

Two IOCP (CPYP) commands and a record name are placed at the beginning of the record by the Editor, as follows:

```
IOCP    LOC1,,1
BCI     1,NAME
IOCP    LOC2,,N
```

where LOC1 is 45000_8, NAME is the record name specified on the *REPLACE or *INSERT card, LOC2 is 45002_8, and N is the number of words in the relocatable record. An IOCT command with a word count of zero is appended to the record if SYSUT1 is 729 tape, or a TCH SYSCYD command is appended to the record if SYSUT1 is Hypertape or disk.

Once a relocatable record is placed in the System Library in the standard System Library format, all maintenance control cards that are applicable to non-relocatable records in the standard System Library format, except the *MODIFY card, are also applicable to the relocatable record.

Editing Examples

The following are examples of typical editing jobs that might be performed by the System Editor.

Example 1

The sample job deck in Figure 21 might be used to prepare a new 729 System Library Tape with corrections to record REC1 and complete replacement of records ABC and XYZ.

Example 2

The sample job deck in Figure 22 might be used to prepare, from a 729 System Library Tape, a new System Library on channel C, Hypertape Drive 2, Data Channel Switch setting 2. The new System Library Tape is to be used on channel C, Hypertape Drive 0, Data Channel Switch setting 2.

Example 3

The sample job deck in Figure 23 might be used to prepare, from the System Library Tape prepared in editing example 2, a new System Library on channel C, disk module 1, Data Channel Switch setting 1.

Example 4

The sample job deck in Figure 24 might be used to rearrange the subsystems on a 729 System Library Tape so that the subsystem SYSTMA, consisting of two files located just after the IBSYS file, will be repositioned just after the subsystem SYSTMX. Assume SYSTMX is presently the last three files just before the System Editor.

```
1     7          16
-----------------------------------------------------------
$JOB             729 TAPE TO 729 TAPE EDIT
$IBSYS
$IBEDT
      *EDIT      MAP,MODS
      *MODIFY    REC1
      .
      .
      .
      .    (*OCT AND/OR COLUMN-BINARY
      .       CARDS ON SYSIN1)
      .
      .
      .
TAPE  *REPLACE ABC
   (COLUMN-BINARY CARD IMAGES ON SYSUT2)
TAPE  *REPLACE XYZ
   (COLUMN-BINARY CARD IMAGES ON SYSUT2, SEPARATED
      FROM ABC CARD IMAGES BY A TRANSFER CARD)
   (END-OF-FILE CARD)
$STOP
```

Figure 21. Sample Job Deck for Making Corrections to a 729 System Library Tape

```
1     7          16
-----------------------------------------------------------
$JOB             729 TAPE TO HYPER EDIT
$IBSYS
$ATTACH          CH2/1
$AS              SYSUT1
$IBEDT
      *EDIT      MAP,MODS,CH0/1
      *CHECK     1,729SYS,HYPSYS
   (END-OF-FILE CARD)
$IBSYS
```

Figure 22. Sample Job Deck for Preparing a System Library on Hypertape from a 729 System Library Tape

```
1     7          16
-----------------------------------------------------------

   (LOAD USING HYPERTAPE LOAD CARD PUNCHED
      DURING JOB LISTED IN FIGURE 22)

$JOB             HYPER TO DISK EDIT
$IBSYS
$ATTACH          CD01/0
$AS              SYSUT1,010,222,BM
$IBEDT
      *EDIT      CD01/0
   (END-OF-FILE CARD)
$STOP
```

Figure 23. Sample Job Deck for Preparing a System Library on Disk Storage from a System Library on Hypertape

```
1     7        16
-----------------------------------------------------------------------------------
$JOB           REARRANGE SYSTMA ON 729 SYSTEM LIBRARY TAPE
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
      *PLACE   SYSTMA                DELETE OLD SYSTMA REFERENCE IN SYSNAM
      *PLACE   SYSTMA,2,1            INSERT NEW SYSTMA REFERENCE IN SYSNAM
      *AFTER   FILEMK                POSITION SYSLB1 AFTER THE IBSYS FILE
      *REWIND  SYSUT4                ENSURE SYSUT4 IS REWOUND
      *DUP     SYSLB1,SYSUT4,2       DUPLICATE SYSTMA (2 FILES)
FILE  *AFTER   SYSTMX                POSITION SYSLB1 AFTER SYSTMX
      *REWIND  SYSUT4                POSITION SYSUT4 AT BEGINNING OF SYSTMA
      *DUP     SYSUT4,SYSUT1,2       PUT SYSTMA ON NEW LIBRARY TAPE
   (END-OF-FILE CARD)                FINISH UP EDIT
$IBSYS
```

Figure 24. Sample Job Deck for Rearranging Subsystems on a 729 System Library Tape

System Library Preparation and Maintenance

Introduction

The complete 7090/7094 IBSYS Operating System is preassembled for use with 729 Magnetic Tape Units and is distributed in binary form on a single 729 System Library Tape. This tape can be used immediately without change by any installation employing 729 Magnetic Tape Units, either for production job processing or for preparing a new System Library that meets the specific requirements of an installation.

In addition to the preassembled 729-capability System Library Tape, symbolic tapes containing the complete Operating System are also available. On these tapes, the IBJOB Subroutine Library is written in Macro Assembly Program (IBMAP) language, and the remainder of the Operating System is written in FORTRAN II Assembly Program (IBSFAP) language. The symbolic tapes may be used, if necessary, to reassemble all or parts of the Operating System. The System Monitor and each of the subsystems on the symbolic tapes contain assembly parameters which may be replaced before assembly to change certain operating characteristics of the System and thereby tailor it to the needs of a particular installation. If an installation is to employ 1301 Disk Storage and/or 7340 Hypertape Drives, parts of the Operating System on one of the symbolic tapes must be updated (to replace assembly parameters), assembled, and then edited onto a new System Library Tape, together with the remainder of the System. The preassembled 729-capability magnetic tape and an update-edit job deck that is distributed with the System may be used to Drum perform these operations.

This section of the publication contains information and procedures for performing the following:

1. Preparing a backup System Library Tape.
2. Changing System Unit assignments by patching.
3. Preparing duplicate 729 System Library tapes for alternate use by the System Monitor.
4. Adding 1301 Disk Storage and/or 7340 Hypertape capability.
5. Reassembling the System Monitor.
6. Incorporating a user-designed installation accounting routine.
7. Incorporating user programs as subsystems under System Monitor control.
8. Incorporating IBM modifications to the 7090/7094 IBSYS Operating System.

Additional information on changing the operating characteristics of the subsystems operating under System Monitor control is contained in the manuals for the subsystems.

Distributed Configuration of Input/Output Units

The distributed 729-capability System Library Tape is assembled for the configuration of referable input/output units shown in Figure 25. This configuration is designed to accommodate the actual configuration of 729 Magnetic Tape Units existing at most installations. A unit control block is generated by the System Monitor for each of the referable units. However, an actual physical unit need not, and normally does not, exist at a particular installation for each unit control block generated when the distributed tape is used.

Channel	Unit Type	Number	Attached Units	Detached Units
Channel A	711 Card Reader	1	RDA	
	721 Card Punch	1	PUA	
	716 Printer	1	PRA	
	729 Magnetic Tape	10	A1...A8	A9,A10
Channel B	729 Magnetic Tape	10	B1...B8	B9,B10
Channel C	729 Magnetic Tape	6	C1...C4	C5,C6
Channel D	729 Magnetic Tape	6	D1...D4	D5,D6

Figure 25. Configuration of Referable Input/Output Units on Distributed System Library Tape

On the distributed preassembled System Library Tape, the card reader, the card punch, the printer, and eight of the 729 Magnetic Tape Units are assigned to the system unit functions shown in Figure 26. The remaining system unit functions have no units assigned. In general, the procedures described in this section are based on the assumption that the units listed in Figure 26 are physically connected.

If the System Monitor is assembled from the distributed symbolic tape without prior changes in assembly parameters, it will have the same configuration of referable units (Figure 25) and system unit assignments (Figure 26) as the distributed preassembled System Library Tape.

Preparing a Backup System Library Tape

Although the following procedure is optional, it is strongly recommended, since it produces a System Library map for future refernce and a duplicate System Library Tape for emergency use.

Unit	System Unit Assignments	Density Assignment
RDA	SYSCRD	
PUA	SYSPCH	
PRA	SYSPRT	
A1	SYSLB1	high
A2	SYSIN1 and SYSIN2	low
A3	SYSUT1	high
A4	SYSUT3	high
B1	SYSOU1 and SYSOU2	low
B2	SYSPP1	low
B3	SYSUT2	high
B4	SYSUT4 and SYSPP2	high

Figure 26. System Unit Assignments on Distributed System Library Tape

1. Place the sample job deck shown in Figure 27 in the system input file on SYSIN1 (A2).
2. Mount the distributed System Library Tape on SYSLB1 (A1).
3. Follow the initial start procedure described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.

A duplicate of the distributed System Library Tape will be produced on SYSUT1 (A3), and a map of the System Library will be produced on SYSOU1 (B1).

```
1     7        16
------------------------------------------------------
$JOB           BACKUP SYSTEM LIBRARY
$IBSYS
$DATE
$IBEDT
      *EDIT    MAP
   (END-OF-FILE CARD)
$STOP
```

Figure 27. Sample Job Deck for Preparing a Backup 729 System Library Tape

Changing System Unit Assignments by Patching

The System Unit Function Table in the System Nucleus is generated at initial start from an auxiliary 19-word System Unit Function Table in the IBSYS record of the System Monitor. A symbolic listing of the distributed version of the auxiliary table is shown in Figure 36. The table is located in core storage, beginning at 15140_8 (SYSLB1) and ending at 15162_8 (SYSUT4). Note that in the distributed version of the table, the same unit (B4) is assigned to both the SYSPP2 and SYSUT4 functions. This unit is so assigned in the distributed version because SYSPP2 functions as a spill tape for the System Core-Storage Dump Program. Therefore, it must have a unit assigned to it. If SYSPP1 and SYSPP2 are to be used alternately for reel switching purposes, then different units must be assigned to SYSPP2 and SYSUT4. The unit assigned as SYSPP2 cannot be disk storage.

Temporary changes in the assignment of units to system unit functions can be made using the $ATTACH and $AS cards or the $SWITCH card. However, to permanently change the assignment of a unit to a system unit function, the entry for the function in the auxiliary System Unit Function Table must be changed. This may be done during an update and reassembly of the System Monitor, as described in the sections "Adding 1301 Disk Storage and/or 7340 Hypertape Capability" and "Reassembling the System Monitor." Alternatively, entries in the table can be changed, without reassembly, by patching them (using octal alteration cards) during an edit run, as described in the following text.

Changing the Assignment of 729 Magnetic Tape Units and Card Equipment

Each entry for 729 Magnetic Tape Units and card equipment in the auxiliary System Unit Function Table has the following format:

PFX CHANNEL,,UNIT

where PFX is the density assigned to the system unit function; MZE for high density and PZE for low density. CHANNEL is a number from 1 through 8 (corresponding to channels A through H) which indicates the channel of the unit assigned to the system unit function. For the configuration of input/output units on the distributed tape, channel can range from 1 to 4. UNIT is the number of the 729 Magnetic Tape Unit (1 through 10), card reader (11), card punch (12), or printer (13) assigned to the system unit function. As an example, if reel switching is desired for the system input, output, and peripheral punch functions, the sample job deck in Figure 28 might be used. Assuming that the System Library Tape on SYSLB1 is the distributed tape, this job deck will assign B5 as SYSOU2, A6 as SYSIN2, and B6 as SYSPP2; all at low density.

```
1     7        16
------------------------------------------------------
$JOB           CHANGE SYSTEM ASSIGNMENTS
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
      *MODIFY  IBSYS
15150 *OCT     000005000002
15152 *OCT     000006000001
15154 *OCT     000006000002
   (END-OF-FILE CARD)
$STOP
```

Figure 28. Sample Job Deck for Changing System Unit Assignments

Changing the Assignment of 7340 Hypertape Drives

Each entry in the auxiliary System Unit Function Table for 7340 Hypertape Drives has the following format:

PFX CHANNEL,,UNIT

where CHANNEL is a number from 1 through 8 (corresponding to channels A through H) and UNIT is the full address of the Hypertape drive specified in the decrement portion of the first word of its unit control block, as shown in Figure 29.

For example, to permanently assign HH9/1 (channel H, Hypertape drive 9, Data Channel Switch setting 2) as SYSUT2, the following octal alteration card would be used in a job deck similar to the one shown in Figure 28:

```
15160        *OCT        030051000010
```

Changing the Assignment of 1301 Disk Storage

An entry in the auxiliary System Unit Function Table for disk storage is similar to an entry for Hypertape, except that bits 14 through 17 of the decrement specify a module number, rather than a unit number, and bit 11 of the decrement is a one instead ~~of a zero.~~ However, if disk storage is assigned to a system unit function, a matching entry for the function must be made in an auxiliary 19-word Disk Limits Table that immediately follows the auxiliary System Unit Function Table. A symbolic listing of the distributed version of this table is shown in Figure 37. In core storage, the auxiliary Disk Limits Table begins at 15163_8 (corresponding to SYSLB1) and ends at 15205_8 (corresponding to SYSUT4). Note that in Figure 37 the locations corresponding to the SYSCRD (15167_8), SYSPRT (15170_8), and SYSPCH (15171_8) functions are used for purposes unrelated to the Disk Limits Table. However, disk storage would never be assigned to these functions.

An entry in the auxiliary Disk Limits Table has the following format:

```
        PZE        DORG,,DEND
```

where DORG and DEND are the numbers of the first and last tracks, respectively, of the consecutive tracks in the disk storage module (which is specified in the corresponding entry in the auxiliary System Unit Function Table) that are to be assigned to the system unit function.

For example, to permanently assign tracks 0040 through 0079 of disk storage module CD01/0 (channel C, Access 0, Module 1, Data Channel Switch setting 1) as SYSUT3, the following octal alteration cards would be used in a job deck similar to the one shown in Figure 28:

```
15161        *OCT        023101000003
15204        *OCT        000117000050
```

Preparing Duplicate 729 System Library Tapes for Alternate Use by the System Monitor

To reduce delays in processing due to rewinding of the System Library Tape, the complete 7090/7094 IBSYS Operating System can be duplicated on two 729 System Library Tapes that are used alternately by the System Monitor. A second, and alternate, method of reducing tape positioning time is described in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6275. With the alternate method, components of the IBJOB Processor are placed on a second System Library Tape which, in conjunction with a prepositioning feature in the IBJOB Processor, reduces delays in processing due to positioning of the System Library Tape. Heavy users of the IBJOB Processor should consider using the method described in the IBJOB Processor manual as an alternative to the method described in this manual.

When duplicate System Library Tapes are used, one tape is placed on the 729 Magnetic Tape Unit assigned as SYSLB1 and the other is placed on the unit assigned as SYSLB4. The entry in the auxiliary System Unit Function Table for SYSLB4 on both tapes must contain the following:

```
        MZE        CHANNEL,1,UNIT
```

where CHANNEL is the number of the channel (1 through 4 on the distributed preassembled tape) and UNIT is the number of the 729 Magnetic Tape Unit (1 through 10) on that channel to be assigned as SYSLB4. The 1 in the tag portion of the auxilary entry indicates to the System Monitor that SYSLB4 does, in fact, contain a duplicate System Library Tape. The System Monitor will not alternate between duplicate System Library Tapes if the tag portion of the SYSLB4 entry is zero or if either SYSLB1 or SYSLB4 or both are not 729 Magnetic Tape Units.

The sample job deck in Figure 30 may be used to produce two duplicate System Library Tapes, in which the SYSLB4 auxiliary entry on both tapes is changed, as previously described, to allow alternate use of the tapes by the System Monitor. This deck will assign A5 as SYSLB4. It assumes that A3 is assigned initially as SYSUT1. The first of the two duplicate tapes will be produced on A3 (SYSUT1). A3 is then attached as SYSLB2 and is used to produce the second of the two duplicate tapes on a 729 Magnetic Tape Unit that is attached as SYSUT1. The 729 Magnetic Tape Unit specified on the second $ATTACH card can be any avail-

3	4	5 — 8	9 — 11	12	13	14 — 17
Reserve Flag (0)	Channel Type (1 for 7909)	Channel (1-8)	Device (000 for Hypertape) (001 for Disk)	Data Channel Switch (0 or 1)	Access (0)	Unit or Module Numbers (0-9)

Figure 29. Decrement of Entry for Hypertape or Disk in Auxiliary System Unit Function Table

```
1     7        16
-----------------------------------------------------------
$JOB           DUPLICATE TAPES FOR PING PONG
$IBSYS
$IBEDT
      *EDIT
      *MODIFY  IBSYS
15143 *OCT     400005100001
   (END-OF-FILE CARD)
$IBSYS
$ATTACH        A3
$AS            SYSLB2,H
$ATTACH        XK
$AS            SYSUT1,H
$IBEDT
      *EDIT    SYSLB2
   (END-OF-FILE CARD)
$STOP

   NOTE, XK CAN BE ANY AVAILABLE 729 TAPE UNIT
```

Figure 30. Sample Job Deck for Preparing Duplicate System Library Tapes for Alternate Use by the System Monitor

able unit. At the completion of the job, one of the two duplicate System Library Tapes will be on A3 and the other will be on the unit that is specified on the second $ATTACH card.

If the System Monitor is reassembled, as described in the section "Reassembling the System Monitor," alternate use of duplicate System Library Tapes can be specified by the assembly parameter PP EQU CUU. This parameter is described in Figure 33, under the subheading "Miscellaneous Assembly Parameters," and is listed in Figure 34.

Adding 1301 Disk Storage and/or 7340 Hypertape Capability

The distributed 729 capability System Library Tape, together with one of the distributed symbolic tapes, may be used to produce a System Library with 1301 Disk Storage and/or 7340 Hypertape capability, in addition to 729 capability, for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, and the Restart Program. The complete 7090/7094 Operating System can reside on disk. However, if a System Library is produced with Hypertape capability for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, and the Restart Program, these portions of the Operating System may reside on Hypertape, but the remainder of the Operating System, if it is to be used, must reside on 729 magnetic tape. To produce a System Library with disk and/or Hypertape capability, portions of the 7090/7094 IBSYS Operating System on the symbolic tape must be updated (to change assembly parameters), assembled, and then, together with the remainder of the 7090/7094 IBSYS Operating System, edited onto a new System Library Tape. These functions may be performed using a 1301/7340 update-edit job deck that is distributed with the 7090/7094 IBSYS Operating System. This deck is listed in Figure 31. The portions of the 7090/7094 IBSYS Operating System that are updated by the deck are contained on the 7090/7094 IBSYS Operating System Symbolic Tape Number 1. The contents of this tape are as follows:

File 1	System Monitor including the System Core-Storage Dump Program
File 2	IBJOB Monitor
File 3	Full IOCS (Independent IOCS)
File 4	System Editor
File 5	IBJOB Subroutine Library (IBLIB)

Operations Performed by the 1301/7340 Update-Edit Job Deck

Before using the 1301/7340 update-edit deck, the user must prepare the deck (as described later in the text) to ensure a logical configuration of input/output units tailored to the physical limitations and operational requirements of his installation. The following sequence of operations is performed by the update-edit job deck:

1. The portions of the 7090/7094 IBSYS Operating System on Symbolic Tape Number 1 that contain 1301/7340 assembly parameters (with the exception of the IBJOB Subroutine Library) are updated with new assembly parameters in the job deck and are assembled by the FORTRAN II Assembly Program (IBSFAP). Other portions that do not contain 1301/7340 assembly parameters are copied without change onto the update output tape.
2. The IBJOB Subroutine Library is updated, without assembly, onto a separate update output tape so that it can be later assembled and edited by the IBJOB Processor. As it is being updated, the IBJOB Subroutine Library is unblocked, since it is distributed in blocked form (16 cards per record) to minimize the number of symbolic tape reels.
3. The System Editor produces an intermediate System Library Tape by replacing records from the distributed binary System Library Tape with the corresponding updated records that were assembled from the symbolic tape.
4. Tape Unit A3, containing the intermediate System Library Tape is dialed from A3 to A1, thereby assigning the intermediate System Library Tape as SYSLB1.
5. The IBJOB Processor assembles and edits the portions of the IBJOB Subroutine Library containing the updated 1301/7340 assembly parameters.
6. The System Editor produces a new 1301/7340 capability System Library Tape by replacing IBJOB Subroutine Library records from the intermediate System Library Tape with the corresponding updated records that were assembled and edited by the IBJOB Processor.

```
$JOB          UPDATE-EDIT DECK FOR 1301 DISK AND 7340 HYPERTAPE
$IBSYS
$LIST
$ATTACH       A3              PROVIDE UNITS FOR UPDATE OUTPUT.
$AS           SYSCK1,H
$REWIND       SYSCK1
$ATTACH       B4
$AS           SYSCK2,H
$REWIND       SYSCK2
$ATTACH       B2              INSURE SYSPP1, SYSUT2 ARE SAME DENSITY.
$AS           SYSPP1,H
$ATTACH       B3
$AS           SYSUT2,H
$REWIND       SYSPP1
$EXECUTE      IBSFAP
       *FAP                                                   *ALL
        UPDATE  8,9
SYSORG  EQU     2000                                          *BOTH IBB00250
                                   SECTION 1
        END                                                         IBD17600
       *FAP
        UPDATE  ,9,,D
        ENDFIL
        ENDUP
       *FAP
        UPDATE  8,9
CH7909  EQU     1                                             *BOTH 1CJ01000
        END                                                         1CJ08390
       *FAP
        UPDATE  8,9
CH7909  EQU     1                                             *BOTH 1CT00930
        END     -1                                                  1CT07570
       *FAP
        UPDATE  8,9,,D
        END     START                                               1FD20580
        ENDUP
       *FAP
        UPDATE  8,9
                                   SECTION 2
        END     -1              END IOCSB.                          1I030640
       *FAP
        UPDATE  8,9
CH7909  EQU     1                                             *BOTH 1PA01250
        END                                                         1PA05860
       *FAP
        UPDATE  8,9
CH7909  EQU     1                                             *BOTH 1PB00920
        END                                                         1PB24040
       *FAP
        UPDATE  8,9
CH7909  EQU     1                                             *BOTH 1PC01490
TRAIL   EQU     1                                             *BOTH 1PC01640
        END                                                         1PC29120
       *FAP
        UPDATE  ,9,,D
        ENDFIL
        ENDUP
       *FAP
        UPDATE  8,9
                                   SECTION 3
        END     -1                                                  IOF00510
       *FAP
        UPDATE  8,9,,D
        END                                                         IOR15470
        ENDFIL
        ENDUP
       *FAP
        UPDATE  8,9,,D
        END     -1                                                  ED051440
        ENDFIL  9
        UNLOAD  9
        ENDUP
       *FAP
        UPDATE  8,10,U,D
                                   SECTION 4
HYPER   SET     1                  =0 FOR NO HYPERTAPE        *HYPR 3AK00210
DISK    SET     1                  =0 FOR NO DISK             *DISK 3AK00230
HYPER   SET     1                  =0 FOR NO HYPERTAPE        *HYPR 3AL00270
DISK    SET     1                  =0 FOR NO DISK             *DISK 3AL00290
        END                                                         3RN15370
        ENDFIL
        REWIND  10
        UNLOAD  8
        ENDUP
```

Figure 31. Distributed 1301/7340 Update-Edit Deck (Sheet 1 of 2)

```
$IBSYS
$PAUSE          MOUNT SCRATCH TAPES ON A3 AND B3.
$SWITCH         SYSPP1,SYSUT2 PUNCH OUTPUT BECOMES EDIT INPUT.
$IBEDT                        PRODUCE INTERMEDIATE SYSTEM TAPE.
      *EDIT     MAP,MODS
TAPE  *REPLACE IBSYS
TAPE  *REPLACE SYSDMP
TAPE  *REPLACE IBJOB
TAPE  *REPLACE VER003
TAPE  *REPLACE IOCSB
TAPE  *REPLACE IBJOBB
TAPE  *REPLACE IBJOBC
      *AFTER    IBCBC9
TAPE  *INSERT   IBJOBY
      *AFTER    IBFTCH
TAPE  *INSERT   IBJOBI
      *AFTER    IBMAPK
TAPE  *INSERT   IBJOBL
TAPE  *REPLACE IOCS
TAPE  *REPLACE POST
TAPE  *REPLACE PREP
TAPE  *REPLACE IOBB
TAPE  *REPLACE IOBM
TAPE  *REPLACE NOBS
   (END-OF-FILE CARD)
$IBSYS
$*      REMOVE SYSTEM TAPE FROM A1 AND REPLACE WITH A SCRATCH TAPE.
$*      DIAL A1 TO A3 AND A3 TO A1.  CLEAR MACHINE AND LOAD TAPE.
$PAUSE
$JOB            PART  2 OF 1301/7340 UPDATE-EDIT DECK
$LIST
$ATTACH         A1
$AS             SYSLB1,H
$ATTACH         A3
$AS             SYSUT1,H
$ATTACH         A4
$AS             SYSUT3,H
$ATTACH         B1
$AS             SYSOU1,L
$ATTACH         B2
$AS             SYSPP1,H
$ATTACH         B3
$AS             SYSUT2,H
$ATTACH         B4
$AS             SYSCK2,H
$*
$*      NINE TAPE UNITS ARE NECESSARY FOR COMPLETION OF THIS JOB.
$*      IF YOU HAVE NINE TAPE UNITS, LOAD A $TAPE CARD IN THE CARD
$*      READER AND PRESS START.  UNIT A5 WILL THEN BE ATTACHED BY THIS
$*      PROGRAM AS SYSUT4 IN HIGH DENSITY.
$*
$*      IF YOU DO NOT HAVE NINE TAPE UNITS, ATTACH THE ON-LINE CARD
$*      READER AS SYSIN1 AND A2 AS SYSUT4.  THE SECOND STEP PROVIDES A
$*      UNIT TO REPLACE THE UNIT PREVIOUSLY USED AS SYSCK2.  NEXT, LOAD
$*      THE DISTRIBUTED IBJOB LIBRARY (IBLIB) EDITOR DECK IN THE CARD
$*      READER.  THIS DECK IMMEDIATELY FOLLOWS THE NEXT $AS CARD.
$PAUSE
$CARDS
$ATTACH         A5            UNIT USED AS SYSCK2 IS REPLACED.
$AS             SYSUT4,H
$EXECUTE        IBJOB
$IBJOB IBLIB    MAP,LOGIC,NOGO
$EDIT           LOGIC
$REPLACE        .JBCON,ORG=03720
$IEDIT          SYSCK2,SRCH
$IBMAP .JBCON   19
$REPLACE        .IODEF
$IBMAP .IODEF   300,()OK
$REPLACE        .IOCSF
$IBMAP .IOCSF   1300,()OK
$REPLACE        .LOVRY
$IBMAP .LOVRY
$REPLACE        .LXSL
$IBMAP .LXSL
$REPLACE        .IOCS
$IBMAP .IOCS    6700,()OK
   (END-OF-FILE CARD)
$IBSYS
$IBEDT
      *EDIT     MAP,MODS
FILE  *REPLACE CIFSR,SYSUT4
FILE  *REPLACE TIFSR,SYSUT4
   (END-OF-FILE CARD)
$STOP
```

Figure 31. Distributed 1301/7340 Update-Edit Deck (Sheet 2 of 2)

The Assignment and Function of Units for the 1301/7340 Update-Edit Job Deck

As distributed, the 1301/7340 update-edit job deck requires eight 729 Magnetic Tape Units, except near the end of the job, when a ninth unit is required for the assembly and editing of the IBJOB Subroutine Library. However, if only eight 729 Magnetic Tape Units are available, some of the cards from the end of the deck can be placed in the card reader, thereby enabling the card reader to be assigned (near the end of the job) as SYSIN1 in order to release a 729 Magnetic Tape Unit for use in assembling and editing the IBJOB Subroutine Library. The procedure for doing this is described by comments within the job deck itself (Figure 31). If a user has only six or seven 729 Magnetic Tape Units, or five 729 Magnetic Tape Units together with 1301 Disk Storage, he should request his IBM representative to obtain a special procedure from the regional Programming Systems Support Group.

The 1301/7340 update-edit deck is designed for use with the distributed System Library Tape with the input/output units initially assigned to system unit functions, as shown in Figure 26. The assignment and function of the units for the 1301/7340 update-edit job are as shown in Figure 32.

Unit	Initial Assignment	Additional Assignment in this Job	Function In this Job	Update Logical Number	Output
A1	SYSLB1	SYSUT1	System Library ([redacted]) Intermediate		
A2	SYSIN1	None	System Input		
A3	SYSUT1	SYSCK1	Update Output New System Library Tape	9	XX (Two tapes)
A4	SYSUT3	None	Intermediate		
A5	None	SYSUT4	Intermediate		
B1	SYSOU1	None	List Output		
B2	SYSPP1	SYSUT2	Punch Output		X (Optional)
B3	SYSUT2	SYSPP1	Update Input Intermediate	8	
B4	SYSUT4	SYSCK2	Update Output	10	X

Figure 32. The Assignment and Function of Units for the 1301/7340 Update-Edit Deck

The distributed System Library Tape is mounted on A1, the 7090/7094 IBSYS Operating System Symbolic Tape Number 1 is mounted on B3, and the System Input Tape (containing the 1301/7340 update-edit deck) is mounted on A2.

The two tapes written on A3 and the tape written on B4 should be saved. These tapes will be rewound and unloaded at the proper time. They will contain the following:

First Tape on A3: Updated Symbolic Tape containing the System Monitor (including the System Core-Storage Dump Program), the IBJOB Monitor, Full IOCS (Independent IOCS), and the System Editor.

Second Tape on A3: New System Library Tape, in binary form, containing the complete 7090/7094 IBSYS Operating System with 729, 1301, and/or 7340 capability for the System Monitor, the IBJOB Processor, the Input/Output Control System, the Utilities, and the Restart Program.

Tape on B4: Updated symbolic tape containing the IBJOB Subroutine Library (IBLIB), unblocked for assembly by the IBJOB Macro Assembly Program (MAP).

Preparing the 1301/7340 Update-Edit Deck

Before using the distributed update-edit deck, certain control cards and assembly parameter cards already in the deck may have to be removed or replaced, and assembly parameter cards and cards for changing system unit assignment must be inserted in the deck. The insertion of these cards establishes the logical configuration of disk and/or Hypertape units required for a specific installation. Although the distributed job deck is designed primarily for changing the operating characteristics of the system to achieve disk and/or Hypertape capability, System Monitor assembly parameters can, if necessary, be inserted in the deck in order to change (during the same job run) operating characteristics of the system that are not directly related to disk or Hypertape.

Assembly parameters that may be inserted in the update-edit job deck, together with the entries in the auxiliary System Unit Function Table and the auxiliary Disk Limits Table, are listed in Figures 34 through 41 as they appear in the distributed IBSYS Symbolic Tape Number 1. They are listed in the relative order in which they would be placed in the job deck. Assembly parameters which are related to disk capability contain *DISK in columns 67 through 71, and parameters related to Hypertape capability contain *HYPR in columns 67 through 71. Assembly parameters that are not directly related to either disk or Hypertape capability contain blanks in columns 67 through 71. The use of these parameters is optional. Other assembly parameters, which merely indicate whether or not disk and/or Hypertape capability are desired, are already included in the distributed 1301/7340 update-edit deck. Most of these parameters are required for both disk and Hypertape capability and therefore contain *BOTH in columns 67 through 71.

Cards listed in Figures 34 through 41 are inserted in four sections of the 1301/7340 update-edit deck, as indicated in the figure title. These sections are numbered from 1 to 4 and are indicated in the distributed deck by cards containing the word SECTION, followed by the section number. When assembly parameter cards are inserted in the deck, the section number cards should be removed. However, before inserting assembly parameter cards, certain cards already in the deck may be removed or replaced, depending on the requirements of an installation. The following procedure should be followed to prepare the distributed 1301/7340 update-edit deck for use.

Removal and Replacement of Cards in the Distributed Deck

The IBSYS Symbolic Tape Number 1 is distributed in either of two densities, 800 cpi or 556 cpi. The user must verify that the 1301/7340 update-edit deck will attach SYSUT2 and SYSPP1 in the density of the IBSYS Symbolic Tape Number 1. In the distributed deck, high density is specified when SYSUT2 and SYSPP1 are attached. However, it may be necessary to replace the following cards with cards containing an L, instead of an H, in colunm 23:

```
$AS          SYSPP1,H
$AS          SYSUT2,H
```

These cards appear both near the beginning of the deck, following the first $JOB card, and near the beginning of Part 2 of the deck, following the second $JOB card. If it is necessary to replace the cards, they should be replaced at both places.

If the System Library generated by the 1301/7340 update-edit deck is to reside in disk storage exclusively, the variable in the following parameter card in the deck may be changed to save disk storage space:

```
TRAIL        EQU        1              *~~BOTH~~1PC01640
                                         ALL
```

The card in the distributed deck will cause the IBJOBY, IBJOBI, and IBJOBL records of the IBJOB Monitor to be assembled. However, if the System Library is to reside on disk storage exclusively, these records are not required. If the 1 in this card is changed to 0, the records will not be assembled, thereby saving 1301 Disk Storage space. When this change is made, the following cards should be removed from the 1301/7340 update-edit deck:

```
             *AFTER     IBCBC9
TAPE         *INSERT    IBJOBY
             *AFTER     IBFTCH
TAPE         *INSERT    IBJOBI
             *AFTER     IBMAPK
TAPE         *INSERT    IBJOBL
```

If the System Library produced by the 1301/7340 update-edit deck is ever to reside on a 729 Magnetic Tape Unit or 7340 Hypertape Drive, the changes described above should not be made.

Immediately following SECTION 4 in the distributed 1301/7340 update-edit deck are four parameter cards. If only 1301 capability is to be added, the two cards containing *HYPR in columns 67 through 71 should be removed from the deck. If only 7340 capability is to be added, the two cards containing *DISK in columns 67 through 71 should be removed. If both 1301 and 7340 capability are to be added, none of the four cards should be removed from the deck.

Inserting Miscellaneous System Monitor Assembly Parameters

If any changes are to be made to the distributed version of the miscellaneous System Monitor assembly parameters in Figure 34, the appropriate cards should be punched and inserted in SECTION 1 of the 1301/7340 update-edit deck. These parameters are described in Figure 33. For disk and/or Hypertape capability, the SYSORG parameter must be at least 2000_{10}. Therefore, a card with SYSORG equal to 2000_{10} is included in the distributed 1301/7340 update-edit deck. If other miscellaneous System Monitor assembly parameter cards are to be inserted in the deck, they should precede or follow the SYSORG parameter card, depending on their serialization.

Inserting Input/Output Configuration Assembly Parameters

The input/output configuration assembly parameters (shown in Figure 35) that are to be changed are inserted in SECTION 1 of the 1301/7340 update-edit deck, immediately following the miscellaneous System Monitor assembly parameters. These parameters are described in detail in Figure 33. Since the parameters for each channel are similar, only the parameters for channel A are described in Figure 34. The parameters that are changed should be inserted in the deck in the same relative order as they are listed in Figure 35.

Inserting Changes to Entries in the Auxiliary System Unit Function and Disk Limits Tables

Changes to entries in the auxiliary System Unit Function Table (Figure 36) are inserted in SECTION 1 of the 1301/7340 update-edit deck, immediately following the input/output configuration assembly parameters. These changes are followed by any changes to the auxiliary Disk Limits Table (Figure 37).

Normally the unit (A2) assigned as SYSIN1 on the distributed System Library Tape would not be changed at an installation. However, if the SYSIN1 entry in the auxiliary System Unit Function Table is changed (during this job) to permanently assign a unit other than A2 as SYSIN1 or to change the density specification for the entry, then sense switch 1 must be placed DOWN during the operator action pause immediately preceding part 2 of the 1301/7340 update-edit deck (Figure 31) and the following control cards must be placed in the card reader:

```
$ATTACH  A2
$AS      SYSIN1,L
$TAPE
```

Symbol	Code	Variable	Description
			Miscellaneous Assembly Parameters
IBM	EQU	x	The decimal number x specifies the machine type; 709 for IBM 709 and 7090 for IBM 7090/7094. If IBM EQU 709, the instructions that refer to the "Set Density" operations in (NDATA are deleted.
IBSORG	EQU	x	The decimal number x is the origin of the System Nucleus. It cannot be less than 64.
IOXORG	EQU	x	The decimal number x is the origin of IOEX. It cannot be less than the location of the last word of the last unit control block plus one.
SYSORG	EQU	x	The decimal number x is the common subsystem origin. It cannot be less than the last location used by IOEX plus 51. For disk and/or Hypertape capability SYSORG should be 2000.
SYSEND	EQU	x	The decimal number x is the common upper core limits for all subsystems. SYSEND+1 should be the origin of an installation accounting routine if it exists.
HIGHLO	EQU	x	The decimal number x specifies the density to be assigned to a unit if the density specification is omitted from an $AS card. If x is 1, high density is assigned. If x is 0, low density is assigned.
PP	EQU	cuu	This parameter may be used to specify alternate use of duplicate 729 System Library tapes on SYSLB1 and SYSLB4. The decimal digit c is the channel number (1 through 8) and uu is the number of the 729 tape unit (1 through 10) on that channel to be assigned as SYSLB4. If PP EQU 0 is specified, the System Monitor will not alternate between duplicate System Library Tapes.
EJECT	EQU	x	The decimal number x specifies the printer board sense exit (1 through 10) for an eject.
DBLSP	EQU	x	The decimal number x specifies the printer board sense exit (1 through 10) for a double space.
RDUNRT	EQU	x	The decimal number x is the number of recovery attempts IOEX should make on a read redundancy before the redundancy is considered permanent.
ETMODE	EQU	x	If the decimal number x is 0, trap mode will be saved by IOEX. If x is 1, trap mode will remain in effect in IOEX.
			Input/Output Configuration Assembly Parameters
CHA1	EQU	x	The decimal number x is the number of 729 Magnetic Tape Units existing on channel A.
CHAAT	BOOL	xxxxx	The rightmost 13 bits of xxxxx correspond to the card reader, printer, card punch, and 729 Magnetic Tape Units 1 through 10, respectively, beginning with the leftmost of the 13 bits. If a bit is a zero, the corresponding unit is attached to the channel; a one-bit specifies a detached unit.
CHAMD	BOOL	xxxx	The rightmost 10 bits of xxxx correspond to 729 Magnetic Tape Units 1 through 10, respectively, beginning with the leftmost of the 10 bits. If a bit is zero, the model type of the corresponding 729 Magnetic Tape Unit is specified as II or V; a one-bit specifies a IV or VI model type.
PRNTA	EQU	x	If the decimal number x is a 1, the printer is considered to exist on channel A. If no printer exists on channel A, x should be 0.
PNCHA	EQU	x	If a card punch exists on channel A, the decimal number should be 1; otherwise x should be 0.
CDRDA	EQU	x	If a card reader exists on channel A, the decimal number should be 1; otherwise x should be 0. NOTE: A unit control block is generated at initial start for every unit that exists on channel A as specified by CHA1 for 729 Magnetic Tape Units, PRNTA for the printer, PNCHA for the card punch, and CDRDA for the card reader. The parameter CHAAT merely specifies that the unit control block that is generated should indicate that the unit is either attached or detached.

Symbol	Code	Variable	Description
			Input/Output Configuration Assembly Parameters
HTA1	EQU	x	The decimal number x specifies the number of 7340 Hypertape Drives (0 through 10) that exist on Data Channel Switch setting 1 for channel A.
HTAA1	BOOL	xxxx	The rightmost 10 bits of xxxx specify which of the Hypertape drives (specified in HTA1) on Data Channel Switch setting 1 are attached, beginning with the leftmost of the ten bits. If a bit is a zero, the corresponding drive is attached to the channel; a one-bit specifies a detached unit. For example, if there are eight Hypertape drives on channel A, Data Channel Switch setting 1, and only six of the eight are to be attached, the following should be specified: HTA1 EQU 8 HTAA1 BOOL 0017 Although the two units in this example, AH6/0 and AH7/0, will not be attached at initial start, they may be temporarily attached by means of a $ATTACH control card. Units AH8/0 and AH9/0, however, cannot be attached with a $ATTACH card since HTA1 EQU 8 and, therefore, no unit control block will be generated for them at initial start.
HTA2	EQU	x	HTA2 is the same as HTA1 except that it applies to Hypertape drives on Data Channel Switch setting 2.
HTAA2	BOOL	xxxx	HTAA2 is the same as HTAA1 except that it applies to Hypertape drives (specified in HTA2) on Data Channel Switch setting 2.
DFA1	EQU	x	The decimal number x specifies the number of 1301 Disk Storage modules (0 through 10) that exist on Data Channel Switch setting 1 for channel A.
DFA3	EQU	x	The decimal number x specifies the number of access mechanisms per module (1 or 2) for the disk modules (specified in DFA1) on Data Channel Switch Setting 1. or more
NFA1	EQU	x	
DFAA1	BOOL	xxxx	The rightmost ten bits of xxxx specify which of the disk modules (specified in DFA1) on Data Channel Switch setting 1 are attached, beginning with the leftmost of the ten bits. If a bit is zero, the corresponding module is attached; a one-bit specifies a detached module.
IFA1	EQU	x	The decimal number x specifies the 7631 model type for disk modules on Data Channel Switch setting 1. If the model type is III or IV, x is 1; otherwise x is 0.
DFA2	EQU	x	DFA2 is the same as DFA1 except that it applies to disk modules on Data Channel Switch setting 2.
DFA4	EQU	x	DFA4 is the same as DFA3 except that it applies to disk modules (specified in DFA2) on Data Channel Switch setting 2.
NFA2	EQU	x	
DFAA2	BOOL	xxxx	DFAA2 is the same as DFAA1 except that it applies to disk modules (specified in DFA2) on Data Channel Switch setting 2.
IFA2	EQU	x	IFA2 is the same as IFA1 except that it applies to disk modules on Data Channel Switch setting 2.
			System Core-Storage Dump Program Assembly Parameters
ONLIN	BOOL	x	This parameter determines whether dump output will be listed on the System Output Unit only or whether the tag portion of the control word (Figure 5) or console entry keys will determine the output unit. If x is 0, the tag will determine the output unit.
DBLSPC	BOOL	x	If x is 0, dump output will be single-spaced. If x is 1, output will be double spaced.
KEYSWT	EQU	x	The decimal number x determines the sense switch (1-6) that will be tested for the console entry keys option.
FORMAT	EQU	x	The decimal number x determines the output format (1-6) that will be used by the dump program when none is specified. Output formats and their numbers are shown in Figure 6.

Figure 33. Explanation of System Monitor Assembly Parameters

Changing the Assignment of 729 Magnetic Tape Units and Card Equipment: Each entry for 729 Magnetic Tape Units and card equipment in the auxiliary System Unit Function Table has the following format:

```
PFX    CHANNEL,,UNIT
```

where PFX is the density assigned to the system unit function, MZE for high density and PZE for low density. CHANNEL is a number from 1 through 8 (corresponding to channels A through H), which indicates the channel of the unit assigned to the system unit function. UNIT is the number of the 729 Magnetic Tape Unit (1 through 10), card reader (11), card punch (12), or printer (13) assigned to the system unit function. As an example, the following card would be inserted in the 1301/7340 update-edit deck to assign 729 Magnetic Tape Unit B5 as SYSOU2 at low density:

```
SYSOU2    PZE    2,,5    IBB42180
```

Changing the Assignment of 7340 Hypertape Drives: Each entry in the auxiliary System Unit Function Table has the following format:

```
PZE    CHANNEL,,UNIT
```

where CHANNEL is a number from 1 through 8, corresponding to channels A through H, and UNIT is the full address of the Hypertape drive specified in the decrement portion of the first word of its unit control block, as shown in Figure 29. The decimal equivalent of a Hypertape drive address may be specified in the decrement portion of an entry, or a BOOL pseudo-operation may be used to define the address. For example, to assign HH9/1 (channel H, Hypertape drive 9, Data Channel Switch setting 2) as SYSUT2, the following cards would be inserted in the 1301/7340 update-edit deck:

```
H         BOOL   30051    IBB42259
SYSUT2    PZE    8,,H     IBB42260
```

Changing the Assignment of 1301 Disk Storage: An entry in the auxiliary System Unit Function Table for disk storage is similar to an entry for Hypertape, except that bits 14 through 17 of the decrement (Figure 29) specify a module number, rather than a unit number, and bit 11 of the decrement is a 1, instead of a 0. For example, to assign disk storage module CD01/0 (channel C, Access 0, Module 1, Data Channel Switch setting 1) as SYSUT3, the following cards would be inserted in the 1301/7340 update-edit deck:

```
D         BOOL   23101    IBB42269
SYSUT3    PZE    3,,D     IBB42270
```

When a disk storage module is assigned to a system unit function, a matching entry must be made in the 19-word auxiliary Disk Limits Table (Figure 37) which defines the limits of the disk storage area that is to be assigned to the function. An entry in the auxiliary Disk Limits Table has the following format:

```
PZE    DORG,,DEND
```

where DORG and DEND are the numbers of the first and last tracks, respectively, of the consecutive tracks in the disk storage module (which is specified in the corresponding entry in the auxiliary System Unit Function Table) that are to be assigned to the system unit function. For example, if tracks 0040 through 0079 of the disk storage module in the previous example are assigned as SYSUT3, the following card would be inserted in the 1301/7340 update-edit deck:

```
PZE    40,,79    IBB42500
```

Inserting System Core-Storage Dump Assembly Parameters

If any changes are to be made in the distributed version of the assembly parameters for the System Core-Storage Dump Program (shown in Figure 38), the appropriate cards should be punched and inserted in SECTION 1 of the 1301/7340 update-edit deck, immediately following the cards for changing entries in the auxiliary System Unit Function and Disk Limits Tables. The core-storage dump parameters are described in Figure 33.

Inserting IBJOB Monitor Assembly Parameters for Disk and Hypertape Capability

The IBJOB Monitor assembly parameters for disk and Hypertape capability are listed in Figure 39. If only 1301 capability is to be added, the card containing *DISK in columns 67 through 71 should be punched and inserted in SECTION 2 of the 1301/7340 update-edit deck. The variable field of the card should contain a decimal number which indicates the number of 1301 Disk Storage modules that may be used by the Component IOCS in the IBJOB Monitor. If only 7340 capability is to be added, the card containing *HYPR in columns 67 through 71 should be punched to indicate the number of 7340 Hypertape channels to be used by

```
IBM     EQU   7090                                      IBB00130
IBSORG  EQU   64                                        IBB00170
IOXORG  EQU   450                                       IBB00210
SYSORG  EQU   1450                                      IBB00250
SYSEND  EQU   -1                                        IBB00290
HIGHLO  EQU   0                                         IBB00330
PP      EQU   0                                         IBB00370
EJECT   EQU   1                                         IBB00420
DBLSP   EQU   4                                         IBB00450
RDUNRT  EQU   100                                       IBB00480
ETMODE  EQU   0                                         IBB00530
```

Figure 34. Miscellaneous System Monitor Assembly Parameters (SECTION 1)

```
                                   CHANNEL A
CHA1    EQU     10                                                              IBB00580
CHAAT   BOOL    00003                                                           IBB00630
CHAMD   BOOL    01777                                                           IBB00670
PRNTA   EQU     1                                                               IBB00700
PNCHA   EQU     1                                                               IBB00730
CDRDA   EQU     1                                                               IBB00760
HTA1    EQU     0                                                         *HYPR IBB00800
HTAA1   BOOL    1777                                                      *HYPR IBB00820
HTA2    EQU     0                                                         *HYPR IBB00850
HTAA2   BOOL    1777                                                      *HYPR IBB00870
DFA1    EQU     0                                                         *DISK IBB00910
DFA3    EQU     1                                                         *DISK IBB00930
DFAA1   BOOL    1777                                                      *DISK IBB00970
IFA1    EQU     0                                                         *DISK IBB00990
DFA2    EQU     0                                                         *DISK IBB01020
DFA4    EQU     1                                                         *DISK IBB01040
DFAA2   BOOL    1777                                                      *DISK IBB01080
IFA2    EQU     0                                                         *DISK IBB01100
                                   CHANNEL B
CHB1    EQU     10                                                              IBB01150
CHBAT   BOOL    16003                                                           IBB01200
CHBMD   BOOL    01777                                                           IBB01240
PRNTB   EQU     0                                                               IBB01270
PNCHB   EQU     0                                                               IBB01300
CDRDB   EQU     0                                                               IBB01330
HTB1    EQU     0                                                         *HYPR IBB01370
HTBA1   BOOL    1777                                                      *HYPR IBB01390
HTB2    EQU     0                                                         *HYPR IBB01420
HTBA2   BOOL    1777                                                      *HYPR IBB01440
DFB1    EQU     0                                                         *DISK IBB01480
DFB3    EQU     1                                                         *DISK IBB01500
DFBA1   BOOL    1777                                                      *DISK IBB01540
IFB1    EQU     0                                                         *DISK IBB01560
DFB2    EQU     0                                                         *DISK IBB01590
DFB4    EQU     1                                                         *DISK IBB01610
DFBA2   BOOL    1777                                                      *DISK IBB01650
IFB2    EQU     0                                                         *DISK IBB01670
                                   CHANNEL C
CHC1    EQU     6                                                               IBB01720
CHCAT   BOOL    16077                                                           IBB01770
CHCMD   BOOL    1777                                                            IBB01810
PRNTC   EQU     0                                                               IBB01840
PNCHC   EQU     0                                                               IBB01870
CDRDC   EQU     0                                                               IBB01900
HTC1    EQU     0                                                         *HYPR IBB01940
HTCA1   BOOL    1777                                                      *HYPR IBB01960
HTC2    EQU     0                                                         *HYPR IBB01990
HTCA2   BOOL    1777                                                      *HYPR IBB02010
DFC1    EQU     0                                                         *DISK IBB02050
DFC3    EQU     1                                                         *DISK IBB02070
DFCA1   BOOL    1777                                                      *DISK IBB02110
IFC1    EQU     0                                                         *DISK IBB02130
DFC2    EQU     0                                                         *DISK IBB02160
DFC4    EQU     1                                                         *DISK IBB02180
DFCA2   BOOL    1777                                                      *DISK IBB02220
IFC2    EQU     0                                                         *DISK IBB02240
                                   CHANNEL D
CHD1    EQU     6                                                               IBB02290
CHDAT   BOOL    16077                                                           IBB02340
CHDMD   BOOL    1777                                                            IBB02380
PRNTD   EQU     0                                                               IBB02410
PNCHD   EQU     0                                                               IBB02440
CDRDD   EQU     0                                                               IBB02470
HTD1    EQU     0                                                         *HYPR IBB02510
HTDA1   BOOL    1777                                                      *HYPR IBB02530
HTD2    EQU     0                                                         *HYPR IBB02560
HTDA2   BOOL    1777                                                      *HYPR IBB02580
DFD1    EQU     0                                                         *DISK IBB02620
DFD3    EQU     1                                                         *DISK IBB02640
DFDA1   BOOL    1777                                                      *DISK IBB02680
IFD1    EQU     0                                                         *DISK IBB02700
DFD2    EQU     0                                                         *DISK IBB02730
DFD4    EQU     1                                                         *DISK IBB02750
DFDA2   BOOL    1777                                                      *DISK IBB02790
IFD2    EQU     0                                                         *DISK IBB02810
                                   CHANNEL E
CHE1    EQU     0                                                               IBB02860
CHEAT   BOOL    17777                                                           IBB02910
CHEMD   BOOL    1777                                                            IBB02950
PRNTE   EQU     0                                                               IBB02980
PNCHE   EQU     0                                                               IBB03010
CDRDE   EQU     0                                                               IBB03040
HTE1    EQU     0                                                         *HYPR IBB03080
HTEA1   BOOL    1777                                                      *HYPR IBB03100
HTE2    EQU     0                                                         *HYPR IBB03130
HTEA2   BOOL    1777                                                      *HYPR IBB03150
DFE1    EQU     0                                                         *DISK IBB03190
DFE3    EQU     1                                                         *DISK IBB03210
DFEA1   BOOL    1777                                                      *DISK IBB03250
IFE1    EQU     0                                                         *DISK IBB03270
```

Figure 35. Input/Output Configuration System Monitor Assembly Parameters (SECTION 1) (Sheet 1 of 2)

```
DFE3    EQU    1                                          *DISK IBB03210
NFE1    EQU    0                                          *DRUM IBB03230
DFEA1   BOOL   1777                                       *DKDM IBB03250
IFE1    EQU    0                                          *DKDM IBB03270
DFE2    EQU    0                                          *DISK IBB03300
DFE4    EQU    1                                          *DISK IBB03320
NFE2    EQU    0                                          *DRUM IBB03340
DFEA2   BOOL   1777                                       *DKDM IBB03360
IFE2    EQU    0                                          *DKDM IBB03380
                             CHANNEL F
CHF1    EQU    0                                                IBB03430
CHFAT   BOOL   17777                                            IBB03480
CHFMD   BOOL   1777                                             IBB03520
PRNTF   EQU    0                                                IBB03550
PNCHF   EQU    0                                                IBB03580
CDRDF   EQU    0                                                IBB03610
HTF1    EQU    0                                          *HYPR IBB03650
HTFA1   BOOL   1777                                       *HYPR IBB03670
HTF2    EQU    0                                          *HYPR IBB03700
HTFA2   BOOL   1777                                       *HYPR IBB03720
DFF1    EQU    0                                          *DISK IBB03760
DFF3    EQU    1                                          *DISK IBB03780
NFF1    EQU    0                                          *DRUM IBB03800
DFFA1   BOOL   1777                                       *DKDM IBB03820
IFF1    EQU    0                                          *DKDM IBB03840
DFF2    EQU    0                                          *DISK IBB03870
DFF4    EQU    1                                          *DISK IBB03890
NFF2    EQU    0                                          *DRUM IBB03910
DFFA2   BOOL   1777                                       *DKDM IBB03930
IFF2    EQU    0                                          *DKDM IBB03950
                             CHANNEL G
CHG1    EQU    0                                                IBB04000
CHGAT   BOOL   17777                                            IBB04050
CHGMD   BOOL   1777                                             IBB04090
PRNTG   EQU    0                                                IBB04120
PNCHG   EQU    0                                                IBB04150
CDRDG   EQU    0                                                IBB04180
HTG1    EQU    0                                          *HYPR IBB04220
HTGA1   BOOL   1777                                       *HYPR IBB04240
HTG2    EQU    0                                          *HYPR IBB04270
HTGA2   BOOL   1777                                       *HYPR IBB04290
DFG1    EQU    0                                          *DISK IBB04330
DFG3    EQU    1                                          *DISK IBB04350
NFG1    EQU    0                                          *DRUM IBB04370
DFGA1   BOOL   1777                                       *DKDM IBB04390
IFG1    EQU    0                                          *DKDM IBB04410
DFG2    EQU    0                                          *DISK IBB04440
DFG4    EQU    1                                          *DISK IBB04460
NFG2    EQU    0                                          *DRUM IBB04480
DFGA2   BOOL   1777                                       *DKDM IBB04500
IFG2    EQU    0                                          *DKDM IBB04520
                             CHANNEL H
CHH1    EQU    0                                                IBB04570
CHHAT   BOOL   17777                                            IBB04620
CHHMD   BOOL   1777                                             IBB04660
PRNTH   EQU    0                                                IBB04690
PNCHH   EQU    0                                                IBB04720
CDRDH   EQU    0                                                IBB04750
HTH1    EQU    0                                          *HYPR IBB04790
HTHA1   BOOL   1777                                       *HYPR IBB04810
HTH2    EQU    0                                          *HYPR IBB04840
HTHA2   BOOL   1777                                       *HYPR IBB04860
DFH1    EQU    0                                          *DISK IBB04900
DFH3    EQU    1                                          *DISK IBB04920
NFH1    EQU    0                                          *DRUM IBB04940
DFHA1   BOOL   1777                                       *DKDM IBB04960
IFH1    EQU    0                                          *DKDM IBB04980
DFH2    EQU    0                                          *DISK IBB05010
DFH4    EQU    1                                          *DISK IBB05030
NFH2    EQU    0                                          *DRUM IBB05050
DFHA2   BOOL   1777                                       *DKDM IBB05070
IFH2    EQU    0                                          *DKDM IBB05090
```

Figure 37. Input/Output Configuration System Monitor Assembly Parameters (SECTION 1) (Sheet 2 of 2)

```
SYSLB1 MZE    1,,1            HIGH DENSITY A1          IBB42280
SYSLB2 MZE    0               NO UNIT ASSIGNED         IBB42290
SYSLB3 MZE    0               NO UNIT ASSIGNED         IBB42300
SYSLB4 MZE    PPC,PPE,PPU     NO UNIT ASSIGNED         IBB42310
SYSCRD PZE    1,,11           CARD READER ON A         IBB42320
SYSPRT PZE    1,,13           PRINTER ON A             IBB42330
SYSPCH PZE    1,,12           PUNCH ON A               IBB42340
SYSOU1 PZE    2,,1            LOW DENSITY B1           IBB42350
SYSOU2 PZE    2,,1            LOW DENSITY B1           IBB42360
SYSIN1 PZE    1,,2            LOW DENSITY A2           IBB42370
SYSIN2 PZE    1,,2            LOW DENSITY A2           IBB42380
SYSPP1 PZE    2,,2            LOW DENSITY B2           IBB42390
SYSPP2 MZE    2,,4            HIGH DENSITY B4          IBB42400
SYSCK1 MZE    0               NO UNIT ASSIGNED         IBB42410
SYSCK2 MZE    0               NO UNIT ASSIGNED         IBB42420
SYSUT1 MZE    1,,3            HIGH DENSITY A3          IBB42430
SYSUT2 MZE    2,,3            HIGH DENSITY B3          IBB42440
SYSUT3 MZE    1,,4            HIGH DENSITY A4          IBB42450
SYSUT4 MZE    2,,4            HIGH DENSITY B4          IBB42460
```

● Figure 38. Auxiliary System Unit Function Table (SECTION 1)

```
       PZE                                                  IBB42510
       PZE                                                  IBB42520
       PZE                                                  IBB42530
       PZE                                                  IBB42540
DMPSW  PZE    **              DUMP SWITCH                   IBB42550
DMPSV  PZE    **,7,**         DUMP INFO SAVE                IBB42560
IBSAV  PZE    **              S.S.1 SAVE                    IBB42570
       PZE                                                  IBB42580
       PZE                                                  IBB42590
       PZE                                                  IBB42600
       PZE                                                  IBB42610
       PZE                                                  IBB42620
       PZE                                                  IBB42630
       PZE                                                  IBB42640
       PZE                                                  IBB42650
       PZE                                                  IBB42660
       PZE                                                  IBB42670
       PZE                                                  IBB42680
       PZE                                                  IBB42690
```

● Figure 39. Auxiliary Disk/Drum Limits Table (SECTION 1)

```
ONLIN  BOOL   1                                             IBD00210
DBLSPC BOOL   0                                             IBD00220
KEYSWT EQU    4                                             IBD00230
FORMAT EQU    3                                             IBD00240
```

Figure 40. System Core-Storage Dumps Assembly Parameters (SECTION 1)

```
NOCH   EQU    0     NO. OF 1301 OR 7320 MODULES DEFINED IN IOCS  *DKDM 1I000310
NOHYP  EQU    0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 1I000330
```

Figure 41. IBJOB Monitor Assembly Parameters for Disk, Drum, and Hypertape (SECTION 2)

```
NOCH   EQU    0     NO. OF 1301 OR 7320 MODULES DEFINED IN IOCS  *DKDM IOA00360
NOHYP  EQU    0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR IOA00380
```

Figure 42. Full IOCS (Independent IOCS) Assembly Parameters for Disk, Drum, and Hypertape (SECTION 3)

```
NOCH   EQU    0     NO. OF 1301 OR 7320 MODULES DEFINED IN IOCS  *DKDM 3AE00060
NOHYP  EQU    0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3AE00080
NOCH   EQU    0     NO. OF 1301 OR 7320 MODULES DEFINED IN IOCS  *DKDM 3AF00270
NOHYP  EQU    0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3AF00290
NOCH   EQU    0     NO. OF 1301 OR 7320 MODULES DEFINED IN IOCS  *DKDM 3I000300
NOHYP  EQU    0       NO. OF HYPERTAPE CHANNELS DEFINED IN IOCS  *HYPR 3I000320
```

Figure 43. IBJOB Subroutine Library Assembly Parameters for Disk, Drum, and Hypertape (SECTION 4)

Component IOCS and inserted in SECTION 2 of the 1301/7340 update-edit deck. If both 1301 and 7340 capability are to be added, both cards should be prepared and inserted in SECTION 2 of the update-edit deck.

Inserting Full IOCS (Independent IOCS) Assembly Parameters for Disk and Hypertape Capability

The Full IOCS (Independent IOCS) assembly parameters for disk storage and Hypertape capability are listed in Figure 40. These parameters are inserted in SECTION 3 of the 1301/7340 update-edit deck and indicate the number of 1301 Disk Storage modules and/or 7340 Hypertape channels that may be used by the Full IOCS (Independent IOCS). Otherwise, the procedure for preparing and inserting these parameters is the same as for the IBJOB Monitor assembly parameters.

Inserting IBJOB Subroutine Library Assembly Parameters for Disk Storage and Hypertape Capability

The IBJOB Subroutine Library assembly parameters for disk storage and Hypertape capability, shown in Figure 41, indicate the number of 1301 modules and/or 7340 Hypertape channels that may be used by the Library IOCS in the IBJOB Processor. The first four of these parameters are inserted in SECTION 4 of the 1301/7340 update-edit deck. The last two parameters are inserted, as indicated by their serial numbers, between the fourth and fifth cards following SECTION 4 of the update-edit deck. Otherwise, the procedure for preparing and inserting these parameters is the same as for the IBJOB Monitor and Full IOCS.

Operating Procedure for the 1301/7340 Update-Edit Deck

After the 1301/7340 update-edit deck has been prepared, as described in the preceding section, the following procedure should be followed to generate a new System Library Tape with 1301 and/or 7340 capability.

1. Place the prepared 1301/7340 update-edit deck in the System Input File on SYSIN1 (A2).
2. Mount the distributed 729 capability System Library Tape on SYSLB1 (A1).
3. Mount the distributed IBSYS Symbolic Tape Number 1 on B3.
4. Mount work tapes on B1, B2, B4, A3, A4, and, if it exists at the installation, A5.
5. Follow the initial start procedure described in the publication *IBM 7090/7094 IBSYS Operating System: Operator's Guide*, Form C28-6355.
6. Follow the operating instructions printed on the System Printer.

The output tapes produced during the job run are described in the section "The Assignment and Function of Units for the 1301/7340 Update-Edit Job Deck." Refer to the "System Editor" section of the manual for information on editing the new System Library onto disk storage or Hypertape.

Reassembling the System Monitor

Although reassembly of the System Monitor is normally not required unless 1301 Disk Storage and/or 7340 Hypertape capability is to be added, it may be necessary at some installations to add an installation accounting routine or otherwise tailor the System Library to special installation requirements. The sample job deck in Figure 42 may be used for this purpose. Before the deck is used, any symbolic cards for inserting or changing coding, changing assembly parameters, and/or changing system unit assignments must be inserted in the deck immediately following the

```
1     7        16
------------------------------------------------------------------------
$JOB           UPDATE AND ASSEMBLE SYSTEM MONITOR
$ATTACH        A4            PROVIDE UNIT FOR UPDATE OUTPUT TAPE.
$AS            SYSCK1
$ATTACH        B2            INSURE THAT SYSPP1, SYSUT2 ARE SAME DENSITY
$AS            SYSPP1
$ATTACH        B3
$AS            SYSUT2
$SWITCH        SYSUT1,SYSUT3 RESERVE USUAL UNIT FOR NEW SYSTEM TAPE.
$REWIND        SYSPP1
$EXECUTE       IBSFAP
      *FAP
       UPDATE  8,9
   (INSERT ASSEMBLY PARAMETER CARDS,FIGURES 29 THROUGH 33)
       END                                                      IBD17640
       ENDFIL
$IBSYS
$SWITCH        SYSPP1,SYSUT2
$SWITCH        SYSUT1,SYSUT3 RESTORE UNITS TO ORIGINAL ASSIGNMENTS.
$REMOVE        SYSCK1
$IBEDT
      *EDIT    MAP,MODS
TAPE  *REPLACE IBSYS
TAPE  *REPLACE SYSDMP
   (END-OF-FILE CARD)
$STOP
```

Figure 42. Sample Job Deck for Updating and Assembling the System Monitor

UPDATE pseudo-operation card. The assembly parameter and system unit assignment cards should be inserted in the order in which they appear in Figures 34 through 38. The System Monitor assembly parameters are described in Figure 33. The format of the entries in the auxiliary System Unit Function Table and auxiliary Disk Limits Table are described in the preceding text, in the section "Inserting Changes to Entries in the Auxiliary System Unit Function and Disk Limits Tables."

For the sample job deck, the symbolic tape containing the System Monitor is mounted on B3 and the distributed System Library Tape is mounted on A1. The deck produces an updated symbolic tape containing the System Monitor on B4 and the new System Library Tape on A3.

Incorporating a User-Designed Installation Accounting Routine

A user may, if he wishes, design an installation accounting routine tailored to his requirements and incorporate it in the System Library as part of the 7090/7094 IBSYS Operating System. The accounting routine may be designed to perform a variety of functions, such as job timing, job billing, or producing statistical data on the processing of jobs. Since accounting practices vary considerably from installation to installation, a specific accounting routine is not provided with the 7090/7094 IBSYS Operating System. However, facilities are provided for incorporating into the system an installation accounting routine specifically designed by the user to meet his own requirements.

There are three one-word entries in the Communications Region of the System Nucleus that are used specifically for accounting purposes: SYSIDR, SYSACC, and SYSPID. The function of each of these entries is described in detail in Appendix A.

An installation accounting routine may either be incorporated as part of the System Monitor and be available for use by the System Monitor and each of the subsystems, or it may be incorporated as part of a subsystem where it is available for use by the subsystem only. Both methods of incorporating an accounting routine are described in the following text.

Incorporating an Accounting Routine in the System Monitor

If an installation accounting routine is incorporated in the System Monitor, it must be loaded in the last (high-numbered) location of core storage and must not occupy more than 500 locations.

Coding the Accounting Routine

To incorporate the coding for the accounting routine, the System Monitor must be updated and reassembled, as described in the section "Reassembling the System Monitor." A specific place within the System Supervisor (IBSUP) coding on the symbolic tape has been selected for incorporating the coding of the accounting routine itself. The card serialization numbers from IBB66620 through IBB99990 have been reserved for this purpose. On the distributed symbolic tape, the card with the serialization IBB66620 contains the following pseudo-operation:

```
        TCD        B3ORG                IBB66620
```

This pseudo-operation should either be deleted or be replaced with the first card of the installation accounting routine. The last card of the installation accounting routine must contain the same pseudo-operation, that is:

```
        TCD        B3ORG                xxxxxxxx
```

where xxxxxxxx is the last card serialization number used for the accounting routine.

The accounting routine may also be placed at any other appropriate place within the System Supervisor (IBSUP) coding, except between the symbols IBSNUC and NUCEND (the System Nucleus is derived, at initial start, from the coding between these symbols). However, if it is inserted in a place other than the selected one, reserialization of the existing coding will normally be required.

If the installation accounting routine is to be loaded into upper core storage each time the System Supervisor is loaded, then the coding for the routine should "origin" in the upper core storage area where the assembled routine is to reside. However, if it is desired to have the accounting routine loaded into upper core storage only at initial start, then the coding for the routine must not origin in upper core storage and additional coding must be placed in the System Monitor "COLD" routine to relocate the routine into upper core storage at initial start.

Changing the Assembly Parameter SYSEND

The System Monitor assembly parameter SYSEND (Figures 33 and 34) must be equated to the first location of the installation accounting routine in upper core storage, minus one. The SYSEND parameter defines the address that is loaded into the address portion of the SYSCOR location in the Communication Region of the System Nucleus (Appendix A). This address indicates to the subsystems the address of the last core storage location available for their use. Therefore, the area of core storage beginning at SYSEND+1 and ending at 77777_8 will not be disturbed by any of the subsystems. The SYSEND parameter also indicates to the System

Supervisor the area of core storage in which to load the off-line print routine SPOUT. The System Supervisor loads SPOUT in the 200 locations beginning at SYSEND-199 and ending at SYSEND.

Changing the Location of IBSIDR

Location IBSIDR, in the System Monitor, contains a transfer instruction that is placed, at initial start, into location SYSIDR in the Communication Region of the System Nucleus (Appendix A). The coding for location IBSIDR on the distributed symbolic tape is as follows:

```
IBSIDR     TRA     2,4          IBB41810
```

When the System Monitor is reassembled, this coding must be changed so that the TRA instruction transfers to the beginning of the installation accounting routine, thereby enabling the accounting routine to gain control whenever the System Monitor or a subsystem transfers control to SYSIDR after reading a $JOB or $ID card or whenever a subsystem signs on or off.

Changing the Contents of Location IBSACC

The contents of location IBSACC, in the System Monitor, are placed, at initial start, into location SYSACC in the Communication Region of the System Nucleus (Appendix A). The coding for location IBSACC on the distributed symbolic tape is as follows:

```
IBSACC     PZE                  IBB41840
```

If it is desired to make the installation accounting routine responsible for printing the $JOB and $ID cards, instead of having them printed both off-line and on-line by the subsystem (or the System Monitor) that reads the card, then the coding for the IBSACC location must be changed when the System Monitor is reassembled, so that it does not contain all zeros.

Incorporating an Accounting Routine in a Subsystem

If an accounting routine is designed for use only with a specific subsystem, then the accounting routine should be assembled and then appended to the first record of the subsystem by using an *MODIFY card when editing the System Library (refer to the section "System Editor"). This will result in the accounting routine being loaded whenever the first record of the subsystem is loaded after a $EXECUTE card containing the name of the subsytem is read. The first record of the subsystem should also be appended to overlay the SYSIDR location in the Communication Region of the System Nucleus so that the location contains a transfer to the beginning of the installation accounting routine. If necessary, the SYSCOR location should be overlaid in order to change the address portion of the entry which defines the end of usable core storage. The SYSACC location may also be overlaid if it is desired to change it to nonzero.

A $IBSYS card, followed by a $RESTORE card, should be placed at the end of a job or job segment performed by a subsystem containing an accounting routine if it is followed on the system input file by jobs or job segments to be performed by others subsystems or by the System Monitor. The $RESTORE card will restore SYSIDR, SYSCOR, and SYSACC locations to their original state, thereby nullifying any changes made by the subsystem containing the accounting routine.

Incorporating User Programs as Subsystems Under System Monitor Control

A user may, if he chooses, design a program and insert it in the System Library. The program can then be called into core storage by a $EXECUTE card and can be executed. Once the program is coded and assembled, it may be inserted in the System Library using a job deck similar to the sample job deck in Figure 43.

```
1     7        16
-------------------------------------------------------
$JOB           INSERT SYSTMU AS SUBSYSTEM AFTER IBJOB
$IBSYS
$IBEDT
      *EDIT    MAP,MODS
      *PLACE   SYSTMU,1,1,2
FILE  *AFTER   IBJOB
TAPE  *INSERT
   (COLUMN BINARY CARD IMAGES ON SYSUT2)
      *INSERT  FILEMK
   (END-OF-FILE CARD)
$STOP
```

Figure 43. Sample Job Deck for Inserting a User Program as a Subsystem Under System Monitor Control

For the sample job deck, it is assumed that a user program, which is named SYSTMU, is located on SYSUT2 in the form of absolute column-binary card images that terminate with a transfer card. The job deck is designed to insert the program in the System Library immediately following the IBJOB Processor. Before the program is inserted, it is converted by the System Editor into a self-loading, scatter-load format, which is a standard format for the System Library. For the sample job deck in Figure 43, it is assumed that the user program consists of one record only. However, a program may consist of more than one record. For each additional record, another TAPE *INSERT card must be placed in the job deck and the absolute column-binary card images from which the record is formed must follow (on SYSUT2) the card images for the previous record. The column-binary card images for each record must end with a transfer card. Additional information may be found in the section "System Editor."

In designing and coding a program for insertion as a subsytem in the System Library, a number of rules must be adhered to in order to ensure proper loading of the program, coordinated control of input/output operations, and continuous job processing. These rules are as follows:

1. The program must use the core storage between SYSORG and SYSEND only (refer to SYSCOR in Appendix A).

2. The first word in the first record of the program must be a BCD name without leading blanks and may have an origin at SYSCUR (Appendix A). This name is the name of the program and is the name specified on the $EXECUTE card to call the program. The name must not be the same as the name of any other record in the System Library.

3. The first word of the second and succeeding records, if any, of the program must be a unique BCD name, without leading blanks, and may have an origin at SYSCUR.

4. The first record must contain a TRA instruction, which transfers control to the beginning of the program and which has an origin at SYSTRA (Appendix A).

5. The System Loader (SYSLDR) in the System Nucleus may be used by a program to load the second and succeeding records of the program. In Appendix A, a description of the use of the System Loader may be found under "SYSLDR." If the System Loader is used to load a record, the contents of SYSTRA must be changed to transfer control to the first instruction that is to be executed after the record is loaded. This may be accomplished by placing in the record itself a TRA instruction which has an origin at SYSTRA. The instruction will then overlay the content of SYSTRA when it is loaded.

6. The program must recognize and act upon the $IBSYS, $EXECUTE, $STOP, $ID, and $JOB cards, as described in the section "System Nucleus" under the heading "Job Control Communications with Subsystems."

7. If a system unit is to be used, it must be referred to by way of its entry in the System Unit Function Table and, if it is disk, by its entry in the Disk Limits Table (refer to the section "System Nucleus").

8. If a unit that is not assigned to a system unit function is to be used, it must be referred to by way of an availability chain (refer to the section "System Nucleus").

9. The program must either use IOEX (described in the section "Input/Output Executor") or adhere to the following rules:

   a. If a 729 Magnetic Tape Unit or 7340 Hypertape Drive is used, the program must keep track of the file and record count in the unit control block of the unit.

   b. Before using the System Loader, the program must make certain that there is no activity on the channel that is being used.

   c. If the System Loader is used to load records from disk storage, the program must not overlay IOEX routines (FDAMT, (DECVD, and (DECRQ.

Incorporating IBM Modifications to the 7090/7094 IBSYS Operating System

The 7090/7094 IBSYS Operating System is distributed preassembled in binary form on a single 729 capability System Library Tape and in symbolic form on several tapes. Each of the symbolic tapes will normally contain more than one subsystem. However, since the System Monitor and each of the subsystems are maintained separately, instructions will be provided, when the symbolic tapes are distributed, for copying the System Monitor and each subsystem onto separate tapes. Whenever modifications are to be made to the System Monitor or a subsystem, a modification letter will be distributed. Two job decks will be distributed with the letter; a symbolic modification deck and a binary modification deck. Either deck may be used to produce a new binary System Library with the required modifications incorporated. However, the symbolic modification deck will also produce an updated symbolic tape of the subsystem (or the System Monitor) being modified and a symbolic assembly listing of the portions of the subsystem that are updated and reassembled to incorporate the modifications. In addition to the symbolic and binary modification decks, a third job deck, the cumulative binary modification deck, will be distributed whenever a new version of the 7090/7094 IBSYS Operating System is released. This deck may be used to accumulate modification cards from several binary modification decks and may be used, at any time, to bring a System Library up to the latest modification level in a single job run.

Any one or a combination of the three methods represented by the three job decks may be used at an installation to maintain the System Library, depending upon the particular requirements of the installation.

Symbolic Modification Deck

The control cards in a standard symbolic modification deck are listed in Figure 44. In addition to these cards, each distributed deck will contain symbolic modification cards for updating a subsystem or the System Monitor.

The symbolic modification deck is designed for use with a System Library having the same system unit assignments (Figure 26) as the distributed System Library Tape. If system unit assignments have been changed at an installation, it may be necessary to change one or more of the three $ATTACH cards in the beginning of the deck.

```
$JOB            7090/7094 IOCS, 7090-IO-919, VERSION 5, MODIFICATION 1
$ATTACH         A4              PROVIDE UNIT FOR UPDATE OUTPUT TAPE.
$AS             SYSCK1
$ATTACH         B2              INSURE THAT SYSPP1, SYSUT2 ARE SAME DENSITY
$AS             SYSPP1
$ATTACH         B3
$AS             SYSUT2
$SWITCH         SYSUT1,SYSUT3 RESERVE USUAL UNIT FOR NEW SYSTEM TAPE.
$REWIND         SYSPP1
$EXECUTE        IBSFAP
       *FAP
        UPDATE  8,9,,D          SPACE TO BEGINNING OF PROPER ASSEMBLY.
        END                                                         XXXXXXXX
        ENDUP
       *FAP
        UPDATE  8,9             UPDATE-ASSEMBLE RECORD TO BE MODIFIED.
        (UPDATE CARDS)                                              XXXXXXXX
        END     -1              PRODUCE TRANSFER CARD FOR EDITING   XXXXXXXX
       *FAP
        UPDATE  8,9,,D          COPY REST OF SYMBOLIC TAPE.
        END                     MATCH SERIALIZATION FOR LAST END CARD.  XXXXXXXX
        ENDFIL
        ENDUP
$IBSYS
$SWITCH         SYSPP1,SYSUT2
$SWITCH         SYSUT1,SYSUT3 RESTORE UNITS TO ORIGINAL ASSIGNMENTS.
$REMOVE         SYSCK1
$IBEDT
       *EDIT    MAP,MODS
TAPE   *REPLACE (RECORD NAME)                                       AABCCDDD
   (END-OF-FILE CARD)
$IBSYS
$SWITCH         SYSPP1,SYSUT2
$STOP
```

Figure 44. Symbolic Modification Deck

The assignment and the function of tape units used by the symbolic modification deck are shown in Figure 45.

The old System Library Tape is mounted on SYSLB1 (A1); the symbolic tape, containing the one subsystem (or the System Monitor) that is to be modified, is mounted on SYSUT2 (B3). The density assigned to SYSUT2 and SYSPP1 must be the same as the density of the symbolic tape. The density assigned to a system unit function is determined by the System Monitor assembly parameter HIGHLO (Figures 33 and 34), unless a different density is specified on a $AS card that is used to assign a unit to the system unit function. If the density specified by the HIGHLO parameter does not agree with the density of the symbolic tape on SYSUT2, the correct density should be specified on the $AS SYSUT2 and $AS SYSPP1 cards.

At the completion of the symbolic modification job, the new System Library Tape is on tape unit A3, the updated symbolic tape for the subsystem (or the System Monitor) is on tape unit A4, and the assembly listing of the updated and reassembled portions of the subsystem is on tape unit B1.

The following operations are performed by the symbolic modification deck:

1. Tape unit A4 is attached as SYSCK1, to function as the update output tape.

2. Tape units B2 and B3 are attached as SYSPP1 and SYSUT2, respectively. This is done to ensure that both units are set to the same density and that their density settings do not change when their system unit function assignments are switched later in the job.

Unit	Initial Assignment Function	Additional Assignments in this Job	Function in this Job	Update Logical Number	Output
A1	SYSLB1	none	Old System Library		
A2	SYSIN1	none	System Input		
A3	SYSUT1	SYSUT3	New System Library		X
A4	SYSUT3	SYSCK1 SYSUT1	Updated Symbolic	9	X
B1	SYSOU1	none	List Output		X
B2	SYSPP1	SYSUT2	Punch Output		X (Optional)
B3	SUSUT2	SYSPP1	Old Symbolic	8	
B4	SYSUT4	none	Intermediate		

Figure 45. The Assignment and Function of Units for the Symbolic Modification Decks

3. The two functions, SYSUT1 and SYSUT3, are switched. In addition to the normal list output on SYSOU1, the job deck produces a new System Library Tape and an updated symbolic tape of the subsystem being modified. Switching SYSUT1 and SYSUT3, both before and after the update and assemble portion of the

job, causes the System Library Tape to be written on a different physical tape unit than the one on which the updated symbolic tape is written.

4. The System Peripheral Punch Tape is rewound.

5. The symbolic tape of the subsystem, at the previous modification level, is copied from SYSUT2 onto SYSCK1 (without assembly) up to the portion that requires modification.

6. The portion of the subsystem on the symbolic tape on SYSUT2 that requires modification is updated with the symbolic modification cards in the job deck on SYSIN1 and is written on SYSCK1. The updated portion is assembled and then written in binary form on SYSPP1. The assembly listing is placed on SYSOU1.

7. The remainder of the old symbolic tape on SYSUT2 is copied (without assembly) onto SYSCK1, after which an end-of-file mark is written.

8. SYSPP1, which contains the updated and assembled portions of the subsystem, is switched with SYSUT2, in preparation for editing onto the new System Library Tape. The old symbolic tape is on the unit that is now assigned as SYSPP1, and may be saved.

9. SYSUT1 and SYSUT3 are switched again (described in step 3). This restores their original assignments.

10. The updated symbolic tape on SYSCK1 is rewound and unloaded.

11. The System Library is edited to incorporate the updated and reassembled portions of the subsystem that is in binary form on SYSUT2. The following System Editor control card is normally used:

```
TAPE        *REPLACE (Record Name)
```

In some cases, the following System Editor control card may also be used:

```
TAPE        *MODIFY  (Record Name)
```

Binary Modification Deck

The basic control cards in a standard binary modification deck are listed in Figure 46. In addition to these cards, each distributed deck will contain one or more System Editor *MODIFY and/or *REPLACE cards and a number of column-binary alteration cards (Figure 17) for modifying a subsystem or the System Monitor. This deck can be used without change to produce a new System Library Tape from the System Library at the previous modification level. An *CHECK card is included in the deck to verify that the correct number of *MODIFY, *REPLACE, and column-binary alteration cards have been received.

The *MODIFY, *REPLACE, and column-binary alteration cards in the deck are serialized in columns 73 through 80. The serialization indicates the order, in the System Library, of the record being modified and has the following format:

AABCCDDD

where AA is the order in the System Library of the subsystem (or System Monitor) being modified, multiplied by 10. For example, the System Monitor is 10, the IBJOB Processor is 20, the Generalized Sorting System is 30, etc. B is the number of a file within the subsystem, CC is the number of a record within the file, and DDD is the number of a column-binary alteration card or an *MODIFY or *REPLACE card. For the *MODIFY and *REPLACE cards, DDD is always 000. As an example, the cards for modifying the first record (IBMAPJ) of the fourth file of the IBJOB Processor are serialized as follows:

```
*MODIFY  IBMAPJ              20401000
(First alteration card)      20401001
(Second alteration card)     20401002
(Third alteration card)      20401003
```

```
$JOB                                                                          00000001
$* IBJOB FORTRAN IV COMPILER (IBFTC)      7090-FO-805, VERSION 3, MOD. 2     00000008
$IBEDT                                                                        00000030
      *EDIT    MAP,MODS                                                       00000031
       (MODIFICATION CARDS)                                                   AABCCDDD
      *CHECK   N           VERIFY THAT N CARDS HAVE BEEN RECEIVED.            98999999
   (END-OF-FILE CARD)
$STOP                                                                         99000001
```

Figure 46. Binary Modification Deck

The same type of serialization is used for the maintenance control cards in the System Editor portion of the symbolic modification deck shown in Figure 44.

Cumulative Binary Modification Deck

The basic control cards in the cumulative binary modification deck are listed in Figure 47. This deck is distributed whenever a new version of the 7090/7094 IBSYS Operating System is released. After each binary modification deck is received, modification cards from the deck may be removed and placed in the cumulative binary modification deck. Instructions for doing this will be provided in each modification letter. The cumulative deck may be used to produce a System Library Tape at the current modification level from a backup System Library Tape. Use of the cumulative deck facilitates the determination of System Library modification levels, since the modification cards for the System Monitor and all subsystems are included in one deck.

Occasionally, it may be necessary to perform a special edit to modify library subroutines, using editing facilities in the IBJOB Processor, the FORTRAN II Processor, or the Commercial Translator Processor. When this is necessary, a special modification deck is distributed. However, the cards from this special deck should not be incorporated in the cumulative binary modification deck. The user may keep a copy of the distributed System Library Tape as a backup tape and use it and the cumulative binary modification deck whenever it is necessary to produce a System Library Tape at the current modification level. If a special modification deck is distributed, it may be used to produce a new backup System Library Tape from the old one. The new backup tape and the cumulative binary modification deck may then be used, when necessary, to produce an up-to-date System Library Tape.

Maintaining a Two-Tape System Library

Some installations may use a two-tape System Library, in which parts of the IBJOB Processor are located on a second System Library Tape. To incorporate IBM modifications without changing the distributed modification decks, a duplicate of the System Library should be maintained on a single tape reel. After modifications are incorporated in the single System Library Tape, it may be used to produce a two-tape System Library, in which parts of the IBJOB Processor are located on a second tape. The procedure for dividing the IBJOB Processor is described in the publication *IBM 7090/7094 IBSYS Operating System: IBJOB Processor*, Form C28-6275.

```
$JOB           7090/7094 IBSYS, 7090-PR-130, CUMULATIVE EDITOR DECK        00000001
$*THIS EDITOR DECK WILL GENERATE A SYSTEM TAPE CONTAINING -                00000002
$* IBSYS SYSTEM MONITOR                  7090-SV-918, VERSION 4, MOD. 0 00000003
$* IBJOB MONITOR (IBJOB)                 7090-SV-801, VERSION 3, MOD. 0 00000004
$* IBJOB LOADER (IBLDR)                  7090-SV-802, VERSION 3, MOD. 0 00000005
$* IBJOB SUBROUTINE LIBRARY (IBLIB)      7090-LM-803, VERSION 3, MOD. 0 00000006
$* IBJOB MACRO ASSEMBLY PROGRAM (IBMAP)  7090-SP-804, VERSION 3, MOD. 0 00000007
$* IBJOB FORTRAN IV COMPILER (IBFTC)     7090-FO-805, VERSION 3, MOD. 0 00000008
$* IBJOB COBOL COMPILER (IBCBC)          7090-CB-806, VERSION 3, MOD. 0 00000009
$* GENERALIZED SORTING SYSTEM            7090-SM-922, VERSION 6, MOD. 0 00000010
$* FORTRAN II PROCESSOR                  7090-FO-928, VERSION 3, MOD.18 00000011
$* UTILITIES                             7090-UT-927, VERSION 2, MOD. 0 00000012
$* RESTART PROGRAM                       7090-IO-976, VERSION 1, MOD. 0 00000013
$* COMMERCIAL TRANSLATOR PROCESSOR       7090-CT-921, VERSION 5, MOD. 0 00000014
$* 9PAC PROCESSOR                        7090-PR-924, VERSION 4, MOD. 0 00000015
$* INPUT/OUTPUT CONTROL SYSTEM           7090-IO-919, VERSION 6, MOD. 0 00000016
$IBEDT                                                                  00000030
      *EDIT     MAP,MODS                                                00000031
       (MODIFICATION CARDS)                                             AABCCDDD
   (END-OF-FILE CARD)
$STOP                                                                   99000001
```

Figure 47. Cumulative Binary Modification Deck

Appendix A: System Nucleus Communication Region Entries

The functions of the entries in the Communication Region of the System Nucleus are described in this appendix.

SYSTRA is loaded with a transfer instruction when the first record of a subsystem is scatter-loaded into core storage following the reading of a $EXECUTE card. After the loading of the first record of a subsystem is completed, the System Loader transfers control to this entry, which, in turn, transfers control to the beginning of the subsystem. A transfer instruction may also be loaded in this location when succeeding records of a subsystem are loaded.

SYSDAT contains a six-character word containing the date specified on the last $DATE card processed. The entry should be updated by a $DATE card at the beginning of each day. The date word is provided for use in headings and labels by subsystems and object programs.

SYSCUR contains the name, in BCD form, of the subsystem or subsystem record currently in core storage. The System Supervisor places the subsystem name from the $EXECUTE card in this location before loading the first record of the subsystem. When succeeding records of a subsystem are loaded, the name of the record may be loaded into this location.

SYSRET is the location to which each subsystem transfers control to call in the System Supervisor. Once the System Supervisor is called into core storage, the instruction EMTM is executed. Therefore, the machine is always in the multiple-tag mode when control is passed to a subsystem from the System Supervisor.

SYSKEY is the location in which the entry key settings are stored at initial start. This location may be interrogated to determine what the status of the entry keys was before initial start.

SYSSWS is the location in which sense switch settings are stored at initial start. Sense switches 1 through 4 are reserved for use by the Operating System. This location may be interrogated to determine what the status of the sense switches was before initial start. Bits 30 through 35 of the entry represent sense switches 6 through 1, respectively. A 1-bit indicates that the corresponding sense switch was down at initial start.

SYSPOS contains the number of the System Library Unit and the position on the unit of the subsystem currently in core storage. This information is entered in the SYSPOS location by the System Supervisor after it looks up, in the System Name or System Loader Table, the position of a subsystem specified on the $EXECUTE card. When the System Library is on tape, the format of the entry is as follows:

PZE INDEX,,NFILES

where INDEX is 1, 2, 3, or 4, corresponding to SYSLB1, 2, 3, or 4, and NFILES is the number of files the System Supervisor must skip over before loading the first record of the subsystem.

When the System Library is on disk, the format of the SYSPOS entry is as follows:

PZE INDEX,,L(SYSNAM)

where INDEX is the same as previously described and L(SYSNAM) is the binary disk track location of the subsystem.

107 *SYSUNI* contains the first location (in the address portion) and length (in the decrement portion) of the System Unit Function Table. The Disk Limits Table is assembled in the core-storage locations immediately following the System Unit Function Table.

110 *SYSUBC* contains the first location (in the address portion) and length (in the decrement portion) of the Unit Control Block Table. This table consists of one word for each channel containing the following information:

Prefix	Number of card units assigned to the channel.
Decrement	Total number of units assigned to the channel.
Address	Address of the first unit control block for the channel.

111 *SYSUAV* contains the first location (in the address portion) and length (in the decrement portion) of the Unit Availability Table. This table is described in the section "System Nucleus" under the heading "Unit Availability Table."

112 *SYSUCW* contains the first location (in the address portion) and the combined length (in the decrement) of all unit control blocks. The unit control blocks are assembled in contiguous locations.

13 *SYSRPT* contains a transfer instruction to a System Nucleus routine that detemines whether the System Supervisor or the subsystem receives control when a $JOB card is read by a subsystem. When a subsystem reads a $JOB card, it transfers to SYSRPT. If the sign of SYSJOB is minus, indicating that restoration of unit assignments is required, or if sense switch 1 is down and the card reader is not assigned as the System Input Unit, indicating that a between-jobs interrupt

Form C28-6248-2
Page Revised 9/10/64
by TNL N28-0118

condition exists, control is passed to the System Supervisor. Otherwise, control is returned to the subsystem that read the $JOB card.

SYSCEM normally contains the following instruction:

```
TRA        SYSTRA−2
```

This location is reserved for use by the customer engineer. During machine maintenance periods, it may contain a transfer to a customer engineering diagnostic routine located in core storage between SYSORG−50 and SYSORG−1. The diagnostic routine is transferred to this area by the System Supervisor at initial start and when a $RESTORE card is processed.

SYSDMP contains a transfer to a bootstrap routine for loading the System Core-Storage Dump Program, which is part of the System Monitor (IBSYS) file on SYSLB1. A transfer to SYSDMP initiates a core-storage dump in accordance with the options selected by the programmer or operator. (Additional information is contained in the section "System Core-Storage Dump Program.") The dump spill tape unit is SYSPP2. Neither SYSOU1 nor SYSPP2 can be disk or drum.

SYSIOX contains the first location (in the address portion) and length (in the decrement portion) of the IOEX Communication Table (Figure 16). This table contains entries for transferring control to IOEX subroutines.

SYSIDR is provided for transferring control to an installation accounting routine. Whenever a $ID or $JOB card is processed by a subsystem or the System Monitor, a transfer is made to SYSIDR, as follows:

```
TSX        SYSIDR,4
PZE        L($ID)
      return
```

where L($ID) is the location of the first word of the buffer containing the $ID or $JOB card in BCD form. In the distributed version of the System Monitor, SYSIDR contains:

```
TRA        2,4
```

Therefore, control is returned immediately to the subsystem (or to the System Monitor) that processed the $ID or $JOB card. If an installation accounting routine exists at an installation, SYSIDR should contain a transfer to the routine.

SYSCOR contains the limits of the core-storage area available for use by subsystems operating under control of the System Monitor.

```
PZE             SYSEND,,SYSORG
```

In the distributed version of the System Monitor, SYSEND and SYSORG are defined as follows:

SYSORG = 2652_8 or 1450_{10}
SYSEND = 77777_8 or $32{,}767_{10}$

The FORTRAN II Processor does not refer to location SYSCOR when defining the two symbols SYSORG and SYSEND. These symbols are defined by FORTRAN II as follows:

SYSORG = 3720_8 or 2000_{10}
SYSEND = 77777_8 or $32{,}767_{10}$

The user may redefine SYSEND to allow space for an installation accounting routine in upper core storage. In this case, the following limits apply: for FORTRAN II, SYSEND may not be lower than 77677_8 or $32{,}703_{10}$; for all other subsystems, SYSEND may not be lower than 77013_8 or $32{,}267_{10}$.

SYSLDR contains a transfer to the System Loader. 12
The System Loader may be used to scatter-load subsystem records that have the same standard System Library record format as the first record of the subsystem. When loading is completed, the System Loader transfers to the SYSTRA location. Therefore, the contents of SYSTRA must be modified during or prior to loading.

Whenever the first record of a subsystem is loaded into core storage following the reading of a $EXECUTE card, the System Supervisor places in the decrement portion of SYSLDR the location of the unit control block for the unit from which the subsystem is being loaded. Subsequent transfers to SYSLDR by the subsystem will result in the next sequential record being loaded from the unit indicated in the decrement portion of SYSLDR. Therefore, if the decrement of SYSLDR is not changed by the subsystem, the next record of the subsystem will be loaded from the same unit as the previous record, each time a transfer is made to SYSLDR.

When the subsystem being loaded is on random access storage, the System Loader routine obtains the track address of the next sequential record from the decrement of SYSTCH (described later in the text). When editing records onto disk or drum, the System Editor appends to each record a TCH SYSCYD command containing the track address of the next record in its decrement. When a record is loaded from random access storage by the System Loader, the TCH command at the end of the record ends up in SYSTCH, where it is availabe for loading the next record.

A subsystem may specify the unit from which the next record is to be loaded by changing the unit control block address in the decrement portion of SYSLDR. However, the unit must be on the same channel (and Data Channel Switch setting if it is a 7909 Channel) as the unit from which the first record of the subsystem specified on the $EXECUTE card was loaded. When loading from random access storage, the decrement of SYSTCH must also be changed.

The System Loader may be entered using the following instruction:

```
        TSX       SYSLDR,4
```

If SYSLB1 is random access storage and the System Supervisor that processed the last $EXECUTE card was loaded from random access storage, an alternate entry may be made to the System Loader by using the following sequence:

```
        TSX       SYSLDR,4,1
        BCI       1,SYSREC
```

where SYSREC is the name of a record on SYSLB1. When this entry is made, the System Loader will load the record specified by SYSREC. The System Loader obtains the track address of the specified record from the System Loader Table (SLTABL). This table is generated and placed in the System Library when it is edited onto random access storage by the System Editor. To obtain the track address, the System Loader writes a checkpoint record on SYSLB1 (just behind the System Core-Storage Dump Program), loads the System Loader Table into core storage, looks up the address of the specified record in the table, and restores core storage.

SYSACC is used for communication between the installation accounting routine (if one exists), and the subsystems and System Monitor. In the distributed version of the System Monitor, this location contains the following:

```
        PZE       0,0,0
```

Whenever a $ID or $JOB card is processed by a subsystem or the System Monitor, SYSACC is tested before a transfer is made to SYSIDR. If SYSACC contains all zeros, the subsystem or System Monitor lists the $ID or $JOB card on the System Output and System Printer Units before transferring to SYSIDR. In the case of a $JOB card, a page eject is performed before the card is listed. If the contents of SYSACC are nonzero, the $ID or $JOB card is not listed before the subsystem or System Monitor transfers to SYSIDR. The installation accounting routine is provided with the location of the first word of the buffer containing the $ID or $JOB card (as described previously under SYSIDR) and must list the card if SYSACC is set to nonzero at an installation.

SYSPID is reserved for use in communication between an installation accounting routine, when one is incorporated in the IBSYS Operating System, and the subsystems and the System Monitor. The exact use of this location depends on the design of the installation accounting routine. In the distributed version of the Operating System, SYSPID is not used.

SYSCYD and SYSCYD+1 contain the following input/output commands, which are used by the System Loader when subsystem records are loaded from disk, drum, or Hypertape.

```
SYSCYD     CPYD      0,,0
           TCH       SYSTWT
```

SYSSLD, SYSTCH, and SYSTCH+1 contain the following input/output command sequence, which is used by the System Loader for scatter-loading subsystem records:

```
SYSSLD     CPYP      *+1,,1   (IOCP)
SYSTCH     WTR       *+1
           TCH       *-2
```

Upon completion of loading from random access storage via the System Loader, SYSTCH will contain a TCH command whose decrement will contain the track address of the next sequential record on disk or drum.

SYSTWT contains a TWT instruction which serves as a common 7909 channel transfer point for all users of IOEX.

SYSGET contains a word that indicates to the System Supervisor why control was returned to it by a subsystem. Additional information on SYSGET is contained in the section "System Nucleus."

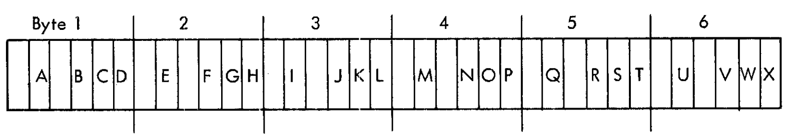
SYSJOB contains a control word that is used by the System Supervisor and the subsystems in controlling the skipping of jobs and job segments. Additional information on SYSJOB is contained in the section "System Nucleus."

Appendix B: Routine to Perform an IOEX Read or Write Using 729 Tape

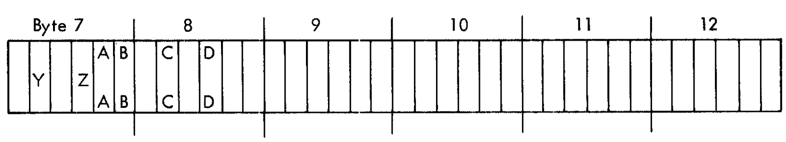
```
 THE FOLLOWING EXAMPLE IS SOLELY FOR THE PURPOSE OF ILLUSTRATION.
   IT DOES NOT REFLECT BUFFER TECHNIQUES IN USE OF 7607 CHANNEL.

*       SAMPLE ROUTINE   PERFORM READ OR WRITE FROM TAPE USING IOEX
*
*       CALLING SEQUENCE
*       TSX      IOXRW,4
*       P        LOCFIL,M,RCHSEQ
*       PZE      EOF,,ERR    EOF OR EOT OR ERROR RETURNS
*
*                WHERE...
*
*       P        IF MINUS, WRITE   IF PLUS, READ
*                BIT 1   IF 1, NO MESSAGE   IF 0, MESSAGE
$       LOCFILE LOCATION OF A WORD WITH UCB IN ADDRESS  (INDIR. REF.)
*       M        IF BCD,1   IF BINARY, 0
*       RCHSEQ   LOCATION OF I/O COMMANDS (ENDING IN TRAP)
*       EOF      END OF FILE OR TAPE EXIT
*       ERR      ERROR EXIT
                 INDICATORS AND ACC ARE DESTROYED.  IRS ARE SAVED.
*
*
IOXRW   SXA      IOXS4,4            SAVE IR4
        CLA      2,4
        STA      IOXEF              SET EOF EXIT
        ARS      18
        STA      IOXER              SET PERMANENT REDUNDANCY EXIT
        CAL      1,4                GET FIRST WORD OF CALL SEQUENCE
        STT      IOXSLL             SAVE MODE FOR SELECT
        STP      IOXSLL             SAVE PREFIX FOR SELECT TYPE
        STA      IOXND              PUT LOC IN ACTIVE CALLING SEQUENCE
        STA      *+1                SET TO PICK UP UCB
        LAC      **,4               -L(UCB)
        ARS      18                 I/O COMMANDS LOC TO ADDRESS
        STA      IOXSLL             PUT IN SELECT WORD
        CLA      IOXSLL             GET SELECT WORD
        ZET      1,4                TEST FOR OTHER USE OF THIS UNIT
        TRA      *-1                WAIT TILL UNIT FREE
        STO      1,4                SIGNED CONTROL WORD TO UCB WORD 2
        STZ      IOXIN              SET IN-OPERATION WORD ON
        TSX      (ACTIV,4           GO TO ACTIVATE
IOXND   PZE      **                 UNIT
        NZT      IOXIN              TEST FOR REQUEST COMPLETE
        TRA      *-1                NOT DONE, WAIT
        LDI      IOXIN              PICK UP COMPLETION BITS SET BY IOXSEL-
IOXS4   AXT      **,4               SET FOR EXIT
        LFT      200000             TEST EOF, EOT
IOXEF   TRA      **                 EOF EXIT
        LFT      100000             TEST FOR PERM. REDUNDANCY
IOXER   TRA      **                 YES, ERROR EXIT
        TRA      3,4                NORMAL RETURN
*
IOXSLL  PZE      0,,IOXSEL          LOCATION OF SELECT ROUTINE
IOXIN   PZE      **                 IN OPERATION CELL
IOXMD   PZE      **                 MODE SWITCH
*
*       IOXSEL ROUTINE ENTERED TWICE BY IOEX FOR EACH I/O OPERATION
*
IOXSEL  SXA      IOXSS4,4           SAVE IR 4
        PAC      0,4                -L(UCB)
        TMI      IOXPST             SELECT MINUS OR POSTING ENTRY
        CLA      0,4                UCB WORD 1
        PDX      0,2                UNIT TO IR 2
        CLA      1,4                UCB WORD 2
        STA*     (RCHXI             STORE LOC TO RCHX
        STT      IOXMD              SAVE MODE FLAG
        NZT      IOXMD              TEST MODE
        TXI      *+1,2,16           SET BINARY MODE FOR UNIT
        TMI      IOXWR              WRITE
        SXA      *+1,2              PLACE READ SELECT ADDRESS
        RDS      **                 READ SELECT
        XEC*     (RCHXI             ISSUE RESET LOAD CHANNEL
IOXSS4  AXT      **,4               RESTORE IR 4
        TRA      1,4                AND EXIT IOXSEL
IOXWR   SXA      *+1,2              PLACE SELECT ADDRESS FOR WRITE
        WRS      **                 WRITE SELECT
        TRA      IOXSS4-1           GO TO ISSUE CHANNEL COMMANDS
IOXPST  STI      IOXIN              POSTING ENTRY   SAVE ERROR FLAGS
        STL      IOXIN              SET  IN-OPERATION WORD OFF
        STZ      1,4                SET UCB WORD 2 ZERO
        TRA      IOXSS4             GO TO EXIT
```

Appendix C: Bit Assignments of 7631 Sense Data Words

SNSDTA



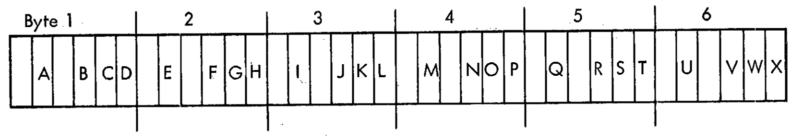
SNSDTA+1



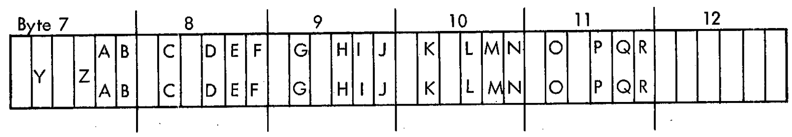
Bit	Assignment	Comment
A	Not Used	Summary Bits
B	Program Check	
C	Data Check	
D	Exceptional Condition	
E	Invalid Sequence	Program Check
F	Invalid Code	
G	Format Check	
H	No Record Found	
I	Invalid Address	Program Check for CE Track
J	Response Check	Data Check
K	Data Compare Check	
L	Parity or Cyclic Code Check	
M	Access Inoperative	Exceptional Condition
N	Access Not Ready	
O	Disk Circuit Check	
P	Control Unit Circuit Check	
Q	Channel Interrupt	
R	Six-Bit Mode	
S	Reserved	
T	Reserved	
U	ACC-MOD 00	Attention
V	ACC-MOD 01	
W	ACC-MOD 02	
X	ACC-MOD 03	
Y	ACC-MOD 04	
Z	ACC-MOD 05	
AA	ACC-MOD 06	
BB	ACC-MOD 07	
CC	ACC-MOD 08	
DD	ACC-MOD 09	

Appendix D: Bit Assignments of 7640 Sense Data Words

SNSDTA



SNSDTA+1



Bit	Assignment	Comment
A	Operator Required	Summary Bits
B	Program Check	
C	Data Check	
D	Exceptional Condition	
E	X	Selected Tape
F	X	Drive Number
G	X	(E + G = Drive 0)
H	X	
I	Drive Not Ready	Operator Required
J	Drive Not Loaded	Program Check
K	Drive File Protected	
L	Not Used	
M	Invalid Order	
N	Drive Busy	
O	Drive at BOT	
P	Drive at EOT	
Q	Correction Occurred	Data Check
R	Parity Check	
S	Code Check	
T	Envelope Check	
U	Overrun Check	
V	Excessive Skew Check	
W	Track Start Check	
X	Not Used	
Y	Read a Tape Mark	Exceptional Condition
Z	Drive in EWA	
AA	No Data Transmitted	
BB	Not Used	
CC	Read Section Busy	
DD	Write Section Busy	
EE	Backward Mode	
FF	Not Used	
GG	Drive 0	
HH	Drive 1	
II	Drive 2	
JJ	Drive 3	
KK	Drive 4	
LL	Drive 5	
MM	Drive 6	
NN	Drive 7	
OO	Drive 8	
PP	Drive 9	
QQ	Not Used	
RR	Not Used	

Appendix E: Sample Scatter-Read Program (Disk)

```
   THE FOLLOWING EXAMPLE IS SOLELY FOR THE PURPOSE OF ILLUSTRATION.
     IT DOES NOT REFLECT BUFFER TECHNIQUES IN USE OF 7909 CHANNEL.

*       SAMPLE ROUTINE   PERFORM SCATTER-READ FROM DISK USING IOEX
*
START   LAC     SYSLB1,1         -L(UCB) TO IR1.
        CLA     SYSTCH           SEEK LOCATION NEXT RECORD.
        STO     FLAG             SAVE IT IN FLAG.
        ARS     18
A       ZET     1,1              UCB 2 FREE...
        TRA     *-1              NO, WAIT.
        STA     3,1              PLACE IN UCB 4 FOR IOEX TO DO SEEK.
        CLS     2,1              SET AF FLAG SO IOEX WILL SEEK.
        STO     2,1
        TSX     (ACTIV,4         PERFORM SEEK WHILE SETTING
        PZE     SYSLB1             UP ORDER, ETC.
*
* UCB  2 CONTENTS ARE ZERO.  A SEEK REQUEST MAY BE PLACED WITHOUT
* PRIORITY PENDING.
*
AA      CLA     FLAG             PICK UP CURRENT SEEK A/O VERIFY
*                                BINARY ADDRESS.
        TSX     (DECVD,4         CONVERT ADDRESS TO BCD
        LAC     SYSLB1,4         -L(UCB) TO IR4.
        TSX     (FDAMT,2         BYTES 3-6 OF MQ, TTTT IN BCD.
        BCI     1,0CBM06         HA2 IDENTIFIER IS BM.
        PZE     DVTA             LOCATION OF ORDER TO BE COMPOSED.
        CLA     9IOSL            SET UCB 2 PRIORITY.
        STO     1,1
        TSX     (ACTIV,4         PERFORM READ.
        MZE     SYSLB1
        ZET     1,1              WAIT TILL DONE.
        TRA     *-1
        LDI     9IOSLI           7909 CONDITION PLACED BY (SEL-).
        LFT     001000           WAS IT UNUSUAL END...
        TRA     (PAWSX           YES.  READ IOEX COMMENTS ON PRINTER.
                                 GO TO ERROR PAUSE.
        CLA     FLAG             NO.  MORE OF THIS IBSYS RECORD...
        TMI     SYSTRA           NO.  GO TO IBSYS CONTROL.
        PDX     0,4              PICK UP NEXT SEEK ADDRESS.
        PXA     0,4              DOES NEXT TRACK IN BINARY REQUIRE
        LRS     35               A SEEK...
        DVP     =40
        TNZ     AA               NO.  SEEK NOT NEEDED UNLESS
        PXA     0,4              REMAINDER IS ZERO.
        TRA     A+2              YES.  SET PRIORITY FOR SEEK TO IOEX.
*
* CONSTANTS
*
9IOSL   PZE     CTLR,,9SL        PRIORITY WORD UCB 2.
FLAG    PZE
DVTA    DVTA    ,,,              VERIFY ORDER COMPOSED BY (FDAMT.
9IOSLI  PZE                      IOEX GIVEN 7909/7631 STATUS
*
* ONE PURPOSE IOEX TYPE SELECT ROUTINE
*
9SL     TRA     SE
        PZE     *+1,,1           EXCLUSION WORD ADDRESS,,COUNT.
        OCT     062001000000     EXCLUSION MASK BITS.
* ERROR PROCEDURE EXCLUDED.  STATUS RESET ON MATCHED EXCLUSION
* BITS FOR ERROR INTERRUPTS.
SE      PAC     0,2              -L(UCB).
        TMI     SM               SELECT MINUS.
        CLA     1,2              UCB2.
        STA*    (RCHXI           SET RSCX ADDRESS, VIA IR1.
        CLA     9SL+1            SET EXCLUSION WORD (OPTIONAL).
        SLW*    (RCTXI           (OPTIONAL).
        XEC*    (RCHXI           RESET AND START CHANNEL.
        TRA     1,4              EXIT TO IOEX.
SM      STZ     1,2              RESET PRIORITY WORD FOR THIS ACCESS.
        STI     9IOSLI           SAVE 7909/7631 STATUS.  ALL ERRORS
*                                ARE INDICATED BY IOEX EVEN IF
*                                EXCLUSION FLAGS ARE USED.
        TRA     1,4              EXIT TO IOEX.
*
* 7909 PROGRAM
*
CTLR    SMS*    AA+4             INHIBIT ATTN AND SELECT INTERFACE
        CTLR    DVTA             DVTA AND SET READ STATUS.
        CPYP    FLAG,,1          FIRST WORD OF A TRACK.  ITS
        CPYP    *+1,,1           DECREMENT USED BY CPU PROGRAM.
        WTR     *                SIMULATE SCATTER-LOAD FROM
        TCH     *-2              TAPE ON DISK.
* PROGRAM NOT TERMINATED BY TWT BECAUSE A TRAP WILL OCCUR DUE
* TO 7909 SEQUENCE ERROR CONDITION, WHEN A CPYD IS LOADED INTO
* WORD  CTLR+4.
```

Index

Reader's Comments

IBM 7090/7094 IBSYS OPERATING SYSTEM SYSTEM MONITOR (IBSYS)

PROGRAMMING SYSTEMS PUBLICATIONS, FORM C28-6248-1

From

Name ______________________________

Address ______________________________

Your comments regarding the completeness, clarity, and accuracy of this publication will help us improve future editions. Please check the appropriate items below, add your comments, and mail.

	YES	NO
Does this publication meet the needs of you and your staff?	___	___
Is this publication clearly written?	___	___
Is the material properly arranged?	___	___

If the answer to any of these questions is "NO," be sure to elaborate.

How can we improve this publication? Please answer below.

☐ Suggested Addition (Page , Timing Chart, Drawing, Procedure, etc.)

☐ Suggested Deletion (Page)

☐ Error (Page)

COMMENTS:

No Postage Necessary if Mailed in U.S.A.

C28-6248-1

FOLD FOLD

FIRST CLASS
PERMIT NO. 81
POUGHKEEPSIE, N.Y.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN U.S.A.

POSTAGE WILL BE PAID BY

IBM CORPORATION
P.O. BOX 390
POUGHKEEPSIE, N.Y. 12602

ATTN: PROGRAMMING SYSTEMS PUBLICATIONS
DEPARTMENT D91

FOLD FOLD

CUT ALONG LINE

IBM®
International Business Machines Corporation
Data Processing Division
112 East Post Road, White Plains, N.Y. 10601

STAPLE

Printed in U.S.A. C 28-6248-1