

**EXEC**

**UNIVAC 1107 Executive System**

**Programmers Reference**

**First Edition**

April, 1962

## TABLE OF CONTENTS

	Page
I. INTRODUCTION.....	1
II. SCHEDULE MAINTENANCE.....	5
A. Submission of Job Requests.....	5
B. Job Request Format.....	6
C. Deletion of Job Requests.....	17
III. SELECTION.....	19
A. Operation.....	19
B. Method of Selection.....	19
C. Facility Check.....	22
D. Selection Interruption.....	23
IV. FACILITY ASSIGNMENT.....	24
A. Facility Descriptions.....	25
B. Changes in Facility Requirements.....	28
C. Allocation of Facilities.....	30
D. Operator Notification.....	32
E. Release of Facilities.....	32
V. LOADING.....	34
A. Simple Program Run Under Executive Control.....	34
B. Complex Program Run Under Executive Control.....	38
C. Program Run Independent of Executive System.....	38
VI. INTERRUPT.....	40
VII. INPUT/OUTPUT.....	42
A. Submission of Requests.....	43
B. I/O Execution Packet: General.....	44
C. I/O Execution Packet: Symbolic Form.....	48
D. I/O Functions.....	52
VIII. SWITCHING.....	61
A. The Dispatcher.....	61
B. The Switch Lists.....	63
C. Storing and Restoring Film Memory.....	64
IX. COMMUNICATION.....	65
A. Communication Requests.....	65
B. Request Parameter.....	65
C. Execution Packet.....	66
D. Communication Functions.....	68
E. Communication Conventions.....	70

TABLE OF CONTENTS

	Page
X. LOGGING.....	74
XI. DUMPING FUNCTION.....	75
A. Automatic Dump.....	75
B. Program Requested Dump.....	75
XII. TERMINATION.....	76
A. Normal Job Termination.....	76
B. Abnormal Job Termination.....	76
C. Termination Specified by the Operator.....	77
D. Temporary Interruption of a Program.....	78
XIII. EXECUTIVE PROCEDURES.....	79
A. Initialization.....	79
B. Execution of Rush Jobs.....	82
C. Independent Operation of Jobs.....	83
D. Facility Transfer Function.....	84
E. Assembly (Compilation) And Testing of Programs...	89
F. Job Program Libraries.....	90
G. Rerun Function.....	94

## I. INTRODUCTION

EXEC, the 1107 Executive System, is a program designed to provide for automatic processing of a scheduled set of computer runs, allocation of memory and peripheral units, input/output operations, concurrent processing of 1107 programs.

EXEC is intended to facilitate efficient use of the UNIVAC<sup>®</sup> 1107 Thin Film Memory Computer by providing the means for automatically processing a scheduled set of jobs with a minimum of operator intervention. Jobs may be processed concurrently or serially as specified in the externally prepared Job Requests.

To accomplish its intended purposes, EXEC must perform a varied number of functions. These include:

### A. Schedule Maintenance

The acceptance of Job Requests from an external medium and the inclusion of these requests in a Job Request Schedule. EXEC will reference the Job Request Schedule to determine the next job to be initiated. Previously submitted requests may be deleted.

### B. Selection

The use of information contained in the Job Request Schedule to select the next job to be initiated. Selection is based on the priority and precedence assigned to the job, the sequence relationship of this job to other jobs with the same priority and precedence, and the availability of facilities required by the job.

### C. Facility Assignment

The assignment of memory and external facilities to meet the requirements which are defined symbolically in a job program selected for initiation. EXEC maintains a list of all allocatable facilities which is updated to reflect assignment of facilities to newly initiated programs and to reflect release of facilities by programs during, or at termination of, a run.

### D. Loading

The transfer of a job program to be initiated from the storage medium to the absolute operational facilities

assigned to the program. Programs which are in relocatable form (ROC) are transferred in their entirety to the assigned operational facilities. All necessary modifications are made during this transfer to make the program operationally compatible with the assigned absolute facilities. Each program is assigned an internal identification upon loading and is then initiated. All programs are loaded by CLAMP, the 1107 Relative Load Routine.

#### E. Interrupt

The act of providing to the operating program the entrances to subroutines which will handle the error interrupts. Upon occurrence of an error interrupt, control is transferred automatically to one of the fixed core memory addresses 192-199. EXEC will provide jump instructions for these locations. These instructions will in turn reference subroutines which will attempt to recover from these errors. The interrupt which is operated by attempts to write into a locked out area of memory is not included in this function of the Executive System. Errors dealt with include illegal operation code, characteristic overflow, characteristic underflow, and divide fault. All I/O and other interrupts are handled by other parts of the Executive System.

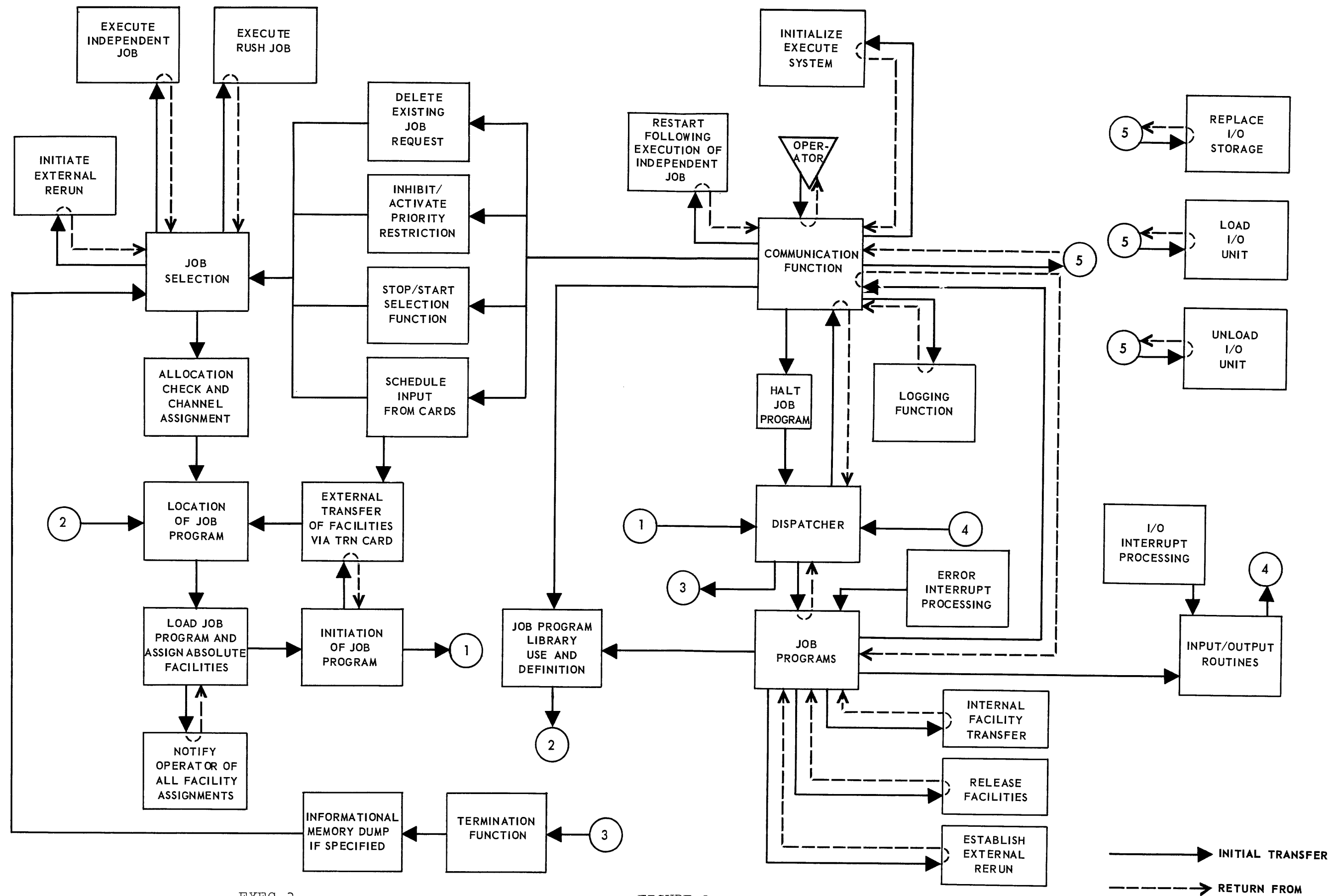
#### F. Input/Output

The acceptance, listing, and processing of all requests for I/O functions from the operating programs. This function of the Executive System makes possible the concurrent operation of several programs using the same I/O channels without the danger of one program interfering with another program's I/O functions. Requests for I/O operations are submitted to EXEC in the form of a parameter specifying the location of an "execution packet" which defines the function to be performed. An attempt is made to recover from I/O errors whenever feasible.

#### G. Switching

Providing for the transfer of operational control among two or more independent programs being operated concurrently. This function makes possible the operation of compute-limited programs concurrently with I/O-limited programs. Operational control is transferred to a compute-limited program whenever the I/O-limited programs must wait, pending completion of requested I/O functions. This function of EXEC allows an installation to make maximum use of its total computer facility.

1107 EXECUTIVE SYSTEM  
FUNCTIONAL DIAGRAM



EXEC 3

FIGURE 1

## H. Communication

Providing for all communication between the operating programs and the computer operator and between the Executive System and the computer operator. These communications take place via the computer keyboard and the on-line typewriter. This function includes the interpretation of all keyboard inputs addressed to the Executive System and the transfer of control to the section of EXEC to which the input pertains.

## I. Logging

The recording of the approximate internal processing times utilized by each program operating concurrently as well as the unused time during which no program can operate pending completion of requested functions. This record will assist the installation scheduler in determining which programs to run in parallel with each other. To facilitate the detection of infinite loops, provision is made to notify the operator of programs which utilize more time than is estimated for them.

## K. Dumping

The facility to obtain printable dumps of the contents of areas of film or core memory in case unexpected errors cause premature termination of supposedly debugged programs. The dumps are recorded on tape for later printing on the High-Speed Printer.

## L. Termination

The normal or abnormal termination of an operating program and the return of its assigned facilities to an "available" status. Termination may be initiated by EXEC, by the job program, or by the operator.

Each of these functions of the Executive System will be considered in detail in the following pages. It is assumed that the reader is familiar with the pertinent facts concerning the hardware and also with SLEUTH, the 1107 Assembly System. The format of the Relative Object Code (ROC) is discussed in the manual on CLAMP, the 1107 Relative Load Routine. Reference should be made to both the SLEUTH and CLAMP manuals and to the interrelationships of these systems with EXEC. A functional diagram of EXEC appears as Figure 1.

## II. SCHEDULE MAINTENANCE

Before a job program can be selected as a candidate for loading and initiation, a Job Request must be submitted to the Executive System. The Job Request consists of a series of input parameters to EXEC specifying the program to be initiated, its location, modifications to be made while loading, and all other pertinent information concerning the program.

### A. Submission of Job Requests

Before Job Requests can be submitted, the operator must notify the Executive System of the location of the Job Request. This is accomplished via a type-in of the form

SCH△Ccc△Uuu or SCH△Ccc△Aa△Ll

where the first of the two forms shown is applicable to Job Requests which are contained on all peripheral units except magnetic drums. The second of the two forms is used to specify the location of a Job Request on the drums. In these type-ins:

- SCH identifies the type-in as a request to EXEC to read in Job Requests.
- Ccc△Uuu specifies the peripheral unit containing the Job Requests. cc is the absolute channel number and uu the absolute unit number.
- Aa△Ll specifies the drum facility containing the Job Request. a is the absolute starting address of the drum table and l is its length.

Units which may be specified are a card reader, a magnetic tape unit, a paper tape reader, or the computer keyboard. The Job Requests are read by EXEC from the specified input device, checked for legality, and stored in the schedule of Job Requests. The existing Job Request Schedule is updated to contain only uncompleted Job Requests.

If the specified input unit is currently assigned to a job program, or to EXEC, or contains a Program Library, the operator is notified by a timeout and the Schedule



Maintenance function of EXEC is terminated.<sup>1</sup> If the input unit has already been placed in a "reserved" or "available" status by EXEC (for use during program execution), the Job Requests are read and the unit is restored to its original status.

For 1107 systems containing a magnetic drum, 2000 drum locations are used by EXEC for storage of Job Requests. The average size of a request is from 15 to 40 words; hence, from 50 to 130 Job Requests can be stored on drum. If magnetic drums are not available, then a limited number of words of core memory are set aside for this purpose.

## B. Job Request Format

A Job Request consists of a minimum of two cards.<sup>2</sup> These are the Priority Card (PTY) and the Facility Description Card (FAC). These two cards must be present for all jobs to be operated under Executive control.

Certain other cards may be present. These include the Table Length Assignment Card (TAL), the Transfer Card (TRN), and the Parameter Card (PMn). Their presence is optional and they are used if specific table lengths are to be increased, if facilities are to be transferred to succeeding jobs, and/or if the job program requires starting parameters which are unique to each run.

Job programs which are to be loaded by EXEC but which operate under their own control, require only the PTY card. All Job Request information is coded in alphanumeric (Fieldata) form.

1. Priority Card (PTY): The PTY card is the basic card for every Job Request and consists of twelve ordered fields separated by commas. Two consecutive commas are necessary to show the omission of a field except when trailing fields are omitted. All spaces and blanks are ignored.

---

<sup>1</sup>Program Libraries are discussed in Section XIII.

<sup>2</sup>The term "card" is used here and throughout this manual to indicate a unit record which could take the form of punched cards, a magnetic tape record, a paper tape record, or a keyboard type-in. Actually, a single record may consist of more than one punched card. Each record, or card, may contain a maximum of 80 characters.

The fields and the information contained therein are as follows:

JOB REQUEST ID: From one to six alphanumeric characters which identify the Job Request.

CARD TYPE: The characters "PTY" to identify the card as a Priority Card.

RUSH: This field may contain the symbol "R" to denote a rush Job Request.

PROGRAM NAME: From one to twelve alphanumeric characters which identify the job program.<sup>3</sup>

PROGRAM MEDIUM: This field defines the type of input medium on which the job program resides. It contains

- a) the symbol "T" to denote UNISERVO\* IIIA, or the
- b) the symbol "A" to denote UNISERVO IIA, or
- c) the symbol "\*" to denote that the MEDIUM NAME field which follows contains an absolute assignment.

MEDIUM NAME: This field contains the name or location of the medium containing the job program. It contains either

- a) the symbol "CccΔUuu" where cc and uu specify the absolute channel and unit location of the job program, or
- b) the symbol "\*t = p/rr" where \* defines the tape symbol t as denoting a library of programs. This designation is optional. The tape symbol t is the alphanumeric

---

<sup>3</sup>This is the tag which appears in the PRO line of a SLEUTH source program. It is expanded to 12 characters by the Assembly System with the right-most 6 characters filled with spaces. See also CLAMP Manual.

\*Trademark of Sperry Rand Corporation.

JOB REQUEST ID	CARD TYPE	RUSH	PROGRAM NAME	PROGRAM MEDIUM	MEDIUM NAME	RUN TYPE	PRIORITY	MIX TYPE	EST. TIME	LIBRARY MEDIUM	LIBRARY NAME
965004	, PTY	, R	, RRU-PAYROL-3,	T	, *PROG=ABLE/04,	S	, 01A23	, P	, 1254	, A	, STRESS.
965006	, PTY	, R	, TIME-FILE	, A	, BF3	, P	, A24	, C	, 2		.
253	, PTY	,	, DUMP	, T	, PT2	, P	, B6	, C	, 1		.

EXEC 8

Three PTY Cards are shown. In the first of these, the job program RRU-PAYROL-3 resides on a UNISERVO IIIA tape unit (denoted by "T" in the PROGRAM MEDIUM Field). The tape containing the job program is named PROG and is a program library tape (as denoted by the "\*"). The job program is the fourth program on the tape PROG. All references within the job program to the tape ABLE are replaced at load time by the absolute channel and unit assignment given to the tape PROG. This program is a rush job which must be run serially under EXEC control. It is I/O limited and utilizes subroutines stored on a tape named STRESS which is mounted on a UNISERVO IIA tape unit.

FIGURE 2: PTY CARD

name (one to six characters) of the tape containing the job program. This is the only subfield which must be present if this form of the MEDIUM NAME field is used.

The = symbol equates the tape symbol to the program symbol p. The latter is the alphanumeric name (one to six characters) used in the job program to refer to a tape. The presence of the symbol p in this field causes all job program references to this symbol to be replaced at load time by the absolute channel and unit assignment of the tape containing the job program. The logical channel associated with this tape is defined with an asterisk on the FAC Card (see below).

rr consists of two decimal digits representing the relative location of this job program on the program library tape named in the t subfield. This subfield is used only when the asterisk (\*) subfield is present.

The MEDIUM NAME field is further discussed in Section V.

RUN TYPE:

This field may contain

- a) the symbol "S" to indicate that this job must run serially.
- b) the symbol "P" to indicate that this job may run concurrently with other jobs.
- c) the symbol "I" to indicate that this job runs concurrently of the Executive System (see Section V).

d) the symbol "E" to indicate that this job is an external rerun (see Section XIII).

**PRIORITY:** From 1 to 5 characters containing the sequence, priority, and precedence assignments for the Job Request. The first two decimal digits are sequence numbers which specify the order of execution of jobs with identical execution priorities and precedences. The third character in this field is the alphanumeric priority, with "A" signifying the highest priority. The last two characters are decimal digits representing the precedence value. The priority and precedence value may be omitted. The precedence values range from 0 to 63, while the sequence values may vary from 1 to 63.

**MIX TYPE:** This field assigns the job to either the compute-limited class or to the I/O limited class of programs. It is used as a basis for distributing control to the operating job programs (see Section VIII). This field may contain.

- a) the symbol "P" to denote an I/O limited program, or
- b) the symbol "C" to denote a compute-limited program.

**ESTIMATED TIME:** Up to four decimal digits giving an estimate, to the nearest minute, of the time needed for the operation of the program. This estimate should be above the maximum and will be used to detect an infinite loop in a program.

**LIBRARY MEDIUM:** This field defines the type of input tape which contains the subroutines to be included with

the job program at load time (see manual on CLAMP). It may contain

- a) the symbol "T" to denote UNISERVO IIIA, or
- b) the symbol "A" to denote UNISERVO IIA, or
- c) the symbol "Z" if the library tape is a program tape (see Section V.B.).

LIBRARY NAME: This field contains the name of the library tape referred to above (see manual on 1107 LIBRARIAN).

The last field is followed by a period to indicate end-of-record. The PROGRAM NAME field may contain a hyphen ("-") character except as the first character in that field. When the request is for an external rerun, the PROGRAM NAME field identifies the rerun dump (of which more than one may exist). Figure 2 contains three PTY cards for three different Job Requests. These cards follow the form described above.

## 2. Facility Description Card (FAC)

This card must be present for all jobs which are to be executed under MASTER control. The card contains a JOB REQUEST ID field which is identical to the one described above for the PTY card, a CARD TYPE field with the symbol "FAC" to identify the card, and a number of FACILITY DESCRIPTION fields. These latter fields will be described in detail in Section IV on Facility Assignment. All required facilities other than those that are being transferred to the job program from another program must be listed on this card. The FAC Card is illustrated in Figure 3.

## 3. Table Length Card (TAL)

This card specifies increments to the minimum data table lengths which were established by the assembler or compiler. The card makes it possible to change at load time the amount of core or drum working area available to the program for a given run. Its inclusion in the Job Request is optional. The card contains a JOB REQUEST ID field identical to those described above, a CARD TYPE field with the

JOB REQUEST ID	CARD TYPE	FACILITY DESCRIPTIONS
965004	FAC	IB/2 , DB/3 , MD/1500 , MA1/2 , MT1/4 , *MT2/3 ,
		MT/2: PAYROL: VTORY , JP/3 .

The facility requirements described by this card are as follows:

<u>Facility</u>	<u>Number of Units</u>
Instruction Bank (IB)	2 core blocks (2048 words in each; 4096 total)
Data Bank (DB)	3 core blocks (2048 words in each; 6144 total)
Magnetic Drum (MD)	1500 registers
UNISERVO IIA tape (MA)	2 tape units on logical channel 1
UNISERVO IIIA tape (MT)	4 tape units on logical channel 1
UNISERVO IIIA tape (*MT)	3 tape units on logical channel 2
UNISERVO IIIA tape (MT)	2 tape units on one or more channels (The units PAYROL and VTORY are not to be assigned for this run of the program (see text)).
Selective Jump Switches (JP)	3 switches

The asterisk indicates that the tape containing the job program is associated with logical channel 2 and, if not already assigned, will be assigned to the same physical channel to which logical channel 2 is assigned.

FIGURE 3: FAC CARD

symbol "TAL" to identify the card, and a number of TABLE LENGTH INCREMENT fields. These latter fields are described in Section IV on Facility Assignment. The TAL card is illustrated in Figure 4.

#### 4. Transfer Card (TRN)

This card is included in the Job Request for a program which uses facilities which are to be transferred to the environment of a following program or which are to be received from other programs. The following program requires a TRN card only if the same or different facility is to be passed on to a subsequent program. (Section XIII describes some limitations.)<sup>4</sup> The card contains a JOB REQUEST ID field, a CARD TYPE field with the symbol "TRN", a RECEIVING COUNT field, and a number of TRANSFER fields.

The RECEIVING COUNT field contains one or two decimal digits denoting the number of facilities to be received by this program from one or more previous jobs in this sequence of jobs.

Each TRANSFER field consists of four subfields separated by "slash" (/) characters. These subfields are:

SUCCESSOR ID:        the JOB REQUEST ID of the program to which the facility transfer is directed.

                          If this subfield is preceded by an \*, then EXEC itself is the recipient and the remainder of the subfield specifies one of two possible types of transfer:

                          REQ    Facility contains one or more Job Requests

                          PLB    Facility contains a Program Library

                          Hence, Job Requests or a Program Library may be transferred to the Executive System.

FACILITY TYPE:        the type of facility involved in the transfer as denoted by

---

<sup>4</sup>Facilities that are being transferred to this program are not reflected in the FAC Card.



JOB REQUEST ID	CARD TYPE	TABLE LENGTH INCREMENTS
-------------------	--------------	-------------------------

965004	,	TAL	,	MATRIX/2500 , COUNT/100 .
--------	---	-----	---	---------------------------

The allocations for Table Length Tags MATRIX and COUNT are increased by 2500 and 100 respectively.

FIGURE 4: TAL CARD

JOB REQUEST ID	CARD TYPE	RECEIVING COUNT	TRANSFERS
-------------------	--------------	--------------------	-----------

965004	,	TRN	,	4 , PAY/T/965005/PAYROL , PAYFIL/D/965006/STATCS ,
--------	---	-----	---	--

	ROE/N/974002/A23 .
--	--------------------

Four facilities are to be received by this program from previous jobs in the sequence. The magnetic tape PAY is to be transferred after rewinding to job number 965005 and is to be referred to therein as PAYROL. (PAY might represent the updated master payroll file which is to become the input to the next payroll maintenance run.)

FIGURE 5: TRN CARD

- a) the symbol "D" for drum storage.
- b) the symbol "T" for a magnetic tape to be transferred after rewinding.
- c) the symbol "N" for a magnetic tape to be transferred without rewinding.

FACILITY NAME 1: the symbol by which this job program refers to the facility to be transferred.

FACILITY NAME 2: the name by which the program to which the facility is being transferred, i.e., the receiving program, refers to the facility. If the receiving program is EXEC and the facility contains a Program Library, then this subfield contains the library identity.

The subfields are written in the order

Facility Name 1/Facility Type/Successor ID/ Facility Name 2

on the TRN card. The card is illustrated in Figure 5.

If more than one TRN Card is necessary, they must be grouped together. The order of grouping may be random and the RECEIVING COUNT field may be on any card of the group.

TRN Cards for rerun Job Requests must be identical with the original cards with two exceptions:

- a. A new SUCCESSOR ID subfield may be specified, and
- b. A new FACILITY NAME 2 subfield may be specified.

## 5. Parameter Card (PMn)

A job program may require a set of input parameters to determine or select options of execution. These parameters are entered via the PMn Card. The card contains a JOB REQUEST ID field, a CARD TYPE field

JOB REQUEST ID	CARD TYPE	PARAMETERS
965004	PM1	(required parameters - up to 66 characters)

The parameters are transferred to the PARAM table of the job program. Since EXEC does not edit these parameters, "blank" characters are also transferred.

FIGURE 6: PMn CARD

with the symbol "PMn" where n is a decimal digit from 0 to 9, and a PARAMETERS field of up to 66 characters. This card is illustrated in Figure 6 and discussed in detail in Section V.

Job Requests for initiating a job program from a previously established rerun point require only the PTY Card. If the job was, and still is, part of a sequence of jobs, then the TRN Cards must be included.

All cards of a Job Request must be grouped together. Two Job Requests must not have the same identity. The order of card types in a Job Request may be random, except that the PTY Card must be first. Cards must, however, be grouped by type whenever more than one card of a type is present.

The last Job Request of a sequence must be followed by a Job Request Termination Card. This card consists of one field with six asterisks:

\*\*\*\*\*

If this card is not present, and there are no more cards, then the last Job Request is considered questionable by EXEC.

### C. Deletion of Job Requests

It may be necessary to delete a job from the schedule before or after it has been initiated. To accomplish this, the operator types in a request of the form:

DELΔid\*

where DEL identifies it as a request to delete a Job Request from the schedule

id represents the alphanumeric JOB REQUEST ID which was associated with this job on all of the Job Request cards.

\* is optional and causes the remaining jobs in the sequence, if any, to be deleted.

If the job has not been initiated EXEC will remove the job from the schedule. If the job is already operating, the following typeout will occur:

idΔISΔOPERATING

Section XII discusses termination of operating programs.

### III. SELECTION

There are two phases to the Selection function of the Executive System. The first phase involves selecting, according to priority and precedence relationships, a candidate for initiation from the job schedule. The second phase is an allocation check to determine if the facilities required by the candidate are available. If they are available, the job program is loaded and initiated. If they are not available, then another candidate is selected. This process is continued until all jobs in the current priority class have been checked for possible initiation. When this occurs, the Selection function terminates its operation.

#### A. Operation

The Selection function operates whenever one of the following changes occur:

1. Job Requests have been added to the schedule and facilities are available.
2. A job program, or EXEC itself, has released a facility from its operating environment.
3. The operator has made a facility available.
4. A job program has terminated and all of its facilities have been released and made available for reassignment.

Following selection of a job, control is given to the Loading function of EXEC to load and initiate the program. When the job has been initiated, control reverts back to Selection which attempts to select another job.

#### B. Method of Selection

Six factors are used to select the next candidate from the job schedule. They are: RUSH designator; priority assignment, in the order A to Z from highest to lowest; precedence within a priority, in the order from 0 to 63; sequence assignment, in the order 1 to 63; Run Type; and Mix Type.

All of this information is made available in the Job Request PTY Card as described in Section II.

## 1. RUSH

When a RUSH job is detected, all currently operating jobs using the needed core facilities are interrupted and the RUSH job is loaded and initiated.

## 2. Priority

All jobs receive a priority from A to Z, A being the highest. If priorities are omitted from the Job Request, EXEC assumes a "Z" priority. All jobs in a given priority class must be initiated before a job from the next lower class will be considered as a candidate. This scheme of stepping within priority class may be altered by operator intervention. A type-in of the form

TPR

will cause the Executive to "terminate priority restriction." The priorities will then be examined sequentially until a candidate is selected or until the end of the job schedule is reached. To return to the scheme of initiating all jobs within a given priority class before examining the next class, the operator may "initiate priority restriction" by means of the type-in

IPR

## 3. Precedence

Precedence assignment ranges from 0 to 63. If precedence is not assigned in the Job Request, EXEC will assume a value of 63. The jobs in a priority class will be examined sequentially according to precedence value until a candidate for initiation is selected.

## 4. Sequence

The sequence assignment serves to specify the order in which a set of associated jobs are to be run. The jobs are associated by assigning the same priority and precedence values to each of them. This association can be used for a set of jobs where the output of one job is used as input for the next job, or for a set of jobs which are to be tested.

Sequence numbers range from 1 to 63. All jobs with identical priorities and precedence values are run serially with respect to each other in the order specified by their sequence numbers. The sequence numbers do not, however, have to be consecutive. If the sequence numbers of two or more associated jobs are identical, these jobs will be run concurrently if facilities permit.

Following completion of a job (or jobs, if two or more ran concurrently) in sequence, the following job in sequence is initiated if facilities permit. If the following job in the sequence does not pass the facility availability check this fact is remembered and the job is always the first candidate for selection following any release of facilities.

This concept of scheduling permits the definition of more than one set of jobs to be run in sequence at the same priority level. The facilities assigned to one job may be transferred to a succeeding job in the sequence. The ROC program output tape from SLEUTH or from an 1107 compiler may be used as the input tape to EXEC for testing the assembled programs. (See Section XIII).

Although jobs in sequence are run serially with respect to each other, they still may be defined for parallel, serial, independent (of EXEC or external rerun types of operation, with respect to all jobs not in this sequence. There are, however, two restrictions:

- a. Facilities cannot be transferred to and from job programs operating independent of EXEC
- b. External reruns may only be initiated at the beginning of a sequence.

When a job program in sequence is terminated, with or without error, prior to its normal completion, the operator has the option of deleting the remaining jobs in the sequence. (See Section XII.)

## 5. Serial Selection

For a given priority, jobs to be run in serial will be selected as candidates for initiation when no job programs are currently operating, or after all jobs to be run concurrently have been initiated. At this time the operator is notified to load the program tape on an assigned tape unit if it is not already loaded. The serial job will



not be loaded, however, until all currently operating programs have terminated.

Job requests for external reruns and for programs which operate independently of EXEC will be selected and initiated in the same manner as for serial programs.

## 6. Mix Type

All job programs that are I/O limited in their operation will be initiated into a mix of concurrently operating programs before more than one computer limited job program is initiated. This check applies to all programs run under EXEC control except to jobs which follow in sequence.

## C. Facility Check

Once a candidate for initiation has been selected, a check is made of the Facility Availability Table to determine if the facilities required by this program are available. (The required facilities are submitted in the Job Request FAC card.)

In order to determine the availability of the requisite facilities, the following questions are asked:

1. Are there enough channels available for each type of facility?
2. Are there enough units available on each channel?
3. Are there enough consecutive locations on each requested drum channel?
4. Does the core bank with the largest group of consecutive 2048-word blocks have enough to satisfy the larger of the two core requirements (IBANK or DBANK).<sup>5</sup>
5. Is there a block of locations in the other core memory bank large enough to satisfy the smaller of the two requirements? If not, is there enough left in the assigned block of the first core bank checked.<sup>5</sup>

---

<sup>5</sup>When a 65K core memory is available, and check 4 fails, then a check is made to determine if a block of core extending into both core banks exists, which is large enough to satisfy both IBANK and DBANK requirements. If the first part of check 5 fails, then the existence of such an overlapping block of core is checked before the final check in 5 is performed. See CLAMP manual for discussion of IBANK and DBANK.

If the answer to any of these questions is NO, the Job Request will be temporarily bypassed as a candidate for initiation.

D. Selection Interruption

In order to provide the computer operator with some measure of control over Executive System operation, EXEC will accept the type-in:

HSL

which causes it to halt the process of selecting jobs for initiation. In order to cancel this order and to direct EXEC to resume the selection of jobs for initiation, the type-in

SEL

is given. This type-in has meaning only when it follows an "HSL" message.

#### IV. FACILITY ASSIGNMENT

Object programs require a certain environment in which to operate. This environment is some subset of the total facilities comprising the 1107 System. Since one or more programs may be operating and using some facilities when a new program is being initiated, all facilities will have to be referred to by the programs in a symbolic manner. The Executive System will assign an available facility to every symbolic facility referenced by the program. Available facilities are those not used by EXEC or by any other program.

The various facilities to be assigned by EXEC may be grouped into three major classes:

1. Core Memory - This category includes
  - a. an instruction storage block
  - b. a data storage block
2. Drum Memory for both program segment storage and data storage.
3. I/O Equipment - This category includes
  - a. magnetic tape
  - b. paper tape reader
  - c. paper tape punch
  - d. card reader
  - e. card punch
  - f. High-Speed Printer

These facilities are uniquely assigned to a job program for its use. They are returned to an available status when the job program terminates its operation or when it releases a facility (see IV. E. below).

The 1107 Assembly System produces two outputs which are significant for facility assignment by the Executive. One of these, the Facility Description data, is used to prepare the FAC Card of the Job Request. The Executive System checks the information on the FAC Card to determine if the required facilities are available. If they are, then the second of the two outputs, the Modification Record of the assembled program, is used to assign the required facilities. Limited changes in the facilities required in the job program may be made via the Job Request.

The FAC Job Request cards are discussed below. The Modification Record is described in the manual on CLAMP.

#### A. Facility Descriptions (FAC Card)

The form of the FAC card was illustrated in Figure 3 of Section II. The card is identified by the symbol "FAC" in the second field. The first field contains the alphanumeric identity of the Job Request.

The remaining columns of the card are divided into fields separated by commas. These fields contain the facility requirements organized by unit type, channel, and number of units. The last field may be followed by a period, comma, or blanks. Additional FAC cards with the same format may be included.

The FACILITY REQUIREMENT fields take the general form

```
*ff cc/uuuuuuu : s,s
```

where

\* identifies the logical channel with which the tape containing the job program is to be associated.

ff is a two-letter code defining the type of facility (see below).

cc are two decimal digits in the range 0-15 denoting the logical channel number. When cc=0, this implies that no specific channel associations are necessary for the units specified in the corresponding units field. This channel designation should be used only when the number of units specified is greater than one.<sup>6</sup> When cc≠0, then logical channel numbers 1 to 15 serve to associate those units which will operate efficiently with each other on a common channel.<sup>7</sup>

uuuuuuu is a one to seven decimal digit sub-field specifying the number of peripheral units, registers, or the number of 2048 word core memory blocks required for the corresponding facility.

---

<sup>6</sup>This concept does not apply to drum channel requirements.

<sup>7</sup>Note that these channel assignments are logical and do not necessarily correspond to physical 1107 I/O channels.

s is the symbol for a facility that is normally required by the job program, but which is not to be assigned for this run. This unit is not included in the count of units required on this logical channel. This subfield is not applicable to core or drum facilities.

The FAC cards describe the required facilities in terms of numbers and logical channels. The same information together with the symbol for each facility is contained in the Modification Record of the job program. The two descriptions must match. The symbol "MT 1/2" on the Job Request FAC card conditions EXEC to expect a minimum of two symbolic tape references to be defined in the Modification Record for logical channel 1.

The asterisk (\*) identifies the logical tape channel with which the magnetic tape containing the job program should be associated for efficient operation. This specification will be honored if the job program tape is not already loaded. If it is already loaded, then an attempt is made to allocate the logical channel requirement to the channel containing the job program tape. The count of the number of units required for a channel marked with an asterisk must include the tape containing the job program.

The possible values of the ff, cc, and uuuuuuu subfields are indicated in Table 1.

The repeated FACILITY REQUIREMENT fields must be organized according to the following rules:

- a. Fields must be grouped according to facility type.
- b. The fields of a given type must be listed according to the amount of units or memory locations required, with the largest requirement listed first.
- c. The logical channel zero requirement, if present, must be listed as the last requirement for that facility.

The efficiency of the facility availability check performed by EXEC will be increased if the facility requirements most apt to be unavailable are listed first on the Job Request FAC Cards. This is left to the discretion of the user.

TYPE OF FACILITY	ff	cc	uuuuuuu
UNISERVO IIIA	MT	0 to 15	1 thru 16 units
UNISERVO IIA	MA	0 to 15	1 thru 12 units
Card Reader	CR	0 to 15	1 unit
Card Punch	CP	0 to 15	1 unit
High-Speed Printer	HP	0 to 15	1 or 2 units
Paper Tape Reader	PR	0 to 15	1 unit
Paper Tape Punch	PP	0 to 15	1 unit
Magnetic Drum	MD	0 to 15	Number of required drum locations.
Data Bank	DB		Number of 2048 word blocks of core memory required for data storage.
Instruction Bank	IB		Number of 2048 word blocks of core memory required for instruction storage.
Selective Jump Switches	JP		1 thru 15

TABLE 1

NOTE: All values of uuuuuuu are given for one required channel. Additional Card Readers, etc., may be assigned to other logical channels.

## B. Changes in Facility Requirements

### 1. Peripheral Equipment (excluding drum)

Certain types of programs (the 1107 SLEUTH Assembly System, the 1107 SORT/MERGE Program) have facility requirements which vary according to the type of operation and other factors. To accommodate programs of this type, EXEC will recognize certain variations in the requirements definition fields of the FAC Card. These variations may specify modifications which are to be made to the program's facility requirements as was established at assembly time. The modifications to be made fall into three categories: partial deletion, complete deletion, and minimum requirements.

- a. **Partial Deletion:** This is accomplished by identifying those units (tape, printer, card reader, etc.) which are not to be assigned to this job for this particular operation. As an example, consider the requirement defined by the field

MT 2/5

This states that logical tape channel 2, as defined in the Modification Record of the job program, must contain exactly 5 units assigned to the program. If two units are not needed for this run of the program, then the field can be changed to read

MT 2/3 : UNIT 1 : UNIT 4

where "UNIT 1" and UNIT 4" are the symbolic unit tags for the units which are not to be assigned.

- b. **Complete Deletion:** This is accomplished by omitting the complete field from the FAC Card. For example, if all tapes on logical channel 2 in the above example are not to be assigned to this job program for this run, then the field MT 2/5 would not be listed on the FAC card. This is legal only if none of the five tapes on logical channel 2 are listed in the Modification Record of the program as part of the minimum equipment configuration.
- c. **Minimum Requirement:** Each unit symbol listed in the Modification Record of the job program is defined in the source code as part of the

minimum operating requirements of the program.

Suppose that 3 of the tape units assigned to logical channel 2 in the above example were defined as belonging to the minimum configuration. Then the field "MT 2/5" could be written as "MT 2/3" on the FAC Card if only the minimum configuration for logical channel 2 is required.

## 2. Core and Drum (TAL Card)

The fields of the FAC Card which define the number of drum and core table locations may be changed as required within the limits set by the program. The minimum number of required locations for each drum channel and for core table storage is calculated by SLEUTH. This number may be increased but never decreased. This number is used to determine in advance of program loading that there are enough drum and core locations available for use by the program. If on loading it is found that the program demands more registers than the number specified in the Job Request, an attempt will be made to allocate that amount. If the additional amount is not available, the job will be terminated with an appropriate message on the typewriter and a new job will be considered for initiation.

If any of these drum or core requirements are to be changed on loading, there must be one or more Table Length (TAL) Cards (see Section II) present in the Job Request. This card permits EXEC to receive and transmit to CLAMP a new length definition for a limited number of tables in the job program. This information is determined at assembly time from information in the job program's source code.

The TAL card is illustrated in Figure 4. The TABLE LENGTH INCREMENT fields are of the form

l/iiiiiii

where l is a core or drum Table Length Tag as defined in the source code and consists of from 1 to 6 alphanumeric characters; iiiiii is a 7 digit decimal number which is the increment to the length of the table.



When the TAL Card is used to increase the length of selected core and drum tables, the overall core and drum requirements will be changed. This change must be reflected in the FAC Card.

### C. Allocation of Facilities

Facilities will be allocated to job programs according to the following rules. The word "channels", as used in these rules, is limited to those channels which contain the same type of peripheral equipment.

All channel assignments are made on the basis of the facility requirements indicated on the FAC Cards. Unit assignments are made at load time on the basis of the information contained in the Modification Record of the program.

#### 1. Peripheral Facilities (excluding drum)

- a. Channel Assignment: The channel assignment procedure takes into consideration the MIX TYPE of the program. Channel assignments for I/O limited programs are taken entirely, if possible, from channels that currently do not contain units assigned to I/O limited programs. If this is not possible, or if the program is compute-limited, then this check is not made and the units are assigned normally.

The logical channel with the largest unit requirement will be assigned first to the channel with the largest number of available units. Then the logical channel with the second largest unit requirement will be assigned to the channel that, prior to the first assignment, had the second largest number of units available. This process is continued until all logical channels have been assigned.

Units not associated by channel will be assigned to available channels in a manner designed to leave, if possible, an equal number of available units on each channel.

- b. Unit Assignment: The symbolic unit definitions contained in the Modification Record of the job program are assigned in order of definition to available units, in ascending sequence, on the previously assigned channel.

When assigning tape units, those that have been defined at assembly as input units are assigned to the lowest numbered available units.

## 2. Magnetic Drum Facilities

- a. Channel Assignment: Drum register requirements are treated as one consecutive block. The size of this block is considered in the same manner as are the number of units per channel described above in paragraph 1a.
- b. Drum Address Assignment: The block of registers (defined in the Modification Record) required by the program is assigned to the smallest available block of drum registers large enough to hold the register requirement. All DTABLE assignments are made from this area at load time.

## 3. Core Memory Facilities

- a. Core Bank Assignment: The single core requirement, or the larger one if there is more than one, is assigned to the core bank with the largest number of available 2048 word blocks. If two core requirements are present, the other one is assigned to the alternate core bank or to the same bank if the alternate bank does not have the required availability.<sup>8</sup>
- b. Core Assignment Within a Bank: Core memory locations are assigned to job programs in consecutive increments of 2048 words. This allows efficient utilization of the memory lockout capability. The core requirement is assigned to the smallest available group of 2048 word blocks that is large enough to satisfy the

---

<sup>8</sup>For an 1107 with a 65K core memory, the Executive program itself is stored in the lower addresses of core bank 1 and its data tables are in the upper addresses of core bank 2. This arrangement leaves the balance of core memory as one continuous block available for assignment to job programs. The procedure for core bank assignment is the same with these exceptions: If there is an available block which extends into both banks and the single core requirement cannot fit in either bank (or if the two core requirements cannot be assigned to alternate banks), then an attempt is made to assign the total core requirements into the block of available registers extending into both banks before assigning the total requirement to one bank.

requirement. These blocks are assigned to the program in ascending sequence if they are contained in core bank 1 and in descending sequence if in bank 2. This procedure promotes an available area of core which extends into both core banks.

#### D. Operator Notification

After all facilities have been assigned, the Relative Load Routine will type out the assignments. Those facilities that were not assigned, for any of a number of possible reasons, are indicated in the typeout. The form of this typeout is described in the manual on the Relative Load Routine.

When facilities are not assigned, either because of an insufficient statement of requirements in the Job Request or because of the lack of additional available facilities to meet the actual requirements, EXEC drops the job program from the schedule. The assigned facilities are restored to an "available" status, and an attempt is made to select another job for initiation.

#### E. Release of Facilities

When the equipment used by a program is no longer required, it should be released by the program so that it may be reassigned to other programs. To release a unit of peripheral equipment, the address of a Transfer Packet specifying the unit to be released is loaded into a designated film memory register and control is transferred to EXEC via a Load Modifier and Jump (LMJP) instruction. EXEC will return the unit to "available" status. Once a unit has been released by a program, it cannot be regained except under special circumstances as described in Section XIII.

The facilities which may be released through this function of EXEC are:

- Paper Tape Reader
- Paper Tape Punch
- Card Reader
- Card Punch
- Magnetic Tape Unit
- High-Speed Printer
- Console Selective Jump Switches

The calling sequence for release of a facility is:

8	9	FUNCTION	14	15	SUB FIELDS	
		L	D	B	\$QØ, tr,, \$UOP :	
		L	M	J	P	\$B1, \$REL :

where tr is the address of a Transfer Packet defining the facility to be released (see Section XIII).

\$REL is the entrance to EXEC for release or transfer of facilities. EXEC returns control to the line following this call after executing the desired release.

If a program requires a unit of peripheral equipment, such as a paper tape or card reader, for reading in initial data, it should release this unit when its function has been accomplished. This may permit other programs requiring such units to be initiated at an earlier time.

When a program is to be terminated, all facilities assigned to the program will be automatically released by EXEC, with the exception of those which are to be transferred to a succeeding program (see Section XIII). Hence, the procedure described above is used only when releasing facilities during intermediate stages of a program's operation.

## A. Simple Program Run Under Executive Control

## 1. Location of Job Program

Before a job program can be loaded, its location must be specified and communicated to the Executive System. This is done via information contained in the PROGRAM MEDIUM and MEDIUM NAME fields of the Job Request PTY Card. The different types of information that can be contained in these fields are indicated in Section II. Table 2 describes various entries which may be contained in these fields and the actions taken by EXEC in each case.

EXEC selects unassigned tape units from the highest numbered unloaded units on the channel that currently has the greatest number of available units. Consequently, when the job program is loaded, the unit it is being loaded from, may also be assigned to one of the job program's symbolic tape requirements. For this reason EXEC will rewind the job program tape with interlock and, if it has been assigned to the job program, the operator will have the opportunity to load it for the job program. This is not true the case where the tape symbol is equated to a symbolic program reference (see Table 2 for t=p). Here the tape unit is not rewound.

If the tape referred to by the symbol t is also to be assigned to the job program and is equated with the p symbol, then EXEC assigns the t tape to the highest numbered available unit on the channel to which the p tape has already been assigned. After the job program has been located, EXEC will read its ID block to verify its existence.

In order to direct the operator to place a named tape on a selected unit, EXEC uses a typeout of the form,

```
LOAD△Ccc △uu△WITH△t
```

where t is the symbolic tape name

cc is the decimal channel number from  
1 to 16

uu is the decimal unit number from 1 to 16

When the operator has conformed to this order he must type in

YES

If the operator cannot conform, he types in

NO

and EXEC will delete the Job Request from the schedule and select another job.

TABLE 2: LOCATION OF JOB PROGRAM (EXECUTIVE)

PROGRAM MEDIUM	MEDIUM NAME	EXECUTIVE ACTION
*	CccΔUuu	EXEC positions the tape (after rewinding) on the assigned unit to the specified job program. A tape containing a rerun dump must be specified in this manner.
T or A	t	EXEC selects an available tape unit and instructs the operator to load the tape identified by the t symbol. After the job program has been transferred to core by CLAMP, EXEC rewinds the tape with interlock. The tape unit containing the program tape is still available for assignment to the job program being loaded.
T or A	t = p	The procedure for this case is identical to that described above with one exception. In the above case the tape unit assigned temporarily to t is still available for assignment to the job program. It may be assigned to any of the symbolic tapes referenced in that program. In this case, however, the tape defined by the symbol p in the job program receives the same absolute assignment given to the t symbol. This feature is useful when data is to be stored on the job program tape following the job program.
T or A	*t	The asterisk identifies the t tape as a library tape containing a file of programs. EXEC checks the Program Library Registration Table to determine if this program library is already loaded. If it is, then the program location information of the PTY Card is redundant. If it is not already loaded, EXEC selects a tape unit from that channel which currently contains the largest number of available units and directs the operator to place the t tape on the selected unit. This tape and its assignment is then entered into the Program Library Registration Table.

PROGRAM MEDIUM	MEDIUM NAME	EXECUTIVE ACTION
T or A	*t/rr	The two-digit field rr denotes the position of the job program on the t tape. This position is compared with the current position of the tape to determine the direction in which to search for the job program. When the job program is found the new relative position of the tape is recorded in the Program Library Registration Table.
T or A	*t=p	During the time that the program library is assigned to the job program, it is not available to EXEC for the purpose of loading job programs.
T or A	*t=p/rr	This is the most general case and is a combination of the above procedures.

## 2. Loading of Job Programs

Job programs scheduled for operation under control of the Executive System will be accepted for loading in relocatable (ROC) form only.

All programs will be loaded by CLAMP, the 1107 Relative Load Routine.

## 3. Job Program Input Parameters

A job program often requires a certain set of input parameters to determine or select options of execution. To facilitate the obtaining of a limited set of parameters by the program, EXEC accepts PMn Cards (Section II) as part of the Job Request. A maximum of ten such cards will be accepted. They must be identified sequentially as

PM $\emptyset$ , PM1, ..., PM9

The PMn card is illustrated in Figure 5. The required parameters are in the form of up to 66 Fieldata characters and are transferred by EXEC into the \$PARAM table of a job program, if and only if a \$PARAM table is defined in the program (see manual on CLAMP).

## 4. Initiation of Job Program

When a job has been loaded and prepared for operation, the Executive System will type out the following message:

```
PARALLEL RUN
JOB $\Delta$ id $\Delta$ READY FOR SERIAL RUN
EXTERNAL RERUN
```

```
MIX $\Delta$ CONTAINS $\Delta$ id1 $\Delta$ id2 $\Delta$ id3 $\Delta$ id4
```

where id is the JOB REQUEST ID of the PTY Card.  
the operator replies

YES or NO

If "YES" the job is initiated. If "NO" the job is terminated and dropped from the schedule.



## B. Complex Program Run Under Executive Control<sup>9</sup>

### 1. Location of Program

If the main program is located on the same tape as the subprograms, the symbol "Z" must appear in the LIBRARY MEDIUM field of the PTY Card. The location of the main program is then specified in the same manner as outlined in the preceding section.

If the subprograms are contained on a library tape, the location of job program tape and of library tape is also as outlined above except that the name of the library tape may not be equated to a tape symbol defined in the main program.

### 2. Loading of Complex Program

All loading and modification for both main program and subprograms is accomplished by CLAMP.

### 3. Job Program Input Parameters

Only the main program may receive input parameters.

### 4. Initiation

All programs running under Executive System control are initiated in the same manner as described in the preceding section.

## C. Program Run Independent of Executive System

### 1. Location

Table 3 describes the entries which may be placed in the PROGRAM MEDIUM and MEDIUM NAME fields of the Job Request PTY Card, and the actions taken by EXEC in each case.

### 2. Loading and Initiation

Jobs which operate serially and independent of EXEC are loaded only after all currently operating jobs have terminated. Control is then given to CLAMP which loads and initiates the program.

---

<sup>9</sup> A complex program is one composed of a main or control, program and one or more subprograms. The manual on CLAMP contains a detailed discussion of complex programs.

EXEC relinquishes complete control of the computer system to this type of program. Control is returned to EXEC by either the terminating independent program or by the operator. Section XIII contains a discussion of independent operation of job programs.

TABLE 3: LOCATION OF JOB PROGRAM (INDEPENDENT)

PROGRAM MEDIUM	MEDIUM NAME	EXECUTIVE ACTION
*	CccΔUuu	EXEC positions the tape on the specified unit to the ID block of the independent job program.
T or A	t	EXEC selects an available tape unit and instructs the operator to load the named tape.
T or A	*t	Same as that described in Table 2 for this entry.
T or A	*t/rr	Same as that described in Table 2 for this entry.

## VI. INTERRUPT

The Interrupt Section of EXEC processes all External Request, Input Data Termination, Output Data Termination, Function Termination, Real-Time Clock, and External Synchronization interrupts. The reentry address, i.e., the address of the instruction in the program which was about to be executed when interrupt occurred, for all interrupts is stored in film memory register \$BØ (address 000000). This register is therefore reserved for the sole use of the Executive System.

EXEC decodes and stores the type and channel number of the interrupt and routes it accordingly. The External Request, Input Data Termination, Output Data Termination, and Function Termination Interrupts, except for those on the channel used for typewriter and keyboard, will be used by the Input/Output section of EXEC to control the execution of all requested I/O functions. The interrupts which occur on the typewriter and keyboard channel will be used by the Communication section of EXEC to control the transfer of data and messages to or from those devices.

Each job program to be run under EXEC control must contain an image of the Error Interrupt locations (core memory addresses 192-199) i.e., the \$ERROR table. The locations within this image which correspond to interrupts which the job program is prepared to handle will contain entrance addresses to error recovery subroutines within the program. The image locations which correspond to interrupts which the job program is not prepared to handle will contain entrance addresses to the Error Interrupt section of EXEC. The section will cause an operating program to be terminated with informational memory dump in the case of interrupts which it is not expecting.

The actual core memory Error Interrupt locations (192-199) will contain references to the Error Interrupt section of EXEC which will determine the location in the program's error image which corresponds to the detected interrupt. The contents of the Program Address Counter at time of interrupt will then be stored at the address specified by the program's error image and control will be transferred to this address plus one. Eight unassigned film memory registers (locations 000120 through 000127 octal) are reserved by EXEC for processing the error interrupts.

The computer's Real-Time Clock register and the associated Real-Time Clock Interrupt will be reserved for use by the Executive System. When control is relinquished to any job program the Real-Time Clock is set to some

predetermined value to insure that the Executive always regains control after a desired time interval. This prevents programs from retaining control of the computer for an unreasonable length of time as a result of infinite loops or failure to return control as expected. This feature also allows the system to check on operator type-ins. Once the operator has initiated a type-in on the keyboard, characters must be received within a certain time interval following receipt of the preceding character. Thus, in the event the operator fails to terminate a type-in, or to terminate it in the manner expected by the Executive System, the operator can be notified to take the necessary corrective action. Otherwise, in such a situation EXEC could wait indefinitely for the completion of a type-in.

The External Synchronization Interrupt will not be used by the Executive System and may not be used by programs operating under Executive control.

## VII. INPUT/OUTPUT

The Executive System accepts requests for the performance of I/O operations from the job programs operating under its control. EXEC must maintain common control of all I/O activities so that I/O operations of concurrently operating programs do not interfere with each other. Direct use of I/O instructions is illegal for programs operating under Executive control.

To request an I/O operation, each job program loads a fixed B-register with the address of an I/O Execution Packet and then enters the I/O section of the Executive System with a Load Modifier and Jump (LMJP) instruction. The Execution Packet is a list of words stored within the job program which specifies the operation to be performed, the logical channel, and the units involved.

When an I/O request is made, the channel on which operation is desired is either busy or not in use.

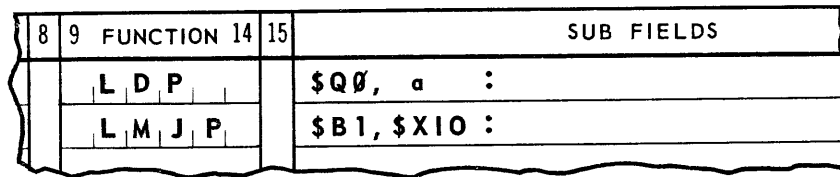
If a request is submitted when the required channel is not busy, the requested operation is initiated immediately. When a request is received and the specified channel is busy servicing a previously-submitted request, EXEC associates the address of the newly submitted Execution Packet with the identity of the requesting program and stores these items in one of three priority lists associated with each channel. Whether or not the channel is busy, control is normally returned to the requesting program after the I/O request is initiated or stored.

The job program specifies in which of the three lists the Execution Packet address is to be stored, should storing be necessary. These lists are known as the High-Priority Request List, the Medium Priority Request List, and the Low Priority Request List, respectively. There are three such priority lists for each of the 15 allocatable channels (i.e., not including the console channel) making a total of 45 in all.

If, during processing of interrupts, it is determined that all work specified by an Execution Packet has been completed, a search is made of the High Priority Request List, Medium Priority Request List, and Low Priority Request List, in that order, to obtain the address of a new Execution Packet. Within each priority group requests are taken, without regard to the identity of the program which entered them, in the same order as that in which they were entered in the priority group.

## A. Submission of Requests

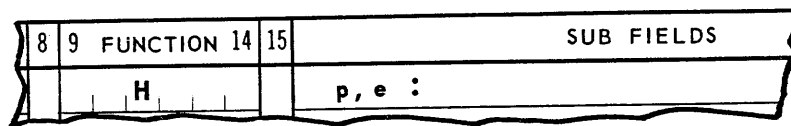
Requests for execution of I/O operations are submitted to EXEC via the following calling sequence:



where \$XIO is the entrance to the I/O section of the Executive System, and

a is the address of a Request Parameter.

The Request Parameter at address a has the form:



where e is the address of the first word of the associated I/O Execution Packet, and

p denotes the request list priority assignment.

The parameter p may take on values from 0 to 3 as follows:

- When p = 0, the request is listed in the Low Priority Request List.
- p = 1, the request is listed in the Medium Priority Request List.
- p = 2, the request is listed in the High-Priority Request List.
- p = 3, a search is made of the Low Priority Request List for a request identified by address e, which was previously submitted by the program currently in control. If such a request is found, it is removed from the Low Priority Request List and placed in the Medium Priority Request List at a logical position such that it will be used after all requests previously entered in the Medium Priority Request List. No error indication is returned to the requesting program if the request whose priority is to be upgraded is not found in the Low Priority Request List.

The priority specification does not apply if the request can be serviced immediately. All values of p other than 0-3 are illegal for job programs.

Interrupts which occur during intermediate phases of request processing will signal EXEC to issue I/O commands to read or write the next block, line or card. If an error is indicated, it is logged in a Summary Table which can be transferred to an external medium, and error recovery is attempted. If manual intervention is required, that fact will be made known to the operator by means of the console typewriter, along with the channel and unit involved. He will reply with a type-in indicating whether or not continued operation on the subject peripheral unit is feasible. After interrupt processing, if more work remains to be done on an Execution Packet current for the interrupting channel, control is returned to whatever program was operating when the interrupt occurred. Under these conditions, the Executive System saves and restores only the film memory registers it needs to process the interrupt. Whenever a request packet is terminated, the Switching section of EXEC may switch control to another program, according to the procedure described in Section VIII.

#### B. I/O Execution Packet: General

The format of an I/O Execution Packet is illustrated in Figure 7. Depending on the operation to be performed, a packet may contain from two to six words. The first two words shown, the STATUS WORD and FUNCTION WORD must always be present. The other words, if applicable, must appear in the order shown and in contiguous memory locations.

This implies that words which are not applicable in the I/O Execution Packet for a particular function are not to be included.

When a packet is submitted for execution, it is checked for legality and the status code is changed to "request in progress." The facility table in the Executive System is compared with the facilities requested in the packet to make sure there is no overlap with other programs. An image of the packet is stored in EXEC so that it can not be accidentally overlaid with erroneous information while the request is in progress. However, the Buffer Status Word and Block Read word are updated at both locations when a block, card, or line has been successfully processed.

When a request packet is terminated because of an End-of-File stop sentinel, non-recoverable equipment

error, or any other conditions under which it would be undesirable to process additional requests for operation on a peripheral unit, EXEC sets a "logical interlock" indicator applying to the unit involved.<sup>10</sup> After this indicator is set, EXEC terminates requests for operation on the interlocked unit, either from the request lists, or which are submitted after the interlock has been set, until a request to "Remove Logical Interlock" is received. Logical interlock does not apply to drum channels.

The words which form the I/O Execution Packet are:

1. STATUS WORD: By inspection of the Status Word a job program determines whether the related I/O request is stored in the waiting list, being serviced, completed normally, or terminated because of a contingency. Before the Executive System returns control to a program which has executed the request submission calling sequence, any former content of bits 35-30 is replaced by a new status code, which is thereafter updated according to changes in the following conditions:

Status Code	Meaning
(40) <sub>8</sub>	The Execution Packet address is stored in the request list.
(41) <sub>8</sub> - (77) <sub>8</sub>	The request is currently being serviced. If the Execution Packet contains fields indicating Buffer Status or number of blocks, lines, or cards processed, and if the requesting program has control, it may determine from these fields how many words of input or output have been processed so far.
(00)	The request has been completed normally.
(01) - (37) <sub>8</sub>	The request has been terminated because of a contingency identified by the value of the status code. Among the contingencies for which unique status codes are defined are:

---

<sup>10</sup>The word "termination" here refers to the deletion of the request from a Request List either because of normal completion or because of an error condition from which recovery is not possible.



The requested unit is in "logical interlock" condition.

The specified identifier has been found in a Read With Sentinel Check operation.

An End of File has been detected.

A non-recoverable read or write error has occurred.

The requested unit has been declared inoperable by an operator type-in in response to an Executive typeout.

It is possible to determine from the status code whether or not a request was terminated because of a logical interlock condition.

An interrupt code is contained in bit positions 29-24 to supplement the information contained in the status code. This is the external interrupt code most recently received from the channel through which the request was processed.

Upon termination of magnetic drum functions, bit positions 22-0 may contain a copy of the last 23 bits of the word following an End-of-Block word (see I/O Functions below), or the address of a drum search "Find" word. In case of a parity error, bit positions 22-0 will contain the address of the word causing the error.

2. FUNCTION WORD: Bit positions 35-30 contain a 6-bit number specifying the operation to be performed. A 4-bit channel designator occupies positions 29-26. If a drum operation is called for, bit positions 22-0 specify a drum starting address. For all other operations, positions 23-18 may contain format information and positions 15-0 may specify (by Master Bit Selection, i.e., by position of a "one" within the binary field) the peripheral unit on which operation is desired. No portion of this word is altered by the Executive System. Positions 15-0 should be zero for card reader or punch operations.

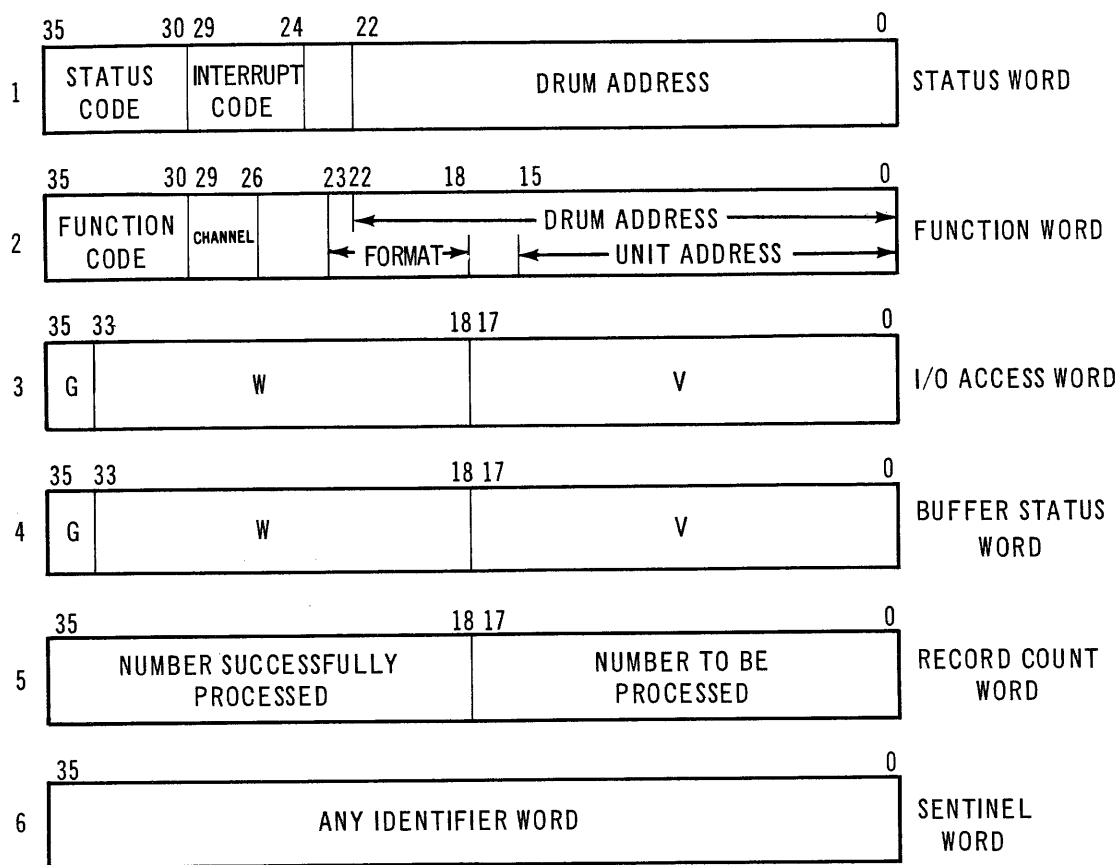


FIGURE 7: I/O EXECUTION PACKET

3. I/O ACCESS WORD: Data transfer to and from core memory is controlled by the I/O ACCESS WORD. This word is entered into the appropriate film memory location by EXEC. Bit positions 17-0 contain the core memory address, V, at which the transfer is to begin. Positions 33-18, W, represents the number of words to be transferred, positions 35-34 form the Transfer Mode Designator, G, which can specify

increment V (G=00)  
 inhibit increment V (G=01)  
 decrement V (G=10)  
 inhibit decrement V (G=11)

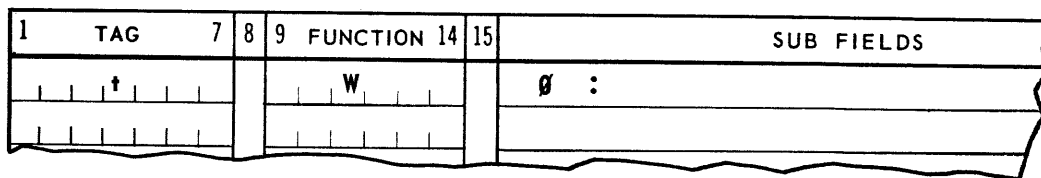
No portion of this word is altered by the Executive System.

4. BUFFER STATUS WORD: While the I/O ACCESS WORD is controlling data transfer from its film memory location, the address of the current data transfer, V, is incremented or decremented if either is specified by the designator G. The number of words to be transferred, W, is decremented. Upon termination of each function. i.e., after each block, line, card, or group of words is successfully processed, the film memory word is stored intact in the BUFFER STATUS WORD location of the Execution Packet. When a non-recoverable error, or end-of-block condition, results in a transfer of fewer words than specified, the number of words actually read or written can be determined by inspecting the BUFFER STATUS WORD. Its contents are ignored when a request is submitted for execution.
  
5. RECORD COUNT WORD: Bit positions 17-0 specify the number of blocks, lines, or cards to be processed. This right half of the word is not altered by EXEC. Upon completion of each function in multi-function requests, any content of positions 35-18 will be incremented to indicate the number of blocks, lines, or cards successfully processed. If no end-of-block or error conditions exist, the number successfully read will equal the number specified when the I/O request is terminated.
  
6. SENTINEL WORD: A 36-bit identifier word will be compared with certain data words during searches on tape or drum. The SENTINEL WORD is not altered by the Executive System. It can be used as an identifier word to terminate a read of magnetic tape or cards.

C. I/O Execution Packet: Symbolic Form

The I/O Execution Packet is coded in the DBANK area of the source program. In the examples which follow, "t" represents the symbolic content of the tag field of the SLEUTH coding line, if any.

WORD 1: This is coded as a whole word of zeros, as



WORD 2: This word is coded as an I/O Function

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	
	t				f			a, x :	

where f is any one of the mnemonic codes recognized by SLEUTH as specifying an Executive I/O function code (see below).

- a is (1) an I/O Unit Tag, or
  - (2) a Drum Table Tag, or
  - (3) a Drum Table Tag  $\pm$  constant, or
  - (4) a Drum Table Tag  $\pm$  Drum Table Length Tag

x is an integer specifying format for those Executive I/O functions where a format specification is required. This field may, for example, specify the number of lines to be spaced (skipped) prior to printing a line on the High-Speed Printer.

WORD 3: This word is coded as an I/O Access Control  
Word

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	
	t				I			n, u :	
					NI				
					D				
					ND				

where I specifies increment

NI specifies no increment

D specifies decrement

ND specifies no decrement

N is (1) a constant, or

(2) a Data Table Length Tag,  
or

(3) a Data Table Length Tag  $\pm$  constant

- u is (1) a label, or  
 (2) a label  $\pm$  constant, or  
 (3) a Data Table Tag, or  
 (4) a Data Table Tag  $\pm$  constant, or  
 (5) a Data Table Tag  $\pm$  Data Table Length Tag

WORD 4: This is coded as a whole word of zeros, as

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS				
<table border="1" style="width:100%; height:1em;"> <tr> <td style="width:25%; text-align:center;">†</td> <td style="width:25%; text-align:center;">W</td> <td style="width:25%; text-align:center;">ø</td> <td style="width:25%; text-align:center;">:</td> </tr> </table>								†	W	ø	:	
†	W	ø	:									

WORD 5: This word is coded as a half-word, as

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS				
<table border="1" style="width:100%; height:1em;"> <tr> <td style="width:25%; text-align:center;">†</td> <td style="width:25%; text-align:center;">H</td> <td style="width:25%; text-align:center;">ø, n</td> <td style="width:25%; text-align:center;">:</td> </tr> </table>								†	H	ø, n	:	
†	H	ø, n	:									

- where n is (1) a constant, or  
 (2) a tag, or  
 (3) a tag  $\pm$  constant

WORD 6: This may be coded using any form which will produce the desired 36-bit identifier, as, for example

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS				
<table border="1" style="width:100%; height:1em;"> <tr> <td style="width:25%; text-align:center;">†</td> <td style="width:25%; text-align:center;">W</td> <td style="width:25%; text-align:center;">i</td> <td style="width:25%; text-align:center;">:</td> </tr> </table>								†	W	i	:	
†	W	i	:									

or

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS				
<table border="1" style="width:100%; height:1em;"> <tr> <td style="width:25%; text-align:center;">†</td> <td style="width:25%; text-align:center;">S C</td> <td style="width:25%; text-align:center;">1, C<sub>1</sub>, ..., C<sub>6</sub></td> <td style="width:25%; text-align:center;">:</td> </tr> </table>								†	S C	1, C <sub>1</sub> , ..., C <sub>6</sub>	:	
†	S C	1, C <sub>1</sub> , ..., C <sub>6</sub>	:									

- where i may be (1) a constant, or  
 (2) a tag, or  
 (3) a tag  $\pm$  constant

and

C<sub>1</sub>...C<sub>6</sub> are any six Fielddata characters.

In addition to these words it is also necessary to code the proper declaratives to define the I/O Unit Tags, Drum Table Tags, and Drum Table Length Tags which are used in coding words 2 and 3.

The following examples illustrate the coding of some commonly used I/O Execution Packets:

1. Three 200-word blocks are to be read in the forward direction from a tape referred to symbolically as "INTAPE". These blocks are to be read into consecutively increasing addresses of a core area beginning at the address referred to as "INBUFF". The I/O Execution Packet may then be coded as

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	R	E	A	D	T	P		Ø	:
					W			INTAPE	:
					R	T	F	600, INBUFF	:
					I			Ø	:
					W			Ø, 3	:
					H				:

Note that only five words are required for this packet. The mnemonic function code "RTFN" indicates to the Executive I/O Functional Routines: "Read Tape Forward."

2. The contents of a drum table are to be read into consecutively increasing addresses of core memory such that the first word of the table read from drum is stored at symbolic core address "INBUFF". The drum table has been defined by a declarative as having symbolic name "DATA" and symbolic length "DATAL". The packet may be coded as

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	R	D	D	R	U	M		Ø	:
					W			DATA	:
					R	D		DATAL, INBUFF	:
					I			Ø	:
					W				:

where the function code "RD" indicates "Read Drum."

- Three lines are to be printed on the High-Speed Printer with 4 lines of spacing preceding each printed line. The first word to be printed is stored at symbolic core address "PTBUFF." The printer to be used has been defined symbolically by an I/O unit declarative as "PRINTR." The packet may be coded as

1	TAG	7	8	9	FUNCTION	14	15	SUB FIELDS	37
	P	R	I	N	T			Ø	:
								PRINTR, 4	:
								Ø, PTBUFF	:
								Ø	:
								Ø, 3	:

where the function code "PHSP" indicates "Print on High-Speed Printer."

- where
- f is the I/O function code
  - a specifies the unit or drum address
  - g is the symbol "I" or "D" to denote increment" or "decrement", or the symbol "N" to denote "neither."
  - w is the number of words to be transferred
  - v is the location of the first transfer
  - n is the number of words to be processed.

#### D. I/O Functions

The I/O Section of EXEC initiates appropriate function and I/O transfers to and from the peripheral equipment, as specified by an I/O Execution Packet in the program's data area. Requests for I/O operations are identified in the Request List by the addresses of the first word of the Execution Packets. A 6-bit function code in the FUNCTION WORD of the packet specifies the operation to be performed. Functions which may be specified correspond in general with commands which may be given to the Channel Synchronizers, with the following differences:

- The initiation of Function Transfer, and of Input or Output Transfer is handled by the I/O section of EXEC rather than by each job program. EXEC also supplies any Output Access Words needed for function transfer.

2. Machine functions which are terminated with an error indication are automatically repeated or re-submitted in a modified form by the I/O section whenever such a procedure would provide a recovery from the error condition. Thus, recovery commands such as "Read Tape Backward at Low Gain" are not specified directly by the job programs but are issued by EXEC in case of reading difficulties. In certain cases, the recovery procedure may be a type-out to the operator to correct a mechanical condition, and an expected reply. In this case, the Execution Packet is temporarily retired until the operator replies. If the operator cannot correct the error condition, the request is terminated.

Table 4 describes the Executive System I/O Function Repertoire. The column titled "EXECUTION PACKET WORDS" specifies which words of the I/O Execution Packet must be present for the particular function. It is important to note that the functions described here are not actual machine instructions, but are instead pseudo-functions (or system macros) which are used by the Executive to reference its own library of I/O Functional Routines.



TABLE 4: EXECUTIVE I/O FUNCTION REPERTOIRE

FUNCTION	EXECUTION PACKET WORDS	EXPLANATION
UNISERVO IIA FUNCTIONS		
Read Tape Forward	1,2,3,4,5,	Magnetic tape on the specified unit is read forward the number of blocks specified by word 5 of the Execution Packet. (For all tape operations where word 5 applies to a packet, the I/O ACCESS WORD that applies to each block is the I/O ACCESS WORD as modified in accordance with data transfers occurring in the preceding block. If the number of words to be transferred is counted down to zero before the required number of blocks have been processed, there will be no data transfer when more blocks are processed.)
Read Tape Backward	1,2,3,4,5	Same as above except that tape movement occurs in the backward direction.
Search Read Tape Forward	1,2,3,4,5,6	Tape on the specified unit is moved forward until a block is detected whose first word is identical to the identifier contained in word 6 of the Execution Packet. The I/O ACCESS WORD then controls input data transfer from the block containing the "find". Normally tape movement is concluded only when the number of blocks processed is equal to the number to be processed. Abnormal operations may cause termination before this condition is satisfied.
Search Read Tape Backward	1,2,3,4,5,6	Tape is moved backward until a block is detected whose first word (in the backward direction) is identical to the SENTINEL WORD. The block containing the "find" is read in the backward direction with the I/O ACCESS WORD controlling the data transfer. Normally tape movement is concluded only when the number of blocks processed is equal to the number to be processed.

Read Tape Forward With Sentinel Check	1,2,3,4,5,6	Magnetic tape on the specified unit is read forward the number of blocks indicated in word 5 of the Execution Packet, or until the first word of a block read is identical to the identifier in word 6. The number of blocks read will include the block containing the sentinel. A "logical interlock" is set for the tape unit referenced if the sentinel is detected.
Read Tape Backward With Sentinel Check	1,2,3,4,5,6	Same as above except that tape movement occurs in the backward direction.
Move Tape Forward	1,2,5	Tape is moved forward the number of blocks specified in word 5.
Move Tape Backward	1,2,5	Tape is moved backward the number of blocks specified in word 5.
Rewind Tape	1,2	The rewind is initiated on the specified tape unit.
Rewind Tape With Interlock	1,2	Rewind with interlock is initiated on the specified tape unit.
Write Tape 12	1,2,3,4	One record is written on magnetic tape in the forward direction at 12.5 KC density. The I/O ACCESS WORD (word 3) determines the number of words in the record.
Write Tape 25	1,2,3,4	Same as above except that density is 25 KC.
MAGNETIC DRUM FUNCTIONS		
Read Drum	1,2,3,4	Starting with the address in bit positions 22-0 of word 2, data transfer continues until the number in positions 33-18 of word 3 is counted down to zero.

Block Read Drum	1,2,3,4,5	The drum is read from the specified starting address. Data transfer is discontinued when the number of blocks specified by word 5, or the number of words specified by word 3, has been processed, whichever count is exhausted first. (A block on drum is, by definition, ended by a word consisting of 36 "one" bits. This word is called the End-of-Block word.)
Search On Drum	1,2,6	Starting with the address specified each drum word is compared with the SENTINEL WORD. If a "find" is made, the address of the "find" word is placed in positions 22-0 of word 1.
Block Search On Drum	1,2,6	Each drum word, from the specified starting address, is compared with the SENTINEL WORD. The search is discontinued when a find is made or when an End-of-Block word is detected. The address of the word terminating the search is placed in positions 22-0 of word 1.
Search Read Drum	1,2,3,4,6	Data transfer begins when a "find" word is detected and ends when an End-of-Block word is detected.
Block Search Read	1,2,3,4,6	The drum is searched from the specified starting address. If a "find" is made, the I/O ACCESS WORD governs data transfer, starting with the "find" word. Data transfer ends when the End-of-Block word is detected, or when the specified number of words is transferred whichever occurs first. The address of the word terminating the search is placed in positions 22-0 of word 1.
Chain Block Read Drum	1,2,3,4,5	The drum is read from the specified starting address one block at a time using the contents of the overflow word as the address of the start of each successive block.* Data transfer is discontinued when the specified number of blocks (word 5) or

\*The overflow word follows the End-of-Block word. This word is transferred by the peripheral subsystem to the External Status Word (octal address 311) of the core memory. The Executive System then picks up the least significant 23 bits of this word and transfers it to word 1 of the Execution Packet.

		number of words (word 3) has been read, whichever count is exhausted first.
Write Drum	1,2,3,4	The data area as defined by the I/O ACCESS WORD is copied on the drum, starting at the specified drum address. Any End-of-Block words to be written must be in the data area.
PUNCHED CARD FUNCTIONS		
Condition for Fielddata Input	1,2	Read and discard any data remaining in the Card Control Unit memory and condition the Control Unit to translate input data from card code to Fielddata code.
Condition For Column-Binary Input	1,2	Read and discard any data remaining in the Card Control Unit memory and condition the Control Unit to transfer input data "card image by column" to the Central Computer.
Condition For Row-Binary Input	1,2	Read and discard any data remaining in the Card Control Unit memory and condition the Control Unit to transfer input data "card image by row" to the Central Computer.
Input Transfer Without Trip	1,2,3,	One card image is transferred to the memory locations specified by word 3. Stacker zero is selected for the card.
Read and Trip Fill	1,2,3,4,5	The number of cards to be read is specified by positions 17-0 of word 5. If the number of words to be read, positions 33-18 of word 3, is less than the number of words in the card images specified in word 5, transfer is discontinued after the specified number of words have been read, but tripping continues.
Trip Without Transfer	1,2	One card is tripped and its contents read to the Card Control Unit memory. The card image may subsequently be read with the command "INPUT TRANSFER WITHOUT TRIP" or "READ AND TRIP FILL".

Read and Trip Fill with Sentinel Check	1,2,3, 4,5,6	This function is the same as READ AND TRIP FILL with the following exception. When the first computer word of data from a card being read is identical to the identifier in word 6, reading is discontinued. A "logical interlock" is set for the referenced card reader if the sentinel is detected.
Condition for Field-data Output	1,2	For succeeding cards, the Card Control Unit is conditioned to receive Field-data output for translation to Card Code.
Condition for Column Binary Output	1,2	For succeeding cards, the Card Control Unit is conditioned to receive binary output to punch "by columns".
Condition for Row Binary Output	1,2	For succeeding cards, the Card Control Unit is conditioned to receive binary output to punch "by row."
Punch 80-Column Cards, Select Stacker Zero	1,2,3, 4,5	The number of cards to be punched is specified in positions 17-0 of word 5. Initially, the content of word 3 is transferred to word 4. Then, for each card punched, the number of words specified in positions 33-18 of word 4 is decremented by 14 words (Fielddata output), 36 words (Row binary output), or 27 words (column binary output). Any cards for which the number of words to be transferred is less than the number appropriate for the current translation mode, will be punched as if binary zero words supplemented the missing "words-to-be-transferred". Cards successfully punched are routed to stacker zero. Cards having verification errors are sent to stacker one.
Punch 80-Column Cards, Select Stacker One	1,2,3, 4,5	Same as above with the following exception: Cards successfully punched as well as cards having verification errors will be sent to stacker one.

HIGH-SPEED PRINTER FUNCTION		
Print on High-Speed Printer	1,2,3,4,5	Groups of 22 computer words or less (containing Fielddata coded characters) starting at the address specified in positions 17-0 of word 3 are transferred to the High-Speed Printer until the number of lines printed is equal to the count specified in positions 17-0 of word 5. For this request the number in positions 33-18 of word 3 is ignored. If a Fielddata stop code (octal 77) is used, only the words preceding and including the word containing the stop code will be transferred to the High-Speed Printer. Data Transfer for the next line of print will begin following the word containing the stop code. Before each line is printed, the paper in the printer is spaced from zero to 63 spaces, as specified in positions 23-18 of word 2. The BUFFER STATUS WORD will contain an address one greater than the address from which the last data was transferred to the High-Speed Printer.
REQUEST CONTROL FUNCTIONS		
Remove Logical Interlock	1,2	Upon receipt of this request EXEC discontinues the mode of operation wherein requests for operation on the output or input/output unit specified in word 2 are unconditionally terminated. This mode of operation, or "Logical Interlock" is initiated upon occurrence of an error or end-of-file condition under which it is generally not desirable to service all the requests which may have been submitted in advance. The condition is made known by an indication in the status code of all requests affected. Logical interlock is not set on drum channels.
Remove Logical Interlock on Input-Only Unit	1,2	Same as above except that it applies to a card reader on the designated channel.

<p>Terminate Outstanding Requests</p>	<p>1,2</p>	<p>Requests identified by words 1 and 2 are cleared from all priority lists. Requests "in-progress" are terminated as soon as the current block, line, card or group of words is processed. Peripheral units other than drums are identified as in other requests, i.e., by a channel number and single master bit unit designator in word 2. For drum channels, each request is terminated where the initial drum address, X, meets the condition <math>L \leq X \leq H</math> where L is the number in positions 22-0 of word 1 of the "terminate" request and H is given by positions 22-0 of word 2.</p> <p>EXEC terminates outstanding requests for job programs which are closed out. To be terminated by any job program, a request must have been originally submitted by that program. The terminate request neither sets nor clears the Logical Interlock described in the preceding functions.</p>
<p>Terminate Outstanding Requests for Input-Only Unit</p>	<p>1,2</p>	<p>Same as above except that it applies to a card reader on the designated channel.</p>

## VIII. SWITCHING

### A. The Dispatcher

The Dispatcher is a part of the Switching section of the Executive System. It accomplishes the switching of control among the various programs operating concurrently. To accomplish this, two Switch Lists are maintained to accommodate programs of two general types:

1. Switch List 1 controls those programs which can be assumed to voluntarily release control when they cannot continue their internal processing pending completion of a requested I/O function. It is further expected that programs in this list will voluntarily release control at relatively frequent intervals.
2. Switch List 2 controls those programs which cannot be expected to release control at frequent intervals. Control will be taken away from these programs when an I/O interrupt occurs signifying completion of a requested I/O function. These programs may also voluntarily release control. In either of these cases, control will be switched to the next program in Switch List 1 which is not in a "wait" condition, i.e., which is not awaiting completion of an I/O or Communication request. If all programs in Switch List 1 are in a "wait" condition, control will be transferred to the next program in Switch List 2.

The Dispatcher is referenced by the operating programs through the Load Modifier and Jump (LMJP) instruction to voluntarily release control. When this is done, it will be assumed that the program has stored the necessary contents of film memory and has provided for restoration of these registers as required. The address entered in the designated index register (film memory register B1) as a result of the LMJP instruction will be stored as the new reentry point for the program in the Switch List. Control will then be transferred to the next program in Switch List 1 regardless of the list assignment of the program releasing control.

An optional entrance to the Dispatcher is provided by which programs may voluntarily release control with the Dispatcher providing for saving and restoring the film memory registers.



The Dispatcher is referenced by the various sections of the Executive System to return control in the normal manner to the program which was operating at the time the Executive obtained control. These references will imply that the Dispatcher restore any film memory registers utilized by the Executive System to their condition at the time of interrupt, record the current clock register setting, and return control to the address at which the program was interrupted. A reference of this type is made by the I/O section of the Executive when it has received and listed an I/O request from an operating program, and when the Request List is not in an overflow condition (see Section VII).

When an I/O or communication request is received which places the Request List in an overflow condition, the Dispatcher takes control away from the requesting program. This means that the Dispatcher must save the contents of the film memory registers for the program and flag that program's switch list entry to indicate that the registers must be restored when the program is again referenced. The Dispatcher retains control until sufficient requests have been completed to alleviate the overflow condition.

When the I/O section of EXEC detects an interrupt indicating the completion of a requested I/O function, the Dispatcher is referenced to return control to a program in Switch List 1. If the interrupted program is from Switch List 1, control is returned to this program in the normal manner. If the interrupted program is from Switch List 2, then control is taken from this program and switched to a program in List 1. If no program in List 1 can operate, control is switched to the next program in List 2. In the latter two cases, the Dispatcher saves the contents of the film memory registers for the program from which control is taken and flags its switch list entry accordingly.

It is intended that Switch List 1 will cycle at a faster rate than List 2 and that, through a judicious assignment of programs to List 1 or List 2 according to their known characteristics, those programs most likely to keep the I/O channels busy will be given control more frequently. When List 1 degenerates to zero, the result will be straight rotational switching among the programs in List 2. This switching will be triggered by I/O interrupts and by voluntary releases of control. On the other hand, when List 2 degenerates to zero, straight rotational switching will occur among the programs of List 1, with the switching triggered only by voluntary release of control. It is therefore

apparent that an imprudent assignment of programs to List 1 can defeat the purpose of concurrent operation because of the dependence in List 1 on voluntary release of control.

## B. The Switch Lists

An entry is maintained in either List 1 or List 2 for each program sharing the computer memory at any given time. The program is assigned to one of these lists, upon loading, according to its characteristics:

1. I/O limited programs to List 1
2. Compute limited programs to List 2

These characteristics are indicated in the PTY Card of the job request. If the characteristics are not indicated, the program is automatically assigned to List 2.

The entries maintained for each program in the Switch Lists contain the following information:

1. REENTRY ADDRESS: The address at which the program's operation was last interrupted and the one to which control is transferred when the program is again initiated.
2. FILM MEMORY INDICATOR: indicates to the Dispatcher whether or not the contents of film memory must be restored for the program before it is given control. This indicator is set at the time of interruption of a program's operation, depending on the conditions under which the interruption occurs.
3. MEMORY LOCKOUT INDICATOR: specifies the quantity to be loaded into the Memory Lockout Register before the program is given control to insure that all areas will be locked against attempted write-ins except those areas assigned to the program currently operating.
4. WAIT INDICATORS: these are set by the various sections of the Executive System to temporarily retire a program from the switching cycle. All wait indicators associated with a program entry must be cleared to zero before the Dispatcher will again switch control to the program.

A wait indicator is set to 1 in response to a type-in request to temporarily halt the execution of a program. This indicator is cleared when an associated type-in is received requesting that this program's execution be resumed.

### C. Storing and Restoring Film Memory

Whenever the Executive System stores or restores film memory it is implied that the following is stored or restored:

Addresses 0 through 37<sub>8</sub>, 100<sub>8</sub> through 117<sub>8</sub>, and 130<sub>8</sub> through 177<sub>8</sub>. Excluded from this range are the 16 Input Access Control Word and 16 Output Access Control Word locations and 8 registers reserved by EXEC to process the error interrupts. The Real-Time Clock register (address 100<sub>8</sub>) is available to programs for reading only.

Arithmetic carry and overflow conditions are tested and recorded when storing takes place. When the film memory is restored, carry and overflow conditions are reset according to the state they were in when the film memory was stored.

Due to the relatively large number of film memory registers which must be stored and restored when control is arbitrarily taken away from a program, considerable stress has been placed in the switching procedure on voluntary release of control with the program storing and restoring its own film memory registers.

## IX. COMMUNICATION

The Communication section of the Executive System will handle all communication between the operator and the operating programs. This communication will take place via the computer keyboard and on-line typewriter on the console channel. Neither the keyboard nor the typewriter can be assigned to operating programs.

### A. Communication Requests

To request a function of the Communication Section, the program must load a designated film memory register with a Request Parameter (described below) and execute a Load Modifier and Jump (LMJP) instruction to the Communication Section. The Request Parameter includes the the address of the Execution Packet for the function to be performed. The Communication Section will store the request in the waiting list for the console channel and set the "status" in the Execution Packet to indicate that the request has not been completed. If there is no request currently in process, the requested function will be initiated. In either case, control will be returned to the requesting program at the address following the LMJP instruction.

The calling sequence used to enter the Communication Section is

8	9	FUNCTION 14	15	SUB FIELDS	37
		L D P		\$QØ, a	:
		L M J P		\$B1, \$COM	:

where \$COM is the entrance to the Communication Section, and

a is the address of the Request Parameter.

### B. Request Parameter

The Request Parameter specifies the address of the Execution Packet for the function to be performed and, if a chain of functions is being requested, the number of additional packets in the chain.

The Request Parameter may be written in the form:

8	9	FUNCTION 14	15	SUB FIELDS	37
		G		z/6, e/30	:

where e is the address of the first word of the Execution Packet, and

z is the number of additional packets in the chain, if any.

### C. Execution Packet

The Execution Packet specifies to the Communication Section the function which is to be performed. Included within each packet is a 6-bit field which is used by the Communication Section to record specific codes to indicate to the requesting program the current status of each requested function. This code will indicate such conditions as "request in progress" or "request completed normally."

A Communication Execution Packet consists of a group of words arranged in a specific order according to the function which is to be performed. The words comprising the packet are of six basic types. Table 5 illustrates both the binary and symbolic forms of the Execution Packets corresponding to the six communications functions. An explanation of the six word-types follows:

#### Type 1: Function and Status Word

- f is a 6-bit code indicating the function to be performed.
- s is a 6-bit code indicating the status of the request. This code may be 00 to indicate normal completion of the request, a number from 01 to 37<sub>8</sub> to indicate that the request has been terminated due to an error, or a number from 40<sub>8</sub> to 77<sub>8</sub> to indicate request in progress.





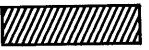
#### Type 2: Output Control Word

- r is the 16-bit address of the word containing the first character to be transferred as output. Bits 3<sub>4</sub> and 3<sub>5</sub> must be zero. Output characters are obtained from successive sixths of a word in order beginning with the most significant sixth. Succeeding words are obtained from consecutively increasing addresses.
- n is the number of characters to be transferred as output.

#### Type 3: Chain Word

This word is optional and is required only if two or more packets are to be chained together in a "multiple" request.

TABLE 5: COMMUNICATION EXECUTION PACKETS

FUNCTION	PACKET							
	WORD TYPE	35	30 29	24 23	18 17	12 11	0	SYMBOLIC FORM
TYPE	1	f		s				F f, Ø :
	2	00	r				n	H r, n :
	3				00		d	H Ø, d :
READ	4	f		s		m		F f, Ø :
	5	00	i				t	H i, t :
	3				00		d	H Ø, d :
TYPE AND READ	4	f		s		m		F f, Ø :
	2	00	r				n	H r, n :
	3				00		d	H Ø, d :
	5	00	i				t	H i, t :
LOAD, UNLOAD, CHANGE	6	f	c	00	s		u	F f, u :
	2	00	r				n	H r, n :
	3				00		d	H Ø, d :
					indicates that this portion of the word is used by the Executive for temporary storage.			

d is the 16-bit address of the first word of the next packet in the chain. Bits 16 and 17 must be zero.

Type 4: Function, Status and Characters Accepted as input

m is the number of characters actually accepted as input.

f  
s are as described for the Type 1 word

Type 5: Input Control Word

i is the 16-bit address of the word into which the first input character is to be transferred. Bits 34 and 35 must be zero. Input characters are stored in successive sixths of a word, in order, beginning with the most significant sixth. Succeeding words are stored at consecutively increasing addresses.

t is the number of characters to be accepted as input.

Type 6: Function, Channel, Status, and Unit

f  
s are as described for the Type 1 word

c is a 4-bit absolute channel designator

u denotes a peripheral unit (by master bit selection)

#### D. Communication Functions

In addition to the more general communication functions which are to be described here, a standard set of messages and type-ins are available for communication between the operator and the Executive System. They are introduced where needed in other sections of this manual. All type-ins intended for the Executive System are prefixed by three letters which define the specific function.

The following general functions are available through the Communication Section of the Executive System:

1. TYPE: This function allows the program to request that a specified number of characters be transferred as output to the typewriter from contiguous memory locations as specified in the Execution Packet. When

the transfer has been completed, an indication will be recorded in the Execution Packet to reflect completion of the request.

2. TYPE AND READ: This function allows the program to request that a specified number of characters be transferred as output to the typewriter and that a specified number of characters be accepted as input from the keyboard in reply. The program's identity will be typed out preceding the program's message. The operator will be expected to type-in this identity preceding his reply. The type-in is deleted if the number of characters specified in the Execution Packet is exceeded. The message will be terminated by an end-of-message code on the keyboard.
3. READ: This function allows the program to request that a specified number of characters be accepted as input from the keyboard. The program's identity, message identity and the word "ACCEPT" will be typed out preceding the reply. When the operator's type-in has been completed, an indication of normal completion will be recorded in the request packet.
4. LOAD: This function provides the operating program with a standard method of notifying the operator that it requires a tape to be mounted on a specified tape unit, a paper tape to be placed in a paper tape reader, or a card deck to be placed in a card reader. The Execution Packet includes the absolute channel and unit numbers of the pertinent peripheral equipment together with the control word for typing out the label by which the tape or card deck may be identified. The Executive will notify the operator (via the typewriter) to load the specified device with the required tape or card decks. When the operator acknowledges that the loading has been accomplished, the Executive records an indication of normal completion in the Execution Packet.
5. CHANGE: This function provides the operating program with a standard method of notifying the operator that a magnetic tape is to be removed from a specified tape unit, and that a tape is to be mounted in its place. The Executive notifies the operator (via the typewriter) of the function to be performed, together with a type-out of the absolute channel and unit numbers of the specified unit. The tape label to be used is also typed-out.



When the operator acknowledges (via the keyboard) that the function has been accomplished, the Executive records an indication of normal completion in the Execution Packet.

6. UNLOAD: This function allows the program to notify the operator that it requires a tape to be removed from a specified tape unit, labeled in a certain way, and a master tape ring inserted if required. It is also used to notify the operator that a card deck is to be removed from a card punch and labeled, or that a paper tape is to be removed from a paper tape punch and labeled. The Executive notifies the operator of the function to be performed (via the typewriter) together with the channel number, unit number, and label to be used.

When the operator indicates completion of the requested function, an indication of normal completion will be recorded in the Execution Packet.

#### E. Communication Conventions

The Executive System will type-out on the Monitor Printer whatever characters are typed-in on the Keyboard, on a character-by-character basis, i.e., as each character is received as input it will be sent as output to the Monitor Printer. This allows the operator character-by-character visual verification of each data character accepted as input, in order that he may immediately detect an error and take the necessary steps to correct it. The Executive System will not accept input from the Keyboard while the Monitor Printer is being used to output other data.

Output on the Monitor Printer requested through a communications Execution Packet will begin with a carriage return followed by three line feeds. This will be followed by six characters identifying the program from which the message originates. If the identity is less than six characters, it will be printed and followed by sufficient spaces to make the total six. For those messages requiring a reply, a space, followed by three alphanumeric characters assigned by the Executive to uniquely identify the message, will be typed. For those messages not requiring a reply, a space followed by three slashes will be typed. Two additional spaces will then be typed. The time, requiring four characters, and two spaces follow. These initial 18 charac-

ters of output are supplied by the Communication Section of the Executive System. The text supplied by the requesting program is then typed, beginning in the 19th character position.

The Executive will maintain a count of characters of text printed. When a carriage return is detected in the text or when the count of characters of text printed equals 54, the count is reset to zero, and a carriage return, a line feed and 18 space codes are sent to the Printer. The typing of the text is then resumed, in the 19th character position. When the specified number of characters of text have been typed, the Executive types a special character to indicate the end of the message.

Each Keyboard input message will be initiated by the operator by depressing the Interrupt Enable button on the operator's panel and then the carriage return key on the Keyboard. If the Monitor Printer is not being used to output some other message, the Executive responds to the resultant External Interrupt by sending a carriage return code, three line feed codes and seven space codes to the Monitor Printer. The Executive then initiates input mode on the channel to allow the operator to proceed to type-in the desired input on the Keyboard. If the Monitor Printer is busy when the above mentioned External Interrupt occurs, the interrupt will be noted by the Executive and the response will be delayed until such time as the Printer is free.

As each input character is received it is typed-out on the Monitor Printer. The first three characters typed-in by the operator must be the alphanumeric identity of the output message to which he is replying or in the case of unsolicited type-ins to the Executive System, the three alphabetic characters identifying a standard type-in. After these characters have been typed on the Printer, the Executive automatically types two spaces, the time, and two additional spaces to align the following input text with the preceding input and/or output texts on the page. The operator may then proceed to type the input text which, in addition to being typed-out, will be transferred character-by-character to the core storage locations set aside for the input by the recipient program.

If at any time the operator wishes to cancel a partially completed input message, he may do so by depressing the Interrupt Enable button on the operator's panel and then a special "delete" code on the Keyboard. This

causes the Executive to send the code to the Printer and to reset its tables to the state in which they were at the time the interrupt occurred, signalling the beginning of the message. Logically no input will have been transferred to the recipient program's input area, nor will the type-in have been initiated.

When the operator wishes to delete a character, the procedure is to depress the Interrupt Enable button and then a special "erase" code on the keyboard. The erase code is printed and the last character is removed from the input area. For each additional character deletion, the above procedure is repeated.

As in the case of output, the Executive maintains a count of the number of characters of input text typed-in. If a carriage return is typed-in or if the count equals 54, whichever occurs first, the Executive sends a carriage return, a line feed, and 18 space codes to the printer and resets the character count to zero. If the carriage return was typed-in, it will be stored in the program's input area as part of the input message. If the carriage return was generated by the Executive, the carriage return will not be stored as part of the input message.

The operator will conclude his type-in by depressing the Interrupt Enable button and then a special "end-of-message" code on the keyboard.

Table 6 illustrates the formats of the type-outs and type-ins as they appear on the Monitor Printer. The symbols appearing in this table are:

iiiiii	the JOB REQUEST ID of the PTY Card
bbbbbb	the word "LOAD" or "UNLOAD" or "CHANGE"
tttt	the time, in the range from 0000 to 2400,
ddd	alphabetic symbol for unsolicited type-in
cc	channel number
uu	unit number
mm	message identification assigned by EXEC.

TABLE 6: MONITOR PRINTER MESSAGES

MESSAGE	EXPLANATION
TYPE-OUTS	
<p>iiiiiiΔPmmΔΔttttΔΔTEXT</p> <p>iiiiiiΔ//ΔΔttttΔΔTEXT</p> <p>ΔΔΔEXEΔXmmΔΔttttΔΔTEXT</p> <p>ΔΔΔEXEΔ//ΔΔttttΔΔTEXT</p> <p>iiiiiiΔPmmΔΔttttΔΔ"ACCEPT"</p> <p>iiiiiiΔPmmΔΔttttΔΔbbbbbbΔCccΔUuuΔTEXT</p>	<p>Program message requiring a reply.</p> <p>Program message not requiring a reply.</p> <p>Executive message requiring a reply.</p> <p>Executive message not requiring a reply.</p> <p>Program "ACCEPT" message requiring a reply.</p> <p>Load, Unload, or Change message.</p>
TYPE-INS	
<p>ΔΔΔΔΔΔΔPmmΔΔttttΔΔTEXT</p> <p>ΔΔΔΔΔΔΔXmmΔΔttttΔΔTEXT</p> <p>ΔΔΔΔΔΔΔdddΔΔttttΔΔTEXT</p>	<p>Reply to program message.</p> <p>Reply to Executive message.</p> <p>Unsolicited type-ins to EXEC.</p>

## X. LOGGING

The Executive System maintains a log of the internal processing time utilized by each operating program. This time is necessarily approximate due to the fact that I/O transfers have priority over operating instructions and variations in the I/O load will therefore affect the number of instructions executed in a given real-time period.

The running total of internal processing time utilized by a program is compared periodically with the estimate of the maximum operating time for the program (if an estimate was provided in the Job Request). If the time utilized exceeds the estimated time by some predetermined amount, the operator is notified. He may then decide what action to take with regard to the continued operation of the program.

If at any time there are no programs which can operate, pending completion of requested I/O functions, the Executive will record the unused internal processing time. This information can be used to aid the scheduler in determining which programs run efficiently in parallel with each other.

The log of all running times is maintained in a table internal to the Executive System. Upon termination of a job, the total internal processing time utilized by the job program is displayed on the Monitor Printer.

A minimal log of peripheral equipment errors is also maintained.

## XI. DUMPING FUNCTION

The Executive System includes a facility for obtaining informational memory dumps in case unexpected errors occur which cause premature termination of supposedly debugged programs. These dumps are provided only in connection with termination of the program run and are not intended as a substitute for dumps obtainable through normal debugging procedures. Informational memory dumps are recorded on tape for later printing on the High-Speed Printer.

### A. Automatic Dump

An automatic dump is provided in conjunction with termination of a job if an error interrupt occurs which the job program is not prepared to handle or the Executive detects an error in a program. (See Section XII.)

The initial part of the dump includes the JOB REQUEST ID associated with the terminating program, the type of error detected, and the contents of the "P" register at the time of interrupt due to error or infinite loop. The remainder of the dump includes the contents of film memory, the state of the carry and overflow indicators, the state of the Console Selective Jump switches assigned to the program, and the contents of all core locations assigned to the program.

### B. Program Requested Dump

A dump is also provided, if requested by a program in conjunction with abnormal termination of the job. In this case the dump includes the same information as automatic dumps with the exception that the "P" setting included is that which is recorded in index register B<sub>1</sub> by the Load Modifier and Jump instruction through which termination is requested. This address identifies the point in the program at which the termination with dump was requested.

## XII. TERMINATION

### A. Normal Job Termination

In order to terminate the operation of a program at the normal end of a job, control is returned to the Executive System through execution of the following instruction:

8	9	FUNCTION 14	15	SUB FIELDS	37
		L M J P		\$B1, \$END	:

When this reference is made, the program is removed from the switching cycle, entries pertinent to the program are deleted from the system tables, and all facilities assigned to the program are returned to available status, except those which have been transferred to another program.

The operator is notified of a normal job termination through a type-out on the Monitor Printer. The type-out includes the JOB REQUEST ID of the terminating job, the time of day, and the address recorded in index register B1 by the referencing instruction. The latter identifies the point in the program at which termination was requested.

### B. Abnormal Job Termination

If a program is to be terminated for reasons other than the normal end of the job, e.g., because of a non-recoverable peripheral equipment error, several options are available. The program may specify termination with or without an informational memory dump and with or without deletion from the schedule of other jobs in sequence with the terminating job.

The calling sequence for specifying such an abnormal termination is:

8	9	FUNCTION 14	15	SUB FIELDS	37
		L D P		\$Q0, p	:
		L M J P		\$B1, \$ERR	:

where p is the address of a parameter word specifying the type of termination.

The parameter at p is of the form

8	9	FUNCTION 14	15	SUB FIELDS	37
		G		d/1, s/1, 0/34	:

where `d` is either 1, specifying that an informational memory dump is to be taken.  
or 0, specifying that an informational memory dump is not to be taken.

and `s` is either 1, specifying that jobs in sequence with the terminating job are to be deleted from the schedule  
or 0, specifying that the operator is to be given the option of deleting jobs in sequence with the terminating job, if any.

This type of termination is used in lieu of an "error stop" by programs operating under Executive control.

Termination procedures are similar to those described for normal job termination with the exception that an information dump is recorded if called for and, if specified, succeeding jobs in sequence with the terminating job are deleted from the schedule. If the program does not specify deletion of succeeding jobs in the sequence, the operator is notified by type-out of this condition and has the option of specifying by type-in whether jobs in sequence with the terminating job are to be run or deleted from the schedule.

Abnormal job termination may also be initiated by the Executive in case an error interrupt occurs which a job program is not prepared to handle or the Executive detects an error in a program, e.g., an illegal parameter is submitted by a program. The former condition exists whenever the location in a program's \$ERROR table corresponding to an interrupt which occurred, contains a zero address or some other address which is not legal for the program. In either case an informational dump is automatically provided and the operator is given the option of deleting succeeding jobs in sequence with the terminating job.

### C. Termination Specified by the Operator

The operator may specify termination of a job through a type-in on the computer keyboard. The format for the type-in is:

```
TER id *
```



where TER identifies the type-in as a request for job termination, and "id" is the JOB REQUEST ID of the job to be terminated. The asterisk, which is optional, causes the remaining jobs in sequence with the terminating job to be deleted from the schedule.

D. Temporary Interruption of a Program

At any time the computer operator may temporarily halt the execution of a job program with the type-in:

HLT id

where the symbol HLT identifies the unsolicited type-in as a request to temporarily halt a job program, and id is the JOB REQUEST ID of the job to be halted.

Following this the computer operator has two options:

1. Resume execution of the program through the type-in:

PRO id

where id is the JOB REQUEST ID of the job in question and the symbol PRO identifies the type-in as a request to proceed with the execution of the job program.

2. Terminate the program's operation.

### XIII. EXECUTIVE PROCEDURES

#### A. Initialization

The Executive System is stored on a System Tape in absolute form. The Executive is initiated by mounting the System Tape on the appropriate tape unit and actuating the automatic bootstrap facility of the computer hardware. This causes the first block of the System Tape to be read into consecutive core memory locations starting with address zero. The block thus loaded is an absolute load routine, termed the Executive Bootstrap.

The Executive Bootstrap queries the operator through the typewriter as to which of two modes of operation is to begin:

- 1) initial loading of the Executive System
- 2) reloading of the Executive System after the operation of an independent job program

The operator may select the first option by typing in any non-numeric character. The Executive Bootstrap then assumes that the Executive System is to begin from a completely "cleared" condition. The complete Executive System is loaded from the System Tape and control is transferred to it.

If the operator responds with a type-in of a channel and unit number, the Executive Bootstrap assumes that the specified tape unit has been loaded with a tape produced as a result of a previous dump of the Executive System. The Executive Bootstrap responds by loading into core and drum the contents of the tape. The Executive itself completes the load at the point where the Executive Bootstrap is to be overwritten by the incoming data. A check sum test is made to insure that the system is in exactly the same state as when the dump was taken.

When the Executive is initiated, it expects to receive from the operator initializing information such as date, time, and status of facilities. It also expects a specification of equipment which is not currently operable, and a specification of the facilities which are to be reserved for the Executive System.

The information is entered from the card reader or from the operators console. Lines of information from the operators console are limited to 80 characters of data. This information may also be entered or modified during normal operation.

The card or keyboard message types and formats are listed below.

MESSAGE	EXPLANATION
DAT mm/dd/yy HRS tttt	There are two messages for entering time: one for month, day, and year and the other for the hour of day to the nearest minute (0000 to 2400). If no time or date is entered, then the date is not printed and the time begins at 0000.
LOG { CccΔUuu or CccΔAdrum addressΔLlength }	At any time the operator may assign an available or reserved tape unit or drum area for use as a log "tape."
RES { CccΔxuu or CccΔAdrum addressΔLlength } DWN { COREΔAaddressΔLlength }	Definition of down (DWN) or reserved (RES) facilities. The symbol "x" may be 1) U, to denote a read and write unit. 2) R, to denote read only. 3) W, to denote write only. See discussion below.
REL { CccΔxuu or CccΔAdrum addressΔLlength } UP { COREΔAaddressΔLlength }	Release of reserved or down facilities. The message format is as described above. See below.

Facilities may be reserved by the operator at any time. If the facility to be reserved is in use, then the operator is notified of its current assignment by the type-out:

$$\left\{ \begin{array}{l} \text{Ccc}\Delta\text{xuu} \\ \text{or} \\ \text{Ccc}\Delta\text{Adrum address}\Delta\text{Llength} \\ \text{or} \\ \text{CORE}\Delta\text{Aaddress}\Delta\text{Llength} \end{array} \right\} \Delta \left\{ \begin{array}{l} \text{P}\Delta\text{job request id}\Delta\text{T} \\ \text{or} \\ \text{E}\Delta\text{LOG} \\ \text{or} \\ \text{L}\Delta\text{library name}\Delta\text{j} \end{array} \right\}$$

where P denotes that the specified facility is used by a job program.

T denotes that the facility is being transferred to another job.

E denotes that the facility is used as the Executive System log tape.

L denotes the use of the facility to store a Program Library

j may be the JOB REQUEST ID of the job program using this Program Library, or the word "LOAD" if a job program is being loaded from the specified library.

The reserve requests, except those for drum or core, are remembered and the facility is reserved when it is released from its current assignment. At this time, EXEC types the message

RES CccΔxuu

Reserved facilities may be released via the REL message or

1. by assigning the facility to EXEC for use as a log tape (LOG message)
2. by declaring the facility to be inoperative (DWN message)
3. by assigning a Program Library to the facility (see EPL message below)

If a facility becomes inoperative and this condition is detected by the I/O Functional Routines, then the operator must either confirm or negate the condition in response to a message from the I/O Functional Routines. If the operator detects this condition himself, the DWN message may then be used to inform the Executive System. EXEC will immediately mark the equipment as "down" and the operating job program that uses this facility will be notified by completing its last I/O Execution Packet with a downed facility error status.

If a log tape is down, the operator may specify a new log tape assignment.

If a Program Library facility goes down the operator may specify a new assignment for the Program Library.

The UP message causes the Executive to change the status of the object facility from "DWN" to available. If the facility specified was not in the "DWN" status the message:

$$\left( \begin{array}{l} \text{Ccc}\Delta\text{xuu} \\ \text{or} \\ \text{Ccc}\Delta\text{Adrum address}\Delta\text{Llength} \\ \text{or} \\ \text{CORE}\Delta\text{Aaddress}\Delta\text{Llength} \end{array} \right) \Delta \left( \begin{array}{l} \text{P}\Delta\text{job request id}\Delta\text{T} \\ \text{or} \\ \text{E}\Delta\text{LOG} \\ \text{or} \\ \text{L}\Delta\text{library name}\Delta\text{j} \end{array} \right)$$

is typed. No other action is taken.

Three messages concerning definition of program libraries can also be given to EXEC at time of initiation. Program Library tapes are discussed in paragraph F of this section.

## B. Execution of Rush Jobs

The Executive system includes the capability to permit the running of a RUSH job as soon as is practically possible following the submission of the Job Request.

The following conventions are applicable to the use of the RUSH Job Request:

1. All input/output facilities required by the RUSH Job Request must be available.
2. Only one RUSH Job Request or sequence of RUSH Job Requests will be accepted at any one time.

Upon detection of a RUSH Job Request, the Executive will perform an I/O facility check. If the check fails, the operator is notified:

job program id FAILS RUSH

and the RUSH job will be retained as a candidate and given another facility check when any of the I/O facilities are made available. Following the above type-out the operator may release reserved facilities needed by the RUSH job request. When the required facilities are available, core memory will be assigned to the RUSH job, which is then loaded and initiated in the normal manner.

Core memory is assigned to the RUSH job according to the following rules:

1. A normal core availability check is made. If the required core is available, then it is assigned to the RUSH job. Otherwise,

2. Determine the jobs that must be retired temporarily, to satisfy the core requirements of the RUSH job, and assign core memory normally.
3. Dump on drum or tape job programs identified in rule 2 above, and temporarily retire them from the switching cycle.
4. Load and initiate the RUSH Job.
5. When the RUSH job terminates, the usurped core facilities and job programs are returned to their previous status, and entered into the switching cycle.

The above sequence of events is repeated for each RUSH job in a sequence of RUSH jobs.

### C. Independent Operation of Jobs

Occasionally special jobs may have to be run which can operate best only if they have full control of the computer. These jobs are handled in the following manner. A Job Request specifies an independent job to be run. The job to be loaded and run must be stored on drum or tape in relocatable or absolute format, which can be loaded by the Relative Load Routine. When this type of job is selected as the next candidate for initiation:

1. Job programs currently operating are allowed to terminate normally.
2. The Executive System saves itself, instructions and data tables, the Job Request Schedule, and all logging data on a tape which is rewound with interlock. The identity of the tape is communicated to the operator, who then protects the dump tape.
3. The Relative Load Routine is given control with the parameters necessary to locate and load the program.
4. The Relative Load Routine loads the independent job program and relinquishes total control of the computer to the job program, leaving a resident bootstrap routine in the lower addresses of core memory.

5. When the independent job terminates, it returns control to the resident bootstrap routine which reloads the Executive System in its interrupted status and transfers control to it. If the terminating program destroys or fails to reference the bootstrap routine, control is returned to the Executive by the operator in the manner described under Initialization (paragraph A).

#### D. Facility Transfer Function

The Executive System provides several methods for changing the assignment of facilities. These procedures are initiated by

- 1) the job program
- 2) the computer operator
- 3) the Job Request TRN Card

The ability to change the assignment of facilities permits sequences of functions to be performed. Some examples of these are:

- 1) Successive job programs may stack output on the same tape.
- 2) A basic set of data may be changed successively by several operating job programs.
- 3) Compiler and/or assemblers may transfer assembled jobs and Job Requests to the Executive for processing.
- 4) The computer operator may assign facilities to running programs for processing.

The facilities that may be transferred from one assignment to another are:

- drum areas
- magnetic tapes
- paper tape equipment
- High-Speed Printer
- card equipment
- Console Selective Jump switches

Core memory areas cannot be transferred. However, the contents of a core memory area may be transferred to a drum facility, which may then be transferred.

The console channel cannot be assigned; hence it is not transferable.

1. Program Transfer

- a. Initiation of Transfer: Operating job programs may transfer facilities to each other, to the Executive, and/or to successive jobs in the same sequence.

To request the transfer of a facility, the following sequence is coded:

8	9	FUNCTION	14	15	SUB FIELDS		37
		L	D	B		\$QØ, p, , \$UOP	:
		L	M	J	P	\$B1, \$REL	:

where p is the address of a Transfer Packet which details the transfer to be made. The Transfer Packet, illustrated in Figure 8, consists of four words as follows:

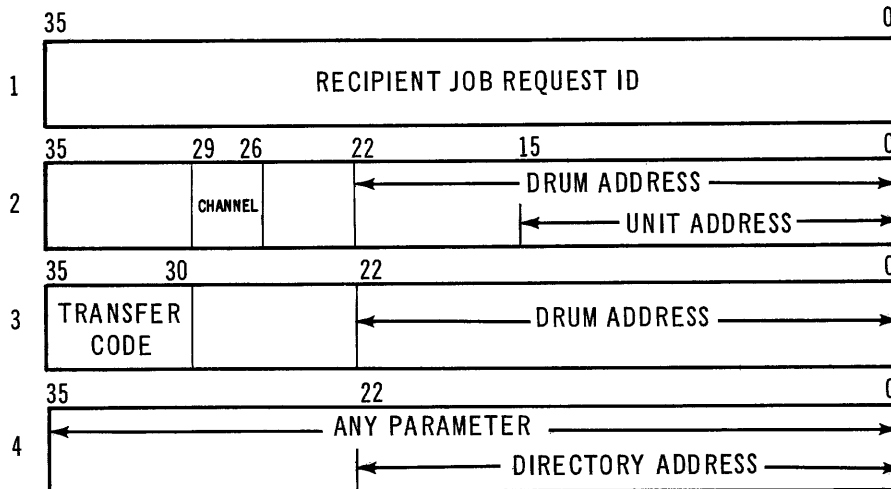


FIGURE 8: TRANSFER PACKET

Word 1: contains the JOB REQUEST ID of the job which is to receive the facility.

WORD 2: contains the channel and unit assignment for the facility or the channel and drum address if a drum facility is being transferred. The unit assignment is denoted by master bit selection.

WORD 3: contains a six bit transfer code which specifies the type of transfer to be made. The possible values of each of these six bits and their explanations are given in Table 7.



WORD 4: may contain any parameter agreed upon by job programs using the facility transfer capability of the Executive. In the special situation where a Program Library on drum is being transferred to EXEC, this word will contain the address of the Program Library Directory Table.

TABLE 7: TRANSFER CODE

BIT POSITION	VALUE	EXPLANATION
35	1	The facility is not a drum facility.
	0	The facility is a drum facility.
34	0	Null
	1	A program Library is being transferred to EXEC.
33	0	Null
	1	Job Requests are being transferred to EXEC.
32	0	Null
	1	The facility is being released by the job program.
31	0	Null
	1	The facility is being transferred to a job program.
30	0	The facility is a write facility.
	1	The facility is a read facility.

The Job Request and Program Library indicators may be set concurrently. In this case, however, the Job Requests must be the initial data on tape.

b. Completion of Transfer: Transfers to the Executive System are completed as follows:

(1) Job Request transfers cause the Executive schedule function to process the requests from the specified facility and then make the facility available.

- (2) Program Library transfers are stored in the Program Library Registration Table.
- (3) Facility release transfers cause the Executive to make the transferred facility available.

Transfers to other job programs are completed when the receiving job program executes the following instructions:

8	9	FUNCTION 14	15	SUB FIELDS	37
		L D P		\$QØ, p,, \$UOP	:
		L M J P		\$B1, \$TRN	:

where p is the address in the receiving program's assigned core area at which EXEC will store the three word Completion Packet. This latter packet is created by EXEC by removing Word 1 of the Transfer Packet. A three word area in the DBANK of the recipient program must be reserved for the Completion Packet.

EXEC will load QØ with the address of the Completion Packet, assign the object facility to the requesting job program and then return control to the address following the above completion calling sequence. If there is no facility for the recipient program, then QØ will contain binary zeros.

The TRANSFER CODE field of the Completion Packet is changed to one of the following octal values which define the type of facility:

VALUE	TYPE OF FACILITY
0	Console Selective Jump Switch
1	UNISERVO IIA tape unit
3	UNISERVO IIIA tape unit
5	High-Speed Printer
10	Magnetic Drum
11	Card Reader
12	Card Punch
15	Paper Tape Reader
16	Paper Tape Punch

- c. Incomplete Transfer: When the recipient job program is deleted for any reason, then all facilities in process of transfer to the deleted job are released to the available pool. The following typeout notifies the operator:

TRANSFERS TO id ARE REL

CccΔUuuΔparameter  
CccΔUuuΔparameter

where id is the JOB REQUEST ID of the job which was to have received these facilities. The parameters are any that may have been agreed upon by the job programs.

## 2. Operator Transfer

The operator may transfer reserved, down, or available facilities to the Executive or to an operating job program. The types of transfers are:

- a. A facility containing a set of one or more Job Requests may be transferred to the Executive.
- b. A facility containing a Program Library may be transferred to the Executive System.
- c. A facility may be assigned to an operating program. The Executive accepts transfer requests from the operator via the console keyboard and stores them in the internal facility transfer table. The request is kept in the table until the recipient job program interrogates the Executive for the presence of a transferring facility. The receiving job program may be in a "Give-up-Control" loop waiting for a facility to be transferred to it. The recommended procedure though is for the program to execute a "TYPE AND READ keyboard" request. If no other work is to be done by the job program this action will effectively remove it from the switching cycle until the operator chooses to answer the "type and read keyboard" message. The operator then may answer the message and inform the program that a facility assignment is being held by the Executive transfer function. The format of the operator's facility transfer request is:

TRNΔidΔCccΔ {<sup>xuu</sup>  
Adrum addressΔLlength} Δparameter

where "id" is the JOB REQUEST ID of the program.

and "parameter" represents any 6-character parameter chosen by the program.

The symbol "x" has the meaning described in paragraph A of this section.

If the facility described by the absolute assignment in the above type-in is not in the reserved, down, or available state then a message describing the current status of the facility is typed out.

3. Job Request Transfer: This type of facility transfer is also restricted to jobs with equal priority and precedence values, and ordered sequence numbers. This function, however, depends upon data on the Job Request Transfer Card (TRN). The TRN Card is necessary in the Job Request which uses facilities that are to be passed on to one or more succeeding jobs in a sequence or to the Executive System. Facilities containing Job Requests and/or a Program Library may be transferred to the Executive System through use of the TRN Card. The TRN Card is described in Section II.

#### E. Assembly (Compilation) and Testing of Programs

In the following paragraphs, the terms "assembly" and "assembler" should be considered synonymous with "compilation" and "compiler", respectively.

The assembly and/or testing of job programs will be handled in the manner described below. The philosophy behind this procedure is two fold. First a minimum of control interface between the assembler and Executive System is desired. This provides more flexibility to assemblers and a less complicated Executive System. Secondly, an environment for job program testing is provided which is almost identical to the operation of proven programs. The procedure is:

1. Job programs are prepared for assembly according to the conventions of the associated assembler. One Job Request to the Executive is necessary to load and operate the assembler.
2. The Job Request(s) for running the assembled programs, and the single object program output tape are transferred to the Executive System via one of the procedures defined in Section XIII.D.
3. All the object programs on the assembler output tape, for which there are Job Requests, will be run in the order specified by the priorities in their respective Job Requests. These Job Request priorities must differ only in their sequence numbers. See Section III.
4. Output tapes (or drum facilities) for stacking of debugging dumps from the jobs being tested are transferred from job to job by use of one of the procedures outlined in Section XIII.D.

The testing of object programs that have been assembled at some previous time is handled in a manner identical to the assembly and test procedure with the exception that a Job Request for the appropriate assembler is not present.

#### F. Job Program Libraries

1. General: It is desirable and profitable to store many programs in ROC format on a single magnetic tape or consecutive group of magnetic drum registers. This greatly facilitates the loading of these programs for execution. The Executive System promotes this useful feature by allowing any named tape or drum facility to be defined as a facility containing a Program Library. Job Requests for programs stored on these named program libraries may show this storage by placing the name of the program library in the MEDIUM NAME field of the PTY Card. The Executive then loads the job program from the referenced Program Library. If the referenced Program Library has not yet been defined, then the Executive System will allocate a tape unit for this purpose and instruct the computer operator to place the named Program Library on the unit. This is achieved via a type-out:

LOAD CccΔUuu WITH library name

The operator acknowledges this message with a "YES" if the named Program Library has been loaded.

"NO" if the named Program Library cannot be loaded.

If a "NO" answer is received, the corresponding Job Request is deleted from the schedule.

The Executive maintains a count of all Job Request references to a named Program Library. This count is reduced by one each time a job program is loaded from the named Program Library. The operator is notified when this count is reduced to zero. The notification message is:

NO REF FOR PL library name

The operator may then replace the Program Library tape on this unit with another Program Library (see below).

This count is kept even though the Program Library has not been loaded. When the Program Library is loaded its assignment will be stored in the previously established item containing the name and count of references.

2. Program Library Definition: Program Libraries may be defined for the Executive System through the use of the internal and Job Request transfer function (see Section XIII.D.), through a reference on the PTY Card to a Program Library not yet defined, and through operator typeins (Sections XIII.A. and XIII.D.).

The operator has three message types at his disposal to handle the assignment and release of program libraries. They are:

- a. Enter Program Library

EPL nameΔCccΔ {Uuu  
Adrum addressΔLlengthΔdirectory}

where "directory" is the address of the drum stored Program Library Directory Table.

At any time the operator may use this message to inform the Executive System that a tape unit or magnetic drum block with a previous status of down, reserved, or available, now contains the named Program Library. If a drum Program Library is defined and the "directory" parameter is absent, the Executive System will assume that its directory is located at the beginning address.

b. Drop Program Library

DPL name

At any time the operator may release a drum or tape facility containing a named Program Library. The Executive releases the facility (tapes are rewound with interlock) and acknowledges action with the message:

DPL name $\Delta$ Ccc $\Delta$  { Uuu  
Adrum address $\Delta$ Llength }

If the named Program Library is not defined in the Executive System, the operator is notified:

PL name NOT DEFINED

c. Replace Program Library

RPL name 1  $\Delta$  name 2

At any time the operator may replace a Program Library assigned to magnetic tape with a new Program Library. The old Program Library tape is rewound with interlock, and the operator action and Program Library assignment are verified by the message:

RPL name1 $\Delta$ name2 $\Delta$ Ccc $\Delta$ Uuu

If the old Program Library is not currently defined, the operator is notified as above. The following message is sent when a drum stored Program Library is specified for replacement:

PL ON DRUM

3. Program Library Format on Magnetic Tape: Each job program must be in one of the formats acceptable by the 1107 Relative Load Routine. The





A Program Library may be stored on magnetic drum in a variety of ways:

- a. A tape-to-drum program
- b. A card-to-drum program
- c. Output from assembler or compilers
- d. Any other procedure which will produce a drum Program Library of the above format.

G. Rerun Function

The Executive System provides a facility by which the operating programs may establish rerun points. The philosophy behind the rerun function provided by the Executive System is:

A rerun is primarily the responsibility of the job program.

A rerun point will be established in a manner and format which allows either the job program to restart itself when trouble is detected, or allows the Executive System to reinitiate the job program at an identified rerun point.

Responsibility for the actual rerun dump and for determining the status of all I/O facilities will remain with the job program.

The Executive will monitor the time of the rerun dump and supply pertinent information from the Executive tables.

1. Establishment of Rerun Point: The job program notifies the Executive of its intention to establish a rerun point by executing the sequence:

8	9	FUNCTION 14	15	SUB FIELDS	37
		L D B		\$QQ, r,, \$UOP	:
		L M J P		\$B1, \$RRU	:

where r is the address in the program's assigned core area at which the Executive will store a rerun table containing information necessary to reset the Executive tables at the time of rerun.

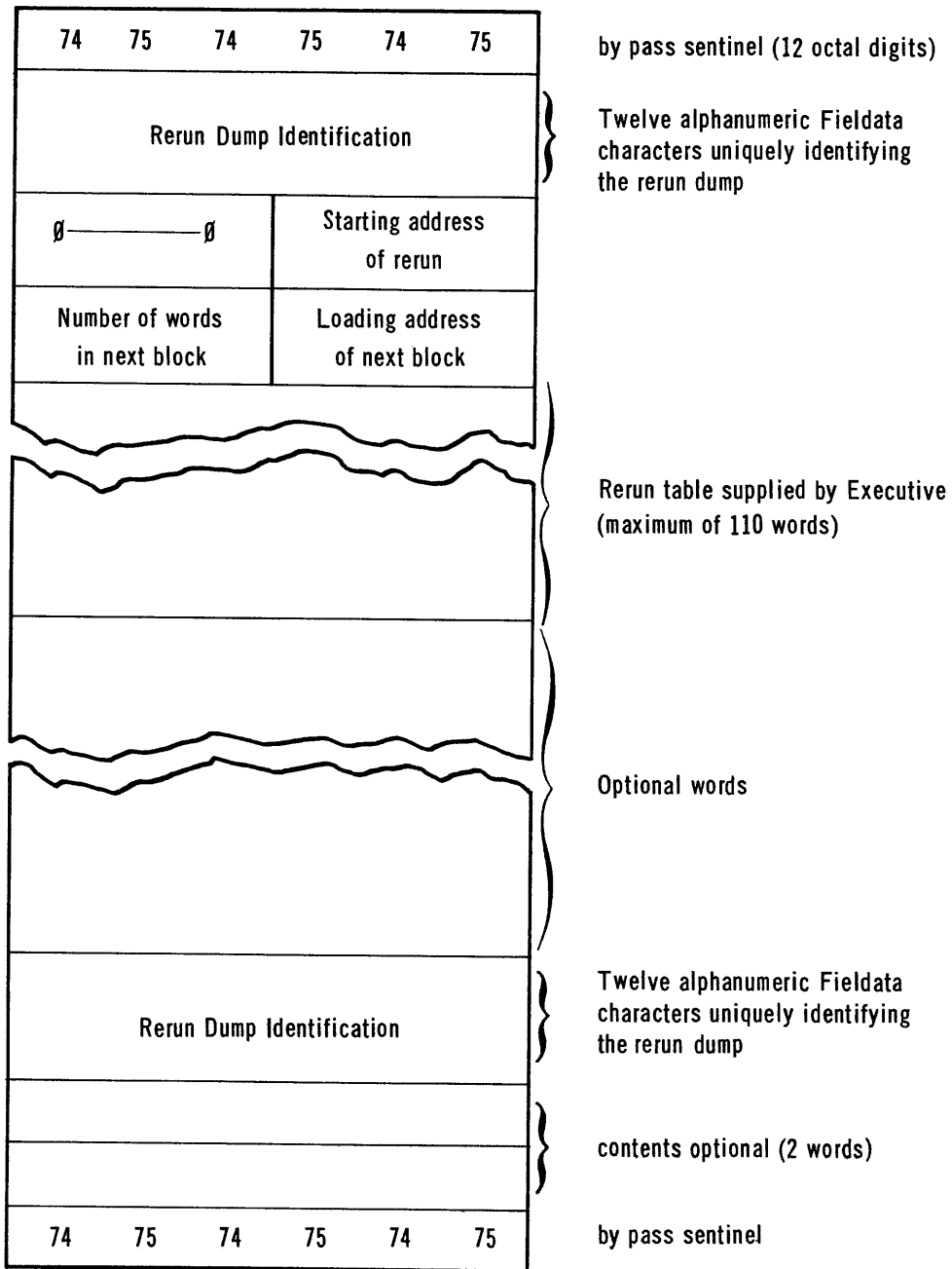
The Executive checks for any outstanding input/output requests from the requesting program. If any are outstanding the message:

## JOB id ILLEGAL RERUN REQUEST

is displayed on the printer and control is returned to the job program. QØ remains unchanged. If there are no outstanding input/output requests then the Executive System will generate a rerun table containing all necessary data to enable the Executive System at some future time to reinitiate this program through the use of a Job Request. This table will be generated at the address specified in QØ. Zero will be stored in QØ to indicate to the program that the rerun table has been generated.

The job program must then use the following procedure to establish a rerun point.

- a. Select a tape from its environment and write a rerun identification block, in the format shown in Figure 10.
  - b. The second block must contain that portion of the job program that contains the rerun procedures for the job program.
  - c. Remaining blocks are formatted according to the requirements of the job program. It is the responsibility of the job program to insure that all other blocks on the rerun tape are different from the rerun identification block in either the first, second, or third word.
2. Initiation at Rerun Point: A job program can be reinitiated from a rerun point in two ways. The first is under control of the job program and the second is under initial control of the Executive and is completed by the job program. The format of the rerun dump is identical for both cases.
- a. Job Program Initiation: The job program locates the identification block, ignores the rerun table generated by the Executive, and reloads the remainder of the rerun dump according to procedures established by the job program itself.
  - b. Executive Initiation: The Executive System accepts Job Requests for programs that are to be initiated at an identified rerun point. The Job Request consists of the PTY Card in the form



*NOTE: The block length is variable from a minimum of 96 words to a maximum of 240 words.*

FIGURE 10: RERUN ID BLOCK

JOB                    RERUN                    RERUN  
REQUEST, PTY,, DUMP, \* , MEDIUM ,, PRIORITY .  
ID                    ID                    NAME

where RERUN DUMP ID is the 12-character alphanumeric identity of the rerun dump.

RERUN MEDIUM NAME is the absolute location of the tape containing the rerun.

The Facility Transfer (TRN) Card is the only other Job Request card allowed in the Job Request for initiating reruns. Reruns in a sequence of jobs must always be the first of the sequence. It follows also that there can be only one rerun in a sequence of jobs.

The procedure used by the Executive System to load and initiate a job at a rerun point is as follows:

- (1) The job is selected according to the rules of selection.
- (2) The operator is notified to load the rerun dump tape on a selected tape unit.
- (3) The bypass sentinel (Figure 10) is used as a search identifier to locate the rerun identification block. The RERUN DUMP ID field of the PTY Card is then compared with the second and third words of the block to verify that it is the rerun dump specified.
- (4) When the appropriate rerun identification block has been located, the data from the rerun table in the block is distributed to the proper Executive tables.
- (5) The next (second) block of the rerun dump is then loaded at the address specified in the fifth word of the rerun identification block.
- (6) Control is then transferred to the job program via a jump to the starting address specified in the fourth word of the identification block.

At this point the job program assumes full responsibility for completing the reload and continuing the run.

After the above procedure is completed the Executive returns to the normal concurrent mode of operation.

## INDEX

### A

Assembly of Programs, 89-90

### C

Chained Communication Requests, 65, 66  
CLAMP (Relative Load Routine), 2, 7, 11, 22, 25, 29, 32, 37, 38,  
83, 92  
Communication Conventions, 70-73  
Communication Execution Packet, 66-68  
Communication, function of EXEC, 4, 65-73  
Communication Functions, 68-70  
Communication Request Parameter, 65  
Communication Requests, 65  
compilation of programs, 89-90  
Completion Packet, 87  
complex programs, 38  
compute-limited job programs, 10, 22, 30, 63  
concurrent operation of job programs, 1, 2, 9, 61-64

### D

Dispatcher, 61, 62  
Dumping Function of EXEC, 4, 75

### E

\$ERROR table, 40, 77  
Execution Packet, Communication, 66-68  
Execution Packet, I/O, 42, 43, 44-52  
Executive Bootstrap, 79

### F

FAC Card, 11, 12, 22, 24, 25-29  
facilities, allocation of, 30-32  
facilities, assignment of, 1, 11, 12, 24-33  
facilities, availability of, 22, 23  
facilities, release of, 32, 33  
facility requirements, changes in, 28-30  
facilities, transfer of, 13-15, 84-89

### I

independent operation of job programs, 9, 38, 39, 79, 83-84  
initialization of EXEC, 79-82  
initiation of job programs, 37  
I/O Execution Packet, 42, 43, 44-52  
I/O Functional Routines, 53  
I/O Functions, Executive System, 52-60

## INDEX

### I

I/O limited job programs, 10, 22, 30, 63  
I/O Request List, 42, 43  
I/O Request Parameter, 43  
I/O Requests, submission of, 43, 44  
Input/Output, function of EXEC, 2, 42-60  
Interrupt, function of EXEC, 2, 40-41, 44, 46

### J

Job Request Cards:  
    FAC, 11, 12, 22, 24, 25-29  
    PMn, 15-17, 37  
    PTY, 6-11, 34, 35, 36, 38, 39, 97  
    TAL, 11, 13, 14, 29, 30  
    TRN, 13-15, 89, 97  
Job Request, definition of, 5  
Job Request, deletion of, 17, 18  
Job Request, format of, 6-18  
Job Request Schedule, 1, 5, 19  
Job Request, submission of, 5, 6

### L

LIBRARIAN, 11, 93  
Loading of job programs, 1, 19, 34-39  
Logging, function of EXEC, 4, 74  
log tape, 80, 81

### M

memory dumps, 75, 76, 77  
memory lockout, 63  
Modification Record, of object program, 24, 30

### P

parameters for job program, 15-17  
PMn Card, 15-17, 37  
precedence assignment to job programs, 10, 20  
priority assignment to job programs, 10, 20  
Program Library, 5, 6, 7, 35, 36, 39, 81, 86, 87, 88, 90-94  
Program Library Directory Table, 93  
PTY Card, 6-11, 34, 35, 36, 38, 39, 97  
\$PARAM table, 37

## INDEX

### R

Real Time Clock, 40, 41, 64  
Request List, I/O, 42, 43  
Request Parameter, Communication, 65  
Request Parameter, I/O, 43  
rerun, 10, 11, 17, 21, 22, 94-98  
RUSH job programs, 20, 82-83

### S

Schedule, Job Request, 1, 5, 19  
Schedule Maintenance, function of EXEC, 1, 5-18  
Selection, function of EXEC, 1, 19-23  
sequence assignment to job programs, 10, 20, 21  
serial operation of job programs, 9, 21  
SLEUTH Assembly System, 7, 21, 24, 28, 29, 48  
SORT/MERGE, 28  
Switching function of EXEC, 2, 44, 61-64  
Switch Lists, 61-64  
System Tape, 79

### T

TAL Card, 11, 13, 14, 29, 30  
termination of job programs, 4, 21, 61, 75, 76-78  
time of job programs, 10  
transfer of facilities, 13-15, 84-89  
Transfer Packet, 32, 85  
TRN Card, 13-15, 89, 97  
type-ins, to EXEC, 5

### V

voluntary release of control by job program, 61



# UNIVAC

DIVISION OF SPERRY RAND CORPORATION