EXECUTIVE ROUTINE - OPR

This document is provisional in nature and is intended as a vehicle for meeting immediate needs with regard to system familiarization and orientation. UNIVAC® Division of Sperry Rand Corporation reserves the right to change and/or modify such information contained herein as may be required by subsequent system developments.

1.0 C O N T E N T S

2.0 INTRODUCTION

OPR is a revised UNIVAC 1050 Executive Routine. It has been necessitated by the addition of peripheral equipment to the 1050 System, which was not contemplated at the time the former Executive Routine (OPS) was designed. OPR contains Executive handling for UNISERVO VIC tapes, Communications Subsystem, and Console Typewriter not available in OPS.

Customers using these devices must use OPR. Those using OPS may convert to OPR, and it will usually be to their advantage to do so in view of the improvements in program switching and interrupt processing. Conversion is accomplished by reassembly of programs, using the OPR oriented systems tape.

2.1 GENERAL DESCRIPTION

OPR is divided into 4 major portions:

A. I/O coordinator.

1. Program Switching
2. Interrupt Priority Routines

B. Tape and FASTRAND order handling and interrupt routines.

C. Program loading and memory allocation.

D. Console typewriter handler.

Eight parameters are provided to enable customers to assemble versions of OPR to suit their system needs.

3.0 ASSEMBLY OPTIONS

OPR is available in source code in the standard library and may be assembled by the user to fit his needs. Parameters may be written as follows:

```
OPR      p1          p2          p3                p4          p5
      Card Type, Tape Type, No. of Programs, Memory Dump, Translation,

         p6                 p7                p8
      FASTRAND Unit, Console Typewriter, Communication
```

Following is a description of each parameter:

3.1 PARAMETER 1, CARD TYPE

80 - Provides for 80 column card loading from row or column readers. Loader will accept 80 column call cards for tape locating.

90 - Provides reading of 90 column call cards from row or column readers, with translation.

NOTE: One of the above must appear as parameter 1. If 90 is used, a translate table is generated for 90 column card code beginning in location 01500.

3.2 PARAMETER 2, TAPE TYPE

A - Provides tape order handling and error recovery for UNISERVO IIIA tapes (up to 6 units).

C - Same as A for IIIC tapes.

6C - Provides for tape order handling and error recovery for UNISERVO VIC tapes up to 16 units, with read-read and read-write overlap.

3.3 PARAMETER 3, NUMBER OF PROGRAMS

CONC - Provides for loading and running of 2 relocatable programs concurrently, or a single absolute program.

SING - Eliminates the portion of OPR which provides for concurrent processing (approximately 1500 characters). Allows for running of any single program.

3.4 PARAMETER 4, MEMORY DUMP OPTION

PDMP - Provides for inclusion of a memory print routine which can be executed by the operator (approximately 1100 characters).

TDMP - Provides the ability to write all memory on tape unit 1 for future printing (65 characters).

NOTE: Parameter 4 is not required.

3.5 PARAMETER 5, TRANSLATION

Since translate tables must be located in the first 4096 characters of memory, absolute locations must be set aside by the Executive Routine for use by relocatable programs. The parameter TRNSn (where n is 1 through 9) provides up to 9 open rows beginning in location 01500 into which translate tables may be transferred and used. In 90 column version of OPR, an input translate table is automatically generated in 01500 which may be used by the worker program, but not disturbed. In these cases, areas provided by the TRNSn parameter begin in 01600.

3.6 PARAMETER 6, FASTRAND UNIT

If parameter 6 is F, the ability to handle FASTRAND orders with error recovery is provided.

3.7 PARAMETER 7, CONSOLE TYPEWRITER

TYP1 - Provides a console typewriter control routine as part of OPR, in which all standard library routine program stops (JD instructions) are converted to typeouts, and operator options are accomplished by typeins rather than by trace switch key ins. An interface is provided which enables worker programs to utilize the typewriter for their own purposes.

TYP2 - Provides only for worker program usage of the typewriter. UNIVAC standard library routines are not affected.

A more complete description of both versions of the typewriter routine and how to program its interface is contained in Section 8.0

3.8 PARAMETER 8, COMMUNICATIONS

If parameter 8 is RT, the Executive is modified to provide priority switching as a result of an interrupt from a communications program, and to provide the interface for COMCO (Communication Interrupt Control Routine).

4.0 I / O COORDINATOR

4.1 PROGRAM SWITCHING

Transfer of control from one program to the other is accomplished when I/O delay has caused release of control to the "link" entry of OPR by an I/O control routine.

Transfer to the other program will be effected immediately. If no other program is running, control will be returned to the original program. All routines which communicate with this portion of the coordinator are prepared to repeat their "ready to go" tests when control is returned. If there is only one program running, the I/O delay will be consumed by the coordinator and the waiting control routine transferring control to one another until the required interrupt occurs.

4.2 CHANNEL STATUS LIST

OPR and the I/O routines which operate with it maintain communication with one another by bit settings in the Channel Status List. The list consists of 1 character for each I/O channel. The least significant 3 bits of each character are used to communicate the following status information:

		ON (1)	OFF (0)
4 bit	I/O Delay	Set by I/O routine when releasing control to OPR.	Set off by OPR when returning.
2 bit	Peripheral running	Set by I/O routine when order issued.	Set off by I/O routine when order completed.
1 bit	Error	Set by I/O routine when error occurs.	Set off by OPR when attempting recovery.

4.3 PRIORITY SWITCHING

In order to increase the efficiency of concurrent processing, and to provide priority transfer of control to a communication program, a priority switch routine in OPR is accessed during peripheral interrupt processing. The function of this routine is to make such alterations as are necessary to assure that the program whose interrupt is being processed is immediately given control at the completion of the interrupt, thereby postponing return to the non-priority program until the priority program releases control. If a communications program is in operation, it will always be the priority program. This requires that it identify itself to OPR during initialization. An entry point is established for this purpose, which is accessed by the Communications Interrupt Control Routine (COMCO).

5.0 TAPE-FASTRAND HANDLER

As determined by Parameter 2, a tape and/or drum handler is included. Appendix A describes basic programming requirements to interface with the OPR handlers if the user is writing his own tape and drum I/O controls. It is assumed, however, that most customers will use standard library tape file control routines, which provide all of the necessary interface coding.

5.1 UNISERVO IIIA AND IIIC HANDLERS

The IIIA and IIIC handlers in OPR are very similar to those produced in the Satellite Coordinator and in OPS. They differ from the VIC handler (see below) in that read-write overlap is not permissable for IIIA and IIIC units. It should be noted that the IIIC handler and the VIC handler are interchangeable from the hardware point of view. Since the IIIC handler uses less memory, it may be advantageous for some users, whose operations do not gain advantage from overlap, to use the IIIC handler for VIC tape units.

5.2 READ-WRITE OVERLAP - VIC TAPES

Since overlap of channels 4 and 5 is permissable with VIC tapes, the handler is equipped to issue more than one order if, having issued one, it finds another on its list for a channel which is not in operation; or, if the first order on the list specifies a channel which is busy, the list will be scanned for an order on a free channel before the handler relinquishes control.

5.3 FASTRAND ROUTINES

A FASTRAND interrupt subroutine is included in the tape handler as an assembly option. FASTRAND I/O routines will communicate with the handler in the same manner as tape routines.

5.4 REAL TIME PRIORITY

The tape handler has the ability to issue tape and drum orders submitted by a communication program. These will be identified at the time they are submitted. The issue subroutine will always attempt to issue priority orders first.

6.0 PROGRAM LOADING FUNCTION

All versions of OPR contain the ability to locate and load programs or segments from a master instruction tape. Card loading ability is dependent on parameter 1 (see 3.1).

6.1 PROGRAM CALL

The program ID may be transmitted to the loader by:

A. A call card which contains $ in column 1, the PID in columns 2-5, MIT unit number in column 6 (if other than 0).

   Columns 7-22 may contain optional information which the loader will leave in AR1 at the completion of a load.

B. Trace switch settings (see Operating Instructions, Section 9.0), if the TYP1 option is not used.

C. Typein of *3$AAAA#, where AAAA is a 4-character PID, if parameter 7 is TYP1.

D. Direct access by program (see Section 7.5), for run-to-run loading.

6.2 SPECIAL CALLS

A blank PID from any source indicates that the next program is to be loaded from the card reader. A PID of ¤¤¤¤ (077777777) indicates that a load is not to be performed; the loader will release control to the coordinator to continue a program already running, if there is one.

If a program ID other than blank or ¤¤¤¤ is received, the locator searches forward on tape unit 0 (unless another has been specified) for a label block (R) containing a matching ID. If a match is not found, the MIT is rewound and the system stops to await further instructions. The MIT is not rewound when a program has been located and loaded.

6.3 PROGRAM MEMORY ALLOCATION

Information in the R block (or card) enables the loader to determine whether or not the program will fit into available memory. In a concurrent system, the first relocatable program is assigned the lowest memory available, and the second is assigned the highest, except where the load key is 5, in which case it is always assigned the lowest.

The memory remains allocated until the program is released or jettisoned.

Absolute programs may be loaded only if no other program is in memory. If an absolute program has been loaded and not released or jettisoned, no other program may be loaded.

A list of the stop displays and procedures to be followed, if the load being attempted is unacceptable, is contained in Section 9.2.

6.4 SEGMENT LOADING

Segments of either relocatable or absolute programs may be loaded from time to time using the methods described above, or a running program may access OPR for the purpose of loading its segment or another run without operator intervention or knowledge. Relocatable segments are always assigned the same base address and memory allocation originally assigned the run. Programming required for segment and run-to-run loading is discussed in Sections 7.4 and 7.5.

7.0 PROGRAM COMMUNICATION WITH OPR

7.1 CLASS II INTERRUPT

A program using decimal arithmetic instructions where the possibility of decimal overflow exists must load the address of the overflow routine into locations 0775-0777. If a class II interrupt occurs which is not an operator request or typewriter interrupt, control will be transferred to that address.

The actual class II interrupt entry channel must not be altered at any time.

7.2 PROGRAM RELEASE

When a program is completed, a JR to the release entry of OPR (0700) must be executed. OPR releases the memory allocated to the finished program and stops. Control is not returned to the program which has been released.

The release entry should not be accessed unless all processing has been completed.

7.3 PROGRAM STOP

In order to bring the computer to an orderly halt, all I/O orders currently being executed must be completed and their interrupts processed. This is accomplished by a JR to the Stop routine (0736). OPR retains control until all pending I/O interrupts have been processed.

The Stop routine must be accessed before executing a JD or JHJ instruction. If the hardware configuration includes a console typewriter, and the TYP1 option is used, Section 8.0 should be consulted regarding use of the Stop routine.

7.4 SEGMENT LOADING

A running program may access OPR for the purpose of loading a segment by performing the following steps in the order shown:

A. Execute the proper Stop routine sequence.

B. Set locations 0541 and 0542 to non-blank. This prevents OPR from stopping (0541) and reading a call card (0542). If it is intended that the segment ID be obtained from a call card, or the console, this step and following step C are not performed.

C. Store the ID (4 characters) in AR2.

D. If the tape unit number on which the segment appears is other than 0, store the appropriate unit number in 0540.

E. Execute JR 0612. The segment will be located, loaded, and executed. Locations 0540 through 0542 will be reset to blank at completion of the load.

7.5 RUN-TO-RUN LOADING

Prior to executing Program Release, a user program may direct the loading of a following run by following steps B through D above, before executing JR 0700.

A suggested method of chaining programs involves the use of columns 7-22 of the call card. As indicated in Section 6.1 above, this field in the card may be used to transmit information to a program being loaded. Thus, the call card for program A might contain in columns 7-22, the PIDs for programs B, C, D, and E. Program A may temporarily store the PIDs for B, C, D, and E until it is ready to release. Then using the above method it directs the loading of

program B. Prior to JR 0700, the PIDs for C, D, E may be stored in the loader's card image area, location 01006-01021, for subsequent use by program B.

7.6 TRACE ROUTINE

Information may be entered into memory from the operator console by using the console trace switches, when trace mode is set on PROC. The routine used by OPR for this purpose is a closed subroutine and is available to worker programs (when TYP1 option is not used) as follows:

A. Execute a display stop (JD) informing the operator of the need for a trace key in of 2 characters.

B. Execute JR 0600.

C. When control is returned, the two characters that have been set in the console trace switches will be in the least significant characters of AR2.

7.7 TRANSLATE TABLES

If translate table areas are included in the Executive Routine being used, they will begin in location 01500. Translate tables should be transferred into the areas as they are used, since they are not preserved when switching programs in concurrent operations and may be overlayed by a concurrent program.

7.8 JETTISON BY PROGRAMS

Programs occasionally encounter circumstances which require them to abort without having come to an orderly completion of their processing. In these cases a program may release its memory allocation by storing 077 into location 0530 and executing JR 0700. See Section 7.2.

8.0 CONSOLE TYPEWRITER

Following is a description of, and programming instructions for, the console typewriter control routine.

For purposes of this description, all programs can be divided into two basic types:

A. Programs or subroutines which are standard software provided by UNIVAC; Standard Library Routines.

B. Programs or subroutines which are coded for a particular installation or configuration; User Programs.

8.1 OPTIONS

Two versions of the typewriter handler are available. One utilizes the typewriter for both standard routines and user programs (TYP1); the other utilizes the typewriter for user programs only (TYP2). Thus, the following sections, 8.2 through 8.3, are applicable only if TYP1 is used.

8.2 TYPEOUTS FROM STANDARD LIBRARY ROUTINES

8.2.1 Changes Required in Existing Programs

Standard routines are programmed to run both with or without a typewriter. They contain a fixed sequence of instructions. These instructions will be such that prior to executing a JR 0736, class III interrupt is inhibited and a character is stored into a fixed memory location (0647) within OPR. The information to be displayed is loaded into the three least significant characters of AR1. All 18 bits are edited; the most significant bits preceding meaningful information are zeros.

The character is stored into location 0647, and has the following values:

1 This is an informational display. No action, or response is expected from the operator. Control will be returned when the typeout has been initiated (or the display executed).

2 This display requires a response from the operator. This program will not again receive control until the operator acknowledges.

When control is returned to the line following the JR 0736, the answer will be edited into AR2, in the same manner as is done when trace switches are used when a typewriter is not present.

Upon return, a JR to the trace routine (0600) is executed. Thus the information from the operator is available following that line, regardless of the presence of a typewriter.

0 This negates the actual type or display request. However, all interrupts will be cleared and control returned to the line following the JR.

The fixed sequence for the above key is as follows:

Load display information (except key 0) into 3 LSC of AR1 and execute:

```
JC $+5, 26                  inhibit class III (busy return line)
SC 0647, (0, 1, or 2)       store key 0, 1, or 2
JR 0736
```

For all keys, upon return, the following is destroyed or altered:

High, low, equal, unequal indicators
4 LSC of AR1
Class III program inhibit is released

For Key 2: 5 LSC of AR2 or 5 LSC characters and 72 depending upon the response (see 8.2.4).

A key 0 or 2 request must not be submitted during interrupt processing. No requests may be submitted if a hardware class III interrupt has occurred and not been reset.

8.2.2 Typeouts from Standard Routines-Typewriter Present (TYP1)

The Stop routine is replaced with the typewriter handler. If the typewriter is not busy, it will accept a JR 0736 as a request to type out. It will edit the 18 bit display code into ASCII code, supply the proper sentinels, and initiate the typeout.

The handler precedes the message with a 3 if an answer is expected, or with a 4 if no response is expected.

8.2.3 Response Not Expected

If the prestored character indicated that an operator response is not expected, control will be returned to the line following the JR 0736, and the typeout will be completed with no further concern by the initiating program.

8.2.4 Response Expected

If the code indicated that an operator response is expected, control will not be returned until the answer has been received. The response is expected to be 4 characters long, preceded by a key. If the key is a space or zero, the next four characters will be edited as an octal number and delivered to the program as 12 bits right justified in AR2, preceded by 3 characters of binary zeros. Four characters must be typed in, unless a 0000 response is expected in which case the sentinel (#) may be typed instead of the key.

If the key is a ($) all five characters will be translated to XS-3 and delivered as five characters right justified. In this case low order location 0110 is destroyed.

The translate table will be available for user programs if TYP1 is used. (ASCII XS-3) will be the last table in the operating system (see Section 3.7).

8.2.5 Typewriter in Use

If the typewriter is in either the process of typing out, or of accepting a typein, the acceptance of a request to type out will be delayed. This will be accomplished by returning immediately to the JC $+5, 26 (busy return) in the requesting program. This will cause a loop until the typewriter is free.

8.2.6 Real Time and Priority Program Considerations

The fact that a typeout may be delayed for some period of time (many seconds) means that, even if the delayed program is a priority program, control will not be switched to it. The reason is that it, in effect, already has control and is merely waiting for the acceptance of its typeouts, and possibly for a typein. (See 8.5 of this document.)

Also, if the non-priority program is delayed because it is waiting for an operator response, the priority program will not have control switched to it until the response is received.

8.2.7 Displays from Standard Routines - Non-Typewriter Usage

When the typewriter is not used, the Stop routine will test to see if there was a prestored character other than zero. If there was, all interrupts will be cleared. Then the information in AR1 will be taken and a JD executed with this information.

Because of the possibility of multiple entries due to display requests during interrupt processing, the Stop routine will list these requests. They will be processed on a first in - last out basis.

8.3 PROVIDING FOR DISPLAYS FROM USER PROGRAMS - TYPEWRITER ABSENT

If no character other than zero store is present, interrupts will be cleared and control returned to the line following the JR 0736. This allows user programs which have been running with OPS to run with OPR without making any changes to current logic.

8.4 TYPEOUTS FROM USER PROGRAMS

For those programs which are written specifically for use in typewriter systems, an alternate method of typing out is provided. It should be noted that this is an alternate method, and that the method so far detailed for standard routines may be used by user programs if preferred.

The advantage of using the alternative method is that the information may be alphanumeric and of unlimited length. Carriage returns within the message are the user's responsibility.

This type of request may not be issued during interrupt processing.

8.4.1 Initiating the Typeout

The user program initiates the typeout by loading AR1 with the address of the start of the message and executing a JR with class III inhibit to the typewriter handler at location 0654.

```
FT START, 03          address of 1st character to AR1
JR 0654,  26          initiate typeout
                      return upon acceptance
```

If the typewriter is busy, the acceptance will be delayed until the typewriter is free.

8.4.2 Message Format

The message must be in ASCII code.

The first five characters of the output message must be as follows:

1	2	3	4	5
111110	001101	111110	001010	110001
>	\	>	?	(
Reserved	Carriage Return	Reserved	Line Feed	Key

The first four characters cause a carriage return and a line feed.

The fifth character of the message must be octal 61. This key will be used by the handler to type out a 1 or a 2, to identify the typeout as originating from the low or high program. This is needed for the operator to enable him to type in. (See Section 8.5.)

The message must be followed by two end-of-transmission characters, plus a completion sentinel. These are of the form:

1	2	3
111110	000100	XXXXXX
Reserved	EOT Character	Completion Sentinel

Characters 1 and 2 are required by the hardware to terminate the typeout, and will be tested for by the handler as each character is typed out. The completion sentinel is a software convention to enable the program to determine when its typeout has been completed.

When a typeout is accepted, control is returned to the line following the JR instruction, and the message is typed from the User Program's memory. When the last character of the message has been typed out, the sentinel character (111111) is stored.

8.5 TYPEINS

Four typein keys are provided:

0 = Typein to OPR (same as operator request)

1 = Typeins to the low program (or single program)

2 = Typeins to the high program

3 = Typeins which are a response to a typeout from Standard Routines

Keys 0 and 3 apply to the TYP1 version only. The others are applicable for both TYP1 and TYP2.

The operator may at any time initiate a typein by typing an asterisk (*). This will cause a carriage return and a line feed. He must then type in one of the above keys and proceed with his typein.

Characters will be accepted until the operator terminates the message.

Hitting the number sign (#) indicates completion of the typein and causes the typein to be processed.

When the typein has been accepted either as a result of the number sign or reaching the memory sentinel (see Sections 8.5.1 and 8.5.2), a carriage return and a line feed will be executed.

Key zero (OPR Request) requires no typein until the operator is notified that OPR is ready for his typein, at which time 070007 will be typed out. He will then respond with his request using key 3.

8.5.1 Typein Area

Associated with each typein key is a reference address of a character in the typein area. This area is arranged as follows:

	- 3	
	- 2	
	- 1	
reference character address	0	object of JR instruction after completion of typein
	1	
	2 - 11	user's option (2 instructions)
	12 - n	typed in message
	n + 1	sentinel character supplied by program

A suggested method of using this area is as follows:

```
            INDIC   J                   Typewriter Indicator
                    SC      TYPE+4,00   Set typein switch
                    J       INDIC       Exit

typein      INPUT   AREA    5           5 character message
area                +1      077         memory sentinel
```

In the above example TYPE would be an instruction in the main chain of the program as follows:

```
TYPE JC   GO, 070    Jump if typein has been received
```

At the label GO is the coding for processing the typein which is available starting at INPUT.

The address INDIC+3 is the reference point used by the handler, and is supplied for keys 1 and 2 in the following manner:

It is absolutely essential that the following three instructions be executed by each program before the operator attempts to typein to that program.

```
SC 0530,077
FT label, 03
JR 0666,26
```

In the previous example, label would be INDIC+3.

It is possible for a run to change its typein area while running, but care should be exercised that the operating instructions tell the operator when he can type in.

The high, low, and equal indicators will not be preserved when the above is done.

8.5.2 Completion of Typein

When the operator has typed in his desired message, he hits the sentinel key (number sign #). The handler will recognize this as the completion of the typein and execute a JR to the first line of the typein area. This subroutine is executed with class III and class II interrupts inhibited and must be very brief. It should be used to set a switch in the program's main chain, to enable it to process the typein at a later time. The following may be used by the subroutine without being restored:

- The three least significant characters of AR1
- IR1
- High, equal, low, unequal indicators

If anything else is to be used, it must be saved and restored before returning to the handler. No I/O orders or typewriter requests may be made. Control must be returned to the handler.

The program can protect its memory against the operator typing in too many characters by supplying the indicated sentinel character at n+1. If this character is encountered, the typein will be treated as if a # had been typed.

In the case of standard routines the response to their typeouts will be with key 3. The typein will be edited for them as described in Section 8.2.4.

8.5.3 Cancelling the Typein

The operator can cancel the typein at any time by hitting the plus sign (+); however, it must be done before the memory sentinel is hit. If the + were to go into that position, the typein will be accepted.

Cancelling the typein will not affect the characters already typed.

The operator must start the typein again by hitting the *.

8.5.4 Character Usage Restrictions

The greater than (>) symbol can never be typed out, since the ASCII code for this symbol is the code for the reserved control character.

A plus sign (+) can never be part of an input message, since this is interpreted to mean cancel the input message.

A question mark (?) should never be part of an input message. If a question mark were to be typed in and the operator cancelled the message, when he started over, this character would be in the input area and would be treated as the memory sentinel.

9.0 OPR OPERATING INSTRUCTIONS

The instructions which follow assume that all displays are JD instructions, and that operator communication with OPR is achieved by trace switch settings and Operator Request.

However, if TYP1 is used, the displays are typed out, and the operator communicates with OPR by typing in, rather than by trace settings. Generally, where these instructions call for a trace key in, the same value will be typed in preceded by *3 and followed by #. The correct typein is noted below in these cases. Since the computer is not stopped when TYP1 is used, a typein of *3# accomplishes the same thing as depressing the START button. These are also noted below. Other variations are noted where applicable.

9.1 INITIAL LOAD

The master instruction tape is mounted on tape unit 0. Using the tape load facility, OPR is loaded and displays 070001. Following is a description of the displays which may be encountered, their causes and the action to be taken:

9.2 DISPLAY STOPS (OR TYPEOUTS)

070001 Ready to Load

a. To load using call card, or preset PID, depress START button (*3#). Call card contains $ in column 1, PID in columns 2-5, tape unit number of MIT, if other than 0, in column 6, optional data in columns 7-22.

b. To load using trace switch setting, set first 2 characters in trace switches, depress OPERATOR REQUEST button, then the START button. After stop 077000, set second 2 characters in trace switches, depress START button.

c. To return to program already running without performing load, set ID of 077777777 in trace switches, as in (b) above.

d. Type in appropriate ID as *3$PPPP#(077777777 is produced by typing in *3$""""#).

070002 No R card. Depress START button (*3#) to repeat load. Display 070001 will be accessed. Correct input.

070003 Not enough memory available. Procedure same as 070002.

070004 Trying to load 3 programs. Procedure same as 070002.

070005 Card read is not call card. Procedure same as 070002.

070007 Program stopped by Operator Request or key 0 typein. To attempt error recovery and/or continue running, depress START button (*3#).

To exercise the following options at this time if TYP1 is not called: depress OPERATOR REQUEST button, set trace mode on PROC, and enter the appropriate key into the trace switches as follows, then depress START button.

If TYP1 is being used, type in the appropriate key preceded by *3 00 and followed by #. Do not depress OPERATOR REQUEST button.

77 Load program. Display 070001 will be accessed.
76 Print memory.
75 Dump all of memory on tape unit 1 (see 9.3 below).

00 Jettison program using FASTRAND unit.
01 Jettison program using FASTRAND unit.
02 Jettison program using tape unit 0.
03 Jettison program using tape unit 1.
04 Jettison program using tape unit 2.
05 Jettison program using tape unit 3.
06 Jettison program using tape unit 4.
07 Jettison program using tape unit 5.
10 Jettison program using tape unit 6.
11 Jettison program using tape unit 7.
12 Jettison program using tape unit 8.
13 Jettison program using tape unit 9.
14 Jettison program using tape unit 10.
15 Jettison program using tape unit 11.
16 Jettison program using tape unit 12.
17 Jettison program using tape unit 13.
20 Jettison program using tape unit 14.
21 Jettison program using tape unit 15.
22 Jettison program using Printer Channel 0.
23 Jettison program using Reader.
24 Jettison program using Punch.
25 Jettison program using Channel 3.
26 Jettison program using Channel 7.

NOTE: Great care should be exercised in making the above settings. Incorrect key ins which are less than 026 may cause unrecoverable problems.

070010 Absolute program load is unacceptable (see Section 6.3). Procedure same as 070002.

070104 Check sum error. Key 1 into location 0 to ignore (*3 0001#). Not recommended unless cause is positively known. Otherwise, procedure same as 070002.

070105 Card or block count error in load. Procedures same as 070104.

070106 Read error during load. Depress CLEAR button for channel on which error occurred, then depress START button (*3#). Loader will return to display 070001.

If a segment is being loaded from tape, the loader may attempt to restart the load while the MIT is rewinding in which case the error stop will be repeated. Wait for rewind to be completed, then repeat procedure.

070707 PID not found on tape. MIT rewinding. Wait for rewind, then depress START button (*3#) to try again.

073636 Entry to typewriter routine, TYP1 not called. Unrecoverable.

077000 First half of trace switch ID acknowledged. Key in other half, depress START button.

070013 Card load being attempted in 90 column system. This version of OPR does not contain a card loader.

077776 Tape abnormal during tape memory dump. Rewind tape unit 1 and depress START button (*3#) to try again.

077777 Tape memory dump completed. Depress START button (*3#) to access display 070007. Program(s) may be continued from this point.

9.3 TAPE MEMORY DUMP

If the OPR in use contains the tape memory dump feature, it may be accessed by a trace switch key in of 075 (*3 0075#) at the appropriate time, following an operator request display (070007). A blank tape should be on tape unit 1 at load point. All of memory is written in 1024 character blocks and the computer is stopped (077777). The tape is not rewound. Depress START button to return to normal operation. The standard library contains the routine TDMP which may be loaded using the normal call procedure for the purpose of printing the memory as written on tape unit 1. The printout obtained is in the same format as the Print Dump routine.

9.4 PRINT MEMORY DUMP

The print memory dump, if provided in the system, produces an octal printout directly from memory and returns to display 070007 when completed. It is accessed by a trace key in of 076 (*3 0076#) following the operator request stop. In view of its high memory requirements, the use of the print dump option is not recommended except for the initial stages of debugging.

9.5 PERIPHERAL ERROR RECOVERY

Each of the I/O control routines used with OPR contains its own error displays indicating the nature of the error which has occurred and identifying the channel, unit, etc. Following is the procedure to be used to attempt error recovery after a peripheral error stop. (Note exception for tape-single program.)

A. Depress START Button (If no TYP1)

If a second program is in memory and is not affected by the error, it will continue processing. If there is no second program, OPR will loop until the operator intervenes. The program in which the error occurred will be bypassed until the condition is corrected.

B. Correct the error condition if possible.

C. Depress OPERATOR REQUEST Button (*0)

The operator request display (070007) will be accessed.

D. Depress START Button (*3#)

If the error condition has been properly corrected, the program(s) will be resumed from the point of error. If not, the error display will reappear, and the procedure must be repeated.

It should be noted here, that a retry will be attempted on each peripheral which has an error condition existing at this time whether or not an attempt has been made to correct it.

Exception: If the single program version of OPR is being used, the above procedure does not apply to tape errors. If a tape error occurs, recovery is attempted immediately, when the START button is depressed after the error display. Control remains in the tape error recovery routine until the situation is corrected.

9.6 PROGRAM JETTISON BY OPERATOR (CONCURRENT PROCESSING ONLY)

The jettison procedure enables the operator to release the memory allocation of a program which is unable to continue; and to replace it with another program and/or continue a program which has been running concurrently.

Normally, this will be done when an unrecoverable peripheral error has occurred and the program is unable to proceed to its normal conclusion and release.

Following is the procedure:

A. Depress OPERATOR REQUEST button (*0) to access display 070007.

B. Enter the jettison code (see above, Section 9.2) of any one of the peripherals being used by the program being jettisoned into the trace switches. If the jettison code is entered by typewriter, the following 3 steps are unnecessary.

C. Set trace mode to PROC.

D. Depress OPERATOR REQUEST button.

E. Depress START button.

The program will be released and display 070001 will appear, and any of the options listed in Section 9.2 may be exercised.

There is not a jettison procedure in the single program versions of OPR. If a program cannot run to normal completion, OPR must be reloaded in order to substitute another program.

9.7 OPERATOR REQUEST

The varied reasons for using the OPERATOR REQUEST button are described above. Following are some general remarks regarding its operation:

- The button must be lit when it is depressed in order to take effect. There are times when, during the running of OPR and the peripheral control routines, it must be inhibited (light out). Sometimes this is obvious when looking at the console, but usually the inhibit periods are so brief that the light seems to be lit continuously or may be flickering. If depressing the button has no effect, it was probably inhibited at the instant it was depressed. Hesitate, then try again.

- If a program contains an error which causes it to enter a loop which does not involve I/O processing, OPR will never be able to secure control in order to process an Operator Request. This condition will usually be apparent in that the processor will be running (looping) and no I/O peripherals will be running. In order to obtain a memory dump and jettison the program:

  A. Depress PROGRAM STOP button

  B. Set top row of console switches to 3000073100_8

  C. Set Display-Alter section to INST

  D. Depress ONE INST button

  E. Depress CLEAR button

  F. Depress ALTER button

  G. Depress START button

  H. Depress CONT button

  I. Depress START button

  If the computer does not now display 070007, depress OPERATOR REQUEST button (*0). Follow memory dump and/or jettison procedure. If a second program had been running before the problem developed, it may now be continued.

- When programs are running normally with OPR, use of the OPERATOR REQUEST button (*0) is the only safe way to stop the computer without risking the loss of I/O images. Use of the PROGRAM STOP button is not recommended.

- If TYP1 is used, the operator request function is accomplished by a typein of *0. However, if the OPERATOR REQUEST button is depressed while programs are running, the result is the same, i.e., display 070007 is typed out and OPR will await further instructions.

10.0 ASSEMBLY INSTRUCTIONS

OPR source code is a part of the standard library and may be assembled using the parameters in Section 3.0, by preparing three cards as follows

LABEL	OPERATION	OPERANDS
PID	BEGIN	0520
	OPR	p1, p2, p3, p4, p5, p6, p7, p8
	END	START

Any 4 character PID may be used in the BEGIN card. However, Systems Programming has assigned names to 10 versions for future reference accordingly. They are:

Name	Parameter Combinations
OPA1	80, A, SING, TDMP
OPA2	80, A, CONC, TDMP
09A1	90, A, SING, TDMP, TRNS1
09A2	90, A, CONC, TDMP, TRNS1
OPC1	90, C, CONC, TDMP, TRNS1
OPC2	90, C, CONC, TDMP, TRNS1
09C1	90, C, CONC, TDMP, TRNS1
09C2	90, C, CONC, TDMP, TRNS1
OPA4	80, A, CONC, TDMP,,F, TYP1, RT
OPC4	80, C, CONC, TDMP,,F, TYP1, RT
09A4	90, A, CONC, TDMP,,F, TYP1, RT
09C4	90, C, CONC, TDMP,,F, TYP1, RT

APPENDIX A: INTERFACE WITH THE TAPE HANDLER

The user who wishes to write his own file control routines will find it necessary to communicate with OPR and the tape handler. This must be done in the specific manner which is discussed below. Deviations from the prescribed manner would be ill advised, and no responsibility on the part of Systems Programming will be assumed for any problems which might therefore arise.

1. INITIALIZATION

Before using the tape handler to issue any orders, the following lines must be executed:

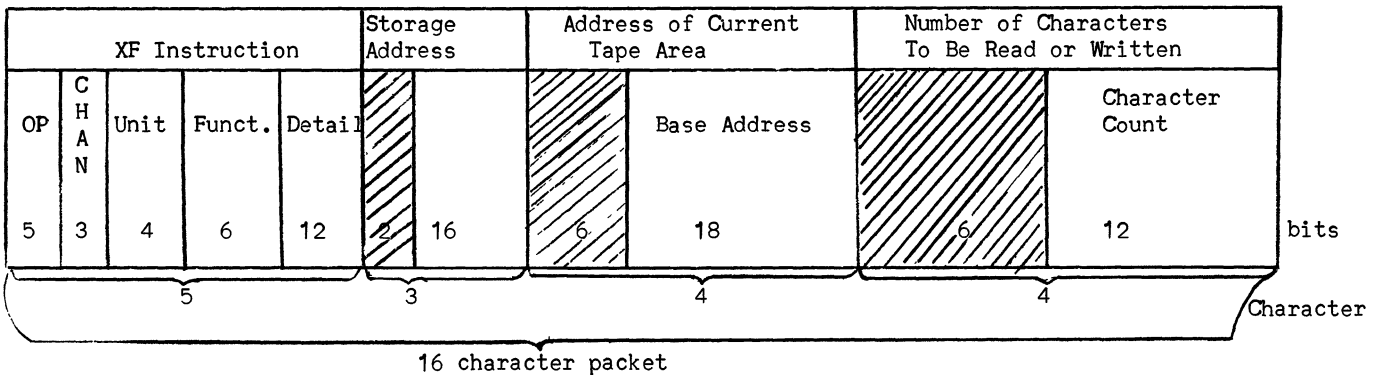
```
FT | 2+n, 7
JR | 0666
```

where n is the number of the tape unit to be referenced in the tape orders. This pair of instructions must be repeated for each new tape unit used. For example, if the tape unit he is using is number 5, the instructions would read as follows:

```
FT | 7,7
JR | 0666
```

2. ISSUING THE ORDER

When the user wishes to issue a tape order he must execute a JR 0750 with the tape order and associated parameters already placed in AR1. The format of the 16 character packet of information that is placed in AR1 is as follows:



The shaded areas represent bits that do not contain information for the handler. They should be preset to zero.

The tape instruction (XF) must contain all the necessary bits to execute the desired tape instruction in appropriate mode, density, direction, etc. The address of the tape area to be read into or written from appears in the field base address. This information will be transferred by the tape handler to either tetrad 48 or 52 as specified by the channel bits. The number of characters to be read or written must appear in the field marked character count. This will be transferred by the handler to tetrad 49 or 53 as necessary.

The address which must appear as the storage address must be the right hand address of a 4 character control field in the users program. The handler will store pertinent information, concerning the status of the order, into that field. The character at storage address minus three (known as the completion status character) should be preset to binary 0. For a read order, one of the following sets of information may be placed in the storage field by the handler:

Storage Field		Meaning
1 Character	3 Characters	
01	Last address ± 1 (depending on direction of read) into which information from tape has been transferred.	Successful Completion
02		End of Tape
03		Tape Mark detected
07		Block bypassed due to error conditions (see Special Options).

As long as the completion status character remains at 0 the user knows his order has not been finished. In a concurrent system the user must give up control to OPR while waiting for the completion of his order.

An example of the necessary instructions to issue an order and to test for completion follows. Let us assume the following lines appear in the user's constant area and will be used to specify the order:

```
LABEL    OP      OPERANDS
         XF      01,0,5,4        Tape Order
         +3      COMPL           Completion storage address
         +4      TAREA           Area label
ORDER    +4      100             Block size
```

The coding to issue the order and test for completion follows:

```
         PD      COMPL-3,1
         BA1     ORDER,16        Bring order to AR1
         JR      0750            Execute tape routine
TEST     CC      COMPL-3,0
         JS      DONE            Test for completion
         LS      0524,04         Set 4 bit channel status list
         JR      0724            Go to OPR
          J      TEST            Go to test completion
DONE     CC      COMPL-3,03      Tape mark test
         JE      END
          J      MAIN
```

For a write order, one of the following sets of information will appear in the completion storage field:

Storage Field		Meaning
1 Character	3 Characters	
X1	ignore	Successful completion
X2	this	End of tape
X4	information	Parity Error Zero

The X will contain a number from 0 to 4 which will indicate the number of times it was necessary for the handler to issue a skip erase order before completing the write specified. The coding to test for completion must, of course, take cognizance of the variability of these three most significant bits.

3. SPECIAL OPTIONS

In addition to the Standard Tape order which has an octal 41 as most significant character, there are some additional options available. These options allow loader orders to be issued through the framework of the tape handler, priority orders to be specified as such to the handler, special 1st block handling and persistent errors of certain types to be ignored. For a loader order the leading octal digit must be a 6; a priority order is signified by a 5. The following bit conditions affect the second octal digit of the leading character. To specify that the order to be issued is a first block order, the 2 bit should appear as a 1. To specify that the handler should ignore certain errors associated with this order, the 4 bit should be a 1.

If the 1st block condition is specified when the order is given to the handler, and an error occurs which cannot be bypassed, the operator is notified of this special condition by a display, and he is then at liberty to move or replace the tape. If, on reading, a tape parity error occurs and cannot be corrected with successive attempts and the special "ignore error" bit is set in the order, the order is stripped from the list; the tape positioned past the bad parity point, and the user notified with a 7 stored as the MSD of tetrad 50 which is then stored in the specified storage address.

In the case of parity errors while writing, a count of the number of skip erase orders given in attempting to issue an order is kept and stored for the user. The maximum number of times tried will be five. The count is stored in the three MSB's of the successful completion character. Thus if this character equals 31, the user knows that to obtain completion 3 skip erase orders were given before the order was successfully completed. When this character equals 50, the user knows that the issuance of this order was unsuccessful and that the tape handler tried 5 times to skip-erase and issue.

APPENDIX B: SUMMARY OF ABSOLUTE LOCATIONS FOR PROGRAM COMMUNICATION

Location	Reference	Description
0520-0527	4.2	Channel Status List
0530	7.8	Jettison Switch, Typewriter Initialization
0531	4.3	COMCO Identification Character
0532, 0533	4.3	COMCO High, Low Switches
0534-0537	6.0	PID of Last Program or Segment Loaded
0540	7.4	MIT Tape Unit Number
0541	7.4	OPR Stop Switch (1 = Bypass Stop 070001)
0542	7.4	OPR Call Card Switch (1 = Bypass Call Card Read)
0545-0547	4.1	Switch Return, Channel 0.
0550-0552	4.1	Switch Return, Channel 1.
0553-0555	4.1	Switch Return, Channel 2.
0556-0560	4.1	Switch Return, Channel 3.
0561-0563	4.1	Switch Return, Channel 7.
0564-0566	6.3	Highest Location in Program A.
0567-0571	6.3	Lowest Location in Program B.
0572-0574	6.3	Highest Location in OPR, +1 (Lowest Location in Program A).
0575-0577	6.3	Highest Location in Memory.
0600	7.6	Trace Routine Entry. Access by JR.
0612	7.4	Loader Entry. Access by JR.
0624-0642		Peripheral I/O Issue Addresses. Same Sequence as Switch Returns.
0647	8.3.1	Typewriter Request Character.
0654	8.5.1	Typewriter Entry. Access by JR.
0666	APP. A	I/O Initialize Entry. Access by JR.
0700	7.2	Release Entry.
0712	4.3	Entry to Priority Interrupt Switch Routine.
0724	4.1	OPR Switch Routine (Link)
0736	7.3	Stop (Typeout) Entry.
0750	APP. A	Tape-Drum Handler Entry.
0762-0777	7.1	Decimal Overflow Load Entry.
01000-01244	7.5	Load Image Area.
01300-01377		Temporary Storage of Vital Tetrads Prog. A. (Concurrent Only).
01400-01477		Same as above, Prog. B.
01500	7.7	Beginning of Translate Tables per P4.
01500+# of Trans. Tables		Temp Storage of AR's; Prog. A, Prog. B.

APPENDIX C: USING THE SORT WITH OPR

When employing the Sort routine with OPR instead of OPS, the following changes must be made to the calls generating the various sort phases:

WITH OPS	WITH OPR
SORT1	SRRT1
SORT2	SRRT2
SORT3	SRRT3
SORT4	SRRT4
SORT5	SRRT5

All other parameters when running under OPS still apply for sorts run under OPR control.