

SOFTWARE OPERATING PROCEDURES



SIO SUBSYSTEM OPERATION

PREREQUISITE SOP MODULES:

Introduction (5951-1369)
Front Panel Procedures Module
Software Input/Output System Configuration (5951-1374)

RELATED MANUALS:

HP Assembler (02116-9014)
HP FORTRAN (02116-9015)
HP ALGOL (02116-9072)



11000 Wolfe Road
Cupertino, California 95014

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

SIO SUBSYSTEM OPERATION

INTRODUCTION



The HP Assembler and the HP FORTRAN and HP ALGOL compilers are computer language processors which translate source programs into machine instructions for computer execution. The Cross-Reference Symbol Table Generator, Punch/Verify Routine and SDUMP are utility programs. All are subsystems of the SIO (Software Input/Output) operating system, and are run under the control of SIO.

Data transfer between the subsystems and peripheral devices is handled through noninterrupt Input/Output routines called SIO drivers. These drivers are loaded into memory before subsystem execution begins.

The following procedures give explicit instructions for loading, configuring and running the subsystems mentioned above in the SIO operating system.

The procedures are:

Procedure 1: HP Assembler or Extended Assembler.....	SSO-2
Procedure 2: Cross-Reference Symbol Table Generator.....	SSO-6
Procedure 3: HP FORTRAN.....	SSO-9
Procedure 4: HP ALGOL.....	SSO-13
Procedure 5: Punch/Verify Routine.....	SSO-16
Procedure 6: SDUMP.....	SSO-22

PROCEDURE 1

HP ASSEMBLER AND EXTENDED ASSEMBLER

The Assembler or Extended Assembler converts assembly language source programs into object programs by translating each assembly instruction into one machine language instruction.

One reading of the source program by the assembler is a pass. If a teleprinter is used without another output device such as a high-speed punch or lineprinter, then assembly is completed in three passes (if both binary output and a listing are requested). If the teleprinter punches tape and lists separately, or if an extra output device (list or punch) is used, the assembler makes only two passes.

Assembler output (object program tapes, program listings) is controlled through the assembly control statement. The control statement is input to the Assembler as the first statement of the source program (either on the same tape or on a separate tape) or through the terminal keyboard.

The control statement starts with the word "ASMB" followed by one or more of these parameters (in any order), separated by commas.

<u>Parameter</u>	<u>Meaning</u>
A	Process the source program in absolute format.
R	Process the source program in relocatable format.
B	Binary output in the "A" or "R" format.
L	List the source program during pass 2 (or during pass 3 if binary output was selected in a three-pass configuration).
T	List the symbol table at the end of pass 1.
N	Assemble all instructions between the IFN and XIF instructions.
Z	Assemble all instructions between the IFZ and XIF instructions.

PROCEDURE 1

Both IFN-XIF and IFZ-XIF pseudo instructions may be used in the program; however, only one type is selected in a single assembly. If both "N" and "Z" appear in the control statement, the character last listed controls which block of statements are assembled.

Sample Control Statement: ASMB, R, B, L, T produces a symbol table listing after pass 1, a relocatable program object tape after pass 2, and a source listing at the end of pass 2 or 3, depending upon the hardware configuration.

Using the Assembler or Extended Assembler

1. Turn on all necessary peripheral devices.
2. If an appropriately configured Assembler or Extended Assembly tape does not exist, configure the Assembler or Extended Assembler with the appropriate SIO drivers by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374).
3. Load the configured Assembler or Extended Assembler tape into memory using the Basic Binary Loader (BBL) or the Basic Binary Disc Loader (BBDL).
4. Place the source tape in the tape input device and ready that device.
5. If the assembler control statement is to be entered on the source tape (either on separate tape or on the same tape as the source program) set a starting address of 100_8 .

If the assembler control statement is to be entered on the terminal keyboard, set a starting address of 120_8 .

6. Start program execution.
7. If the starting address set in step 5 is 100_8 , the Assembler reads the source tape.

If the starting address set in step 5 is 120_8 , enter the assembler control statement on the terminal keyboard and press the return, then linefeed keys. The assembler reads the source tape (pass 1).

PROCEDURE 1

8. The assembler halts with halt code $1020xx_8$ displayed. Consult Table SSO-1 for the meaning of the halt and follow the procedures outlined for that code.
9. Replace the source tape in the input device. Ready that device and any other needed output devices.

Set switch register bit 0 on, if desired, to suppress leader and trailer which the Assembler automatically provides for the object tape. Otherwise, set bit 0 off.

If both listing and punching are specified on the control statement, and the hardware configuration does not allow listing and punching separately (3-pass operation), set bit 15 on.

If the hardware configuration allows listing and punching separately (2-pass operation), set switch register bit 15 off.

10. Press RUN. To halt pass 2 at the end of the current source line, set switch register bit 1 on. To continue pass 2 without halting at the end of the next source line, set bit 1 off and press RUN. To halt pass 2 at the end of the next source line, leave bit 1 on and press RUN.
11. After the assembler starts reading the source tape, it halts with halt code $1020xx_8$ displayed. Consult Table SSO-1 for the meaning of the halt and follow the procedures outlined for that code.

Table SSO-1
Halt Code Interpretation

<u>Halt Code</u>	<u>Meaning</u>
102011 (xx = 11)	The first pass is complete. The assembler types the symbol table before halting, if specified in the control statement. To begin pass 2, return to step 9.
102023 (xx = 23)	The second of three passes has ended. To continue with pass 3, replace the source tape in the input device, and press RUN. To omit pass 3 and assemble another source program, return to step 4.

PROCEDURE 1

Table SSO-1. Halt Code Interpretation (cont.)

<u>Halt Code</u>	<u>Meaning</u>
102054 (xx = 54)	The assembler has halted because switch register bit 1 was set on. To continue without halting at the end of the next source line, set bit 1 off and press RUN. To continue and halt at the end of the next source line, leave bit 1 on and press RUN.
102055 (xx = 55)	Switch register bit 15 is set on. The assembler halted before printing a diagnostic message. Turn off the tape punch and press RUN. The message is typed and the assembler again halts with 102055 ₈ displayed. Turn the tape punch back on and press RUN. At the end of pass 2, set switch register bit 15 off.
102057 (xx = 57)	The assembler has read to the end-of-tape but still expects more source program. Put the next source program segment tape into the input device and press RUN.
102066 (xx = 66)	The control statement is either missing or illegal. Correct the control statement and return to step 4.
102077 (xx = 77)	End of assembly. Return to step 4 to assemble another source program.

PROCEDURE 2

CROSS REFERENCE SYMBOL TABLE GENERATOR

The Cross Reference Symbol Table Generator (X-REF) processes assembly language source program tapes and lists specified symbols along with the sequence numbers of all statements defining or referring to each symbol. Programs may be on more than one tape. X-REF processes all of the symbols, or a range of symbols (such as symbols starting with the letters A through C only).

If the source program uses the IFN or IFZ psuedo-instructions, doubly defined symbols may appear in the table. X-REF also processes literals. The statement number is always 0000 00 for literals; literal definition is not assigned a statement number. Only the first five characters of a literal, including the =, are cross-referenced. Therefore, = D3156 and = D3157 are listed under = D315. Negative literals are all listed under =D.

X-REF reads source programs from paper or magnetic tape. X-REF checks location 107₈ to see if an SIO magnetic tape driver is resident. If so, X-REF reads the source program from magnetic tape.

Using X-REF

1. Turn on all necessary peripheral devices.
2. If an appropriately configured Cross Reference Symbol Table Generator (X-REF) does not exist, configure X-REF with the appropriate SIO drivers* by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374).
3. Load the configured X-REF tape using the Basic Binary Loader (BBL) or Basic Binary Disc Loader (BBDL).
4. Set a starting address of 100₈.

*For paper tape operation, use the SIO teleprinter driver, and the tape input device driver. For magnetic tape operation, use the SIO teleprinter driver and the magnetic tape unit driver.

PROCEDURE 2

5. Place the source program paper tape in the tape input device, or mount the magnetic tape in the magnetic tape device. The source program must be on the third file of the magnetic tape, if it is used.

NOTE: If the magnetic tape driver is in core, but X-REF is to read from paper tape, memory location 107_g must be set to 000000_g and the paper tape input device driver must be loaded into memory. To use the SIO magnetic tape driver at a later time, location 107_g must be restored or the driver reloaded into memory.



6. Set all switch register bits off.
7. If the entire symbol table is to be printed, set switch register bit 15 off. If only a selected range of symbols is to be processed, set bit 15 on.
8. Start program execution.
9. If switch register bit 15 was set off, the X-REF starts reading the source tape.

If bit 15 was set on, X-REF types.....**ENTER CHARACTER RANGE

Type the range of starting characters for the symbols to be processed by X-REF on the teleprinter keyboard.

Example: If labels starting with the letters A, B, and C only are to be listed, type AC return linefeed on the teleprinter.

If labels starting with the character = only are to be listed, type = = return linefeed on the keyboard.

After the range is typed, X-REF starts reading the tape.

10. Once X-REF starts reading the tape, it halts with the halt code 1020xx displayed, where:

xx = 57 End-of-tape or end-of-file has occurred, but the tape does not contain an END statement (signifying that the source program continues on one or more tapes). Place the next tape in the input device and ready that device, then press RUN (without modifying the starting address).

PROCEDURE 2

xx = 77 The last tape (containing the END statement to signify the end of source program) has been read. Before halting, X-REF prints out the symbol table according to the character range specified. If no symbol exists in the character range specified, X-REF types E before halting.

11. To list the symbol table for the same (or different) tape(s) (any character range desired), return to step 5. When X-REF halts with halt code 102077_8 displayed it returns to a starting address of 100_8 . Resetting the starting address is unnecessary.

PROCEDURE 3

HP FORTRAN

The FORTRAN Compiler or 4K FORTRAN Compiler translates a source HP FORTRAN program into a machine language object program. The FORTRAN Compiler is used with a minimum 8K words of memory, and is composed of two tapes labeled FORTRAN pass 1, and FORTRAN pass 2. The 4K FORTRAN Compiler is used with 4K words of memory only, and consists of four tapes labeled 4K FORTRAN pass 1, 4K FORTRAN pass 2, 4K FORTRAN pass 3 and 4K FORTRAN pass 4.

Compiler output (object program tapes, program listings) is controlled through the compiler control statement. The control statement is input to the compiler as the first statement of the source program (either on the same tape or a separate one). The FORTRAN Compiler (but not the 4K FORTRAN Compiler) also reads the control statement from the teleprinter keyboard, if desired.

The control statement starts with the word "FTN" followed by one or more of these parameters (in any order), separated by commas.

<u>Parameter</u>	<u>Meaning</u>
B	Binary output; punch the program in relocatable binary format suitable for loading by the Basic Control System Relocating Loader.
L	List output; list the source program during execution of pass 1.
A	Assembly listing; list the object program in assembly level language during the last pass.
T	Symbol table; list the symbol table. If both T and A are specified, the first of the two is ignored by the compiler.

PROCEDURE 3

Using FORTRAN Compiler

1. Turn on all necessary peripheral devices.
2. If an appropriately configured FORTRAN pass 1 tape does not exist, configure the FORTRAN pass 1 tape with the appropriate SIO drivers by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374). FORTRAN pass 2 is not configured.
3. Load the configured FORTRAN pass 1 tape into memory using the Basic Binary Loader (BBL) or the Basic Binary Disc Loader (BBDL).
4. Place the source tape in the tape input device and ready that device.
5. To enter the control statement as part of the source program (either on the same tape or a separate one), set a starting address of 100_8 .

To enter the control statement on the teleprinter keyboard, set a starting address of 50_8 .

6. Start program execution.* If starting address 50_8 was selected in step 5, type in the compiler control statement. The first intermediate binary tape is punched. Pass 1 halts with halt code $1020xx_8$ displayed.

where:

$xx = 57$ End-of-tape has occurred, but more source input is required. Place the next source program segment tape in the tape input device and ready that device. Continue processing by starting program execution.

$xx = 77$ Pass 1 has found the END\$ statement, or has compiled five programs. Pass 1 is complete. Rewind the intermediate binary tape.

7. Load the FORTRAN pass 2 tape using the BBL or BBDL.
8. Place the intermediate binary tape in the input device and ready that device.

*See SOFTWARE OPERATING PROCEDURES -- FRONT PANEL PROCEDURES Module for the computer in use.

PROCEDURE 3

9. Set a starting address of 100_8 .
10. Start program execution. If requested in the compiler control statement, pass 2 punches a final binary object tape. Pass 2 halts with halt code $1020xx_8$ displayed

where:

- $xx = 01$ Produce an assembly listing or symbol table listing at this time by placing the intermediate binary tape in the tape input device, readying that device then pressing RUN.
- $xx = 77$ Compilation is complete. Rewind the final binary object tape. To compile another source tape, return to step 3.

Using 4K FORTRAN Compiler



1. Turn on all necessary peripheral devices.
2. If an appropriately configured 4K FORTRAN pass 1 tape does not exist, configure the 4K FORTRAN pass 1 tape with the appropriate SIO drivers by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374). 4K FORTRAN pass 2, pass 3 or pass 4 are not configured.
3. Load the configured 4K FORTRAN pass 1 tape into memory using the Basic Binary Loader (BBL) or the Basic Binary Disc Loader (BBDL).
4. Place the source tape in the tape input device and ready that device.
5. Set a starting address of 100_8 .
6. Start program execution. The first intermediate binary tape is punched. Pass 1 halts with halt code $1020xx_8$ displayed.

where:

- $xx = 02$ End-of-tape has occurred, but more source input is required. Place the next source program segment tape in the tape input device and ready that device. Press RUN.
- $xx = 77$ Pass 1 is complete. Rewind the first intermediate binary tape.

PROCEDURE 3

7. Load the 4K FORTRAN pass 2 tape into memory using the BBL or BBDL.
8. Place the first intermediate binary tape in the input device and ready that device.
9. Set a starting address of 100_8 .
10. Start program execution. Pass 2 punches the second intermediate binary tape and halts with halt code 102077_8 displayed. Rewind the second intermediate binary tape.
11. Load the 4K FORTRAN pass 3 tape into memory using the BBL or BBDL.
12. Place the second intermediate binary tape in the input device and ready that device.
13. Set a starting address of 100_8 .
14. Start program execution. Pass 3 punches the third intermediate binary tape and halts with 102077_8 displayed. Rewind the third intermediate binary tape.
15. Load the 4K FORTRAN pass 4 tape into memory, using the BBL or BBDL.
16. Place the third intermediate binary tape in the input device and ready that device.
17. Set a starting address of 100_8 .
18. Start program execution. If requested in the control statement, pass 4 punches the final binary object tape. Pass 4 halts with halt code $1020xx_8$ displayed.

where:

- $xx = 01$ Produce an assembly listing or symbol table listing at this time by placing the third intermediate binary tape in the input device, readying that device then pressing RUN.
- $xx = 77$ Compilation is complete. Rewind the final binary object tape. To compile another source tape, return to step 3.

PROCEDURE 4

HP ALGOL

The HP ALGOL Compiler translates an HP ALGOL source program into a machine language object program.

One reading of the source program by the compiler is a pass. If a teleprinter is used without another output device such as a high-speed punch or lineprinter, then compilation is completed in two passes (if both binary output and a listing are desired). If the teleprinter punches tape and lists separately, or if an extra output device (list or punch) is used, compilation requires only one pass (listing and punching occur concurrently).

Compiler output (object program tapes, program listings) is controlled through the compiler control statement. The control statement is input to the compiler as the first statement of the source program (either on the same or on a separate tape).

The control statement starts with the word "HPAL" followed by one or more of these parameters (in any order), separated by commas.

<u>Parameter</u>	<u>Meaning</u>
B	Binary output; punch the program in relocatable binary format suitable for loading by the Basic Control System Relocating Loader.
L	List output; list the source program.
A	Object output; list the assembly level program.
P	The source program is a procedure only, not a main program.
S	Sense switch control; read the compiler options through the switch register (ignore the B, L and A parameters following HPAL):

PROCEDURE 4

<u>Switch Register Bit</u>	<u>Control Parameter Equivalent</u>
15	B - punch binary object tape
14	L - list source program
13	A - produce object program assembly - level listing.

If the S parameter is included in the control statement, the B, L, and A options are read by the compiler from the switch register. B, L, and A parameters in the control statement are ignored. The switches are read at the beginning of each program line so that any option can be altered during compilation.

Using HP ALGOL Compiler

1. Turn on all necessary peripheral devices.
2. If an appropriately configured HP ALGOL tape does not exist, configure the HP ALGOL tape with the appropriate SIO drivers by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374).
3. Load the configured HP ALGOL tape into memory using the Basic Binary Loader (BBL) or Basic Binary Disc Loader (BBDL).
4. Place the source tape in the tape input device and ready that device.
5. Set a starting address of 100_g.
6. Set switch register bits 15, 14 and 13 for the desired options, if the S parameter is included in the control statement.
7. Start program execution. If ALGOL types "HPAL??" and then halts with halt code 102077 displayed, the control statement is missing or incorrect. Place the corrected source tape in the input device and press RUN.
8. The relocatable binary tape is punched, if B is specified in the control statement.

If L is specified, a source listing is printed on the list output device.

PROCEDURE 4

If both A and L are specified, the source listing, interleaved with the object code listing, is printed on the list output device.

Error messages are included in any listing printed. If only B (punch binary tape) is specified or if S is specified but no option switches are set on, error messages and the incorrect source instructions are printed.

9. When the compiler halts with halt code 102077₈ displayed, compilation is complete. The starting address is set to 100₈ (by the program). To recompile the same source program (with different compiler control state-ment parameters) or another source program, place the desired source tape in the tape input device and ready that device. Return to step 6.

NOTE: If a teletype only is used without another punch or list device, then only the punch parameter or the list parameters (but not both) may be indicated at one time. To both punch and list in the teletype only environment, make one pass specifying punch only, and one pass specifying list only.

PROCEDURE 5

PUNCH/VERIFY ROUTINE

Punch/Verify Routine (P-V) copies a master tape of any size, frame for frame, regardless of content. Punch/Verify also verifies the copy tape contents against the master tape contents.

This routine requires a tape punch device other than a teleprinter tape punch and also a tape input device. However, separate SIO drivers for those devices are not required; the necessary code for driving the punch and reader is included within the Punch/Verify Routine.

P-V creates a tape copy of a master tape by alternately reading 30 master tape frames (tape characters) and then punching a copy of those frames on tape. To verify the copy against the master, P-V first stores the entire master tape contents in memory. The total number of frames that can be stored depends upon the memory size of the computer. P-V is adjusted by the user for the memory size of the computer used. The following list shows the capacity for various memory sizes.

<u>Memory Size</u>	<u>Frame Capacity</u>	<u>Memory Size</u>	<u>Frame Capacity</u>
4K	7560	20K	40328
8K	15752	24K	48520
12K	23944	28K	56712
16K	32136	32K	64904

Loading Punch/Verify

1. Load the Punch/Verify Routine into memory using the Basic Binary Loader (BBL) or Basic Binary Disc Loader (BBDL).

PROCEDURE 5

2. If the computer memory size is other than 8K, modify memory location 364₈ according to the memory size of the computer. The number to be loaded (through the computer front panel) is shown in the table below:

<u>Memory Size</u>	<u>Octal Number (in location 364₈)</u>
4K	7700 ₈
12K	27700 ₈
16K	37700 ₈
20K	47700 ₈
24K	57700 ₈
28K	67700 ₈
32K	77700 ₈



3. Proceed to either "Punching a Tape Copy" or "Verifying a Tape Copy."

Punching a Tape Copy

1. If Punch/Verify is not already loaded, perform the steps outlined in "LOADING PUNCH/VERIFY."
2. Punch a length of leader on the tape copy.
3. Set a starting address of 100₈.*
4. Start program execution. The computer halts with halt code 102001₈ displayed.
5. Set switch register bits 5 through 0 to the select code of the tape input device.
6. Press RUN. The computer halts with halt code 102002₈ displayed.
7. Set switch register bits 5 through 0 to the select code of the tape punch.
8. Place the master tape to be copied in the tape input device and ready that device.

*See the *SOFTWARE OPERATING PROCEDURES - FRONT PANEL PROCEDURES* module for the computer in use.

PROCEDURE 5

9. Press RUN. P-V reads 30 frames (tape characters) of the master tape. Then the routine punches an exact copy of those 30 frames. If one or more of the 30 frames just copied contains any code (other than just the feed-frames), P-V reads another 30 frames and punches a copy of those. If none of the 30 frames just read contained any code (other than the feed-frame holes), the routine halts with halt code 102002_8 displayed.
10. If the routine halts with 102002_8 displayed, examine the master tape to see if the end of the tape has been reached.
11. If the end of the tape has not been reached, return to step 9. If the end of the tape has been reached, remove the master tape from the reader. Then punch a length of trailer on the copy tape.
12. To verify the tape copy just punched against the master tape, follow the steps outlined in "Verifying a Tape Copy." To punch a copy of another (or the same) master tape, set switch register bit 0 on and go to step 8.

Verifying a Tape Copy

1. If Punch/Verify has not been loaded, perform the steps outlined in "Loading Punch/Verify."
2. If Punch/Verify has been used since it was loaded, set all switch register bits off and skip to step 5. Otherwise, configure Punch/Verify by setting starting address of 200_8 and switch register bit 0 on and continuing with step 3.
3. Start program execution. The program halts with halt code 102001_8 displayed.
4. Set switch register bits 5-0 to the punched tape reader select code.
5. Place the master tape in the reader.
6. Press RUN. The program reads the master tape until a halt occurs with halt code $1020xx_8$ displayed:

$xx = 01$	Examine the master tape. If all of the tape has been read, remove the master tape from the reader and go to step 7. If
-----------	--

PROCEDURE 5

xx = 01 the tape has not been entirely read, return to the beginning
(cont.) of step 6.

xx = 77 Go to "Master Tape Memory Overflow."

NOTE: If all of the master tape is read, but the program does not halt, the tape does not contain enough trailer. Press HALT (or HALT/CYCLE) and place at least eight inches of blank (feed-frames only) tape into the reader. Ready the reader and press RUN. Continue with step 7.

7. Place the copy tape in the reader and ready the reader.
8. Set switch register bit 15 on.
9. Press RUN. Punch/Verify reads the copy tape until a halt occurs with halt code $1020xx_8$ displayed:

xx = 02 The copy tape is correct. To verify another copy tape against the same master tape, return to **step 7**. To read another master tape and verify copies of it, return to step 2.

xx = 55 The copy tape did not verify. Refer to "Comparing Master Tape With Copy Tape Data" to observe the discrepancy.



Master Tape Memory Overflow

1. The master tape being loaded has overflowed memory. Either use a computer with larger memory size, or follow the steps outlined below.
2. Mark the master tape in the reader so that it can be removed and placed back at a later time. Remove the tape from the reader.
3. Place the copy tape in the reader (either at the beginning or the last point marked on the tape) and ready the reader.
4. Set switch register bit 15 on.
5. Press RUN. Punch/Verify reads the copy tape until a halt occurs with halt code $1020xx_8$ displayed:

xx = 02 The copy tape section verified with the section of the master tape currently in memory. If the copy tape was read to the

PROCEDURE 5

xx = 02 end, then Punch/Verify is finished with that tape. If the
(cont.) copy tape was not read to the end, mark the tape in the reader
 and remove it. To verify another copy section with the master
 tape section currently in memory, return to step 3; otherwise,
 go to step 6.

xx = 55 The copy tape did not verify. Refer to "Comparing Master Tape
 With Copy Tape Data" to observe the discrepancy.

6. Place the master tape back in the reader at the last point marked on the tape before it was last removed.
7. Set switch register bit 15 off.
8. Ready the reader and press RUN. Punch/Verify reads the master tape until a halt occurs with halt code $1020xx_8$ displayed:

xx = 01 Examine the master tape. If all of the tape has been read,
 remove the master tape from the reader. Place the copy tape
 in the reader at the last point marked and return to step 4.
 If the tape has not been entirely read, return to the beginning
 of step 8.

xx = 77 Mark the master tape in the reader so that it can be removed
 and placed back later. Remove the tape from the reader.
 Place the copy tape in the reader at the last point marked
 and return to step 4.

Comparing Master Tape With Copy Tape Data

1. The copy tape did not verify with the master tape.
2. Write down the contents of the computer memory address register.
3. Write down the two characters (feed-frames) just read from the copy tape by copying the A-register contents.
4. Set a starting address of 153_8 .
5. Set switch register bit 15 off.
6. Press the SINGLE CYCLE (or HALT/CYCLE) button three times, or INSTR STEP once.

PROCEDURE 5

7. The two characters (feed-frames) from the master tape are loaded into the A-register. Copy the A-register contents and compare them with the contents copied in step 3.
8. Set a starting address equal to the one copied from the memory address register in step 2.
9. To continue verifying the tape, return to either "Verifying a Tape Copy" step 10, or "Master Tape Memory Overflow," step 5.

PROCEDURE 6

SDUMP

SDUMP is a utility stand-alone routine which dumps information from disc to either paper or magnetic tape. SDUMP also verifies the copy dump against the disc contents or loads the copy information back onto the disc, if desired.

Using SDUMP

1. Turn on all necessary peripheral devices.
2. If an appropriately configured SDUMP tape does not exist, configure SDUMP with the appropriate SIO drivers (teleprinter and magnetic tape drivers for magnetic tape; teleprinter, tape reader, tape punch drivers for paper tape operation), by following the steps outlined in the *SOFTWARE INPUT/OUTPUT SYSTEM CONFIGURATION* module (HP 5951-1374).
3. Load the configured SDUMP into memory using the Basic Binary Loader (BBL) or the Basic Binary Disc Loader (BBDL).
4. Set a starting address of 100_8 .
5. Start program execution. SDUMP types a request guide on the teleprinter.....DUMP=D,T[-S][,T[-S]]([]=OPTIONAL)
VERIFY = V
LOAD = L
TERMINATE = T
6. SDUMP requests the lower-numbered octal select code (I/O address) for the disc.....DISC CHNL?
Type the disc select code, followed by pressing return then linefeed.
7. SDUMP then types.....COMMAND
Respond according to the option desired:

PROCEDURE 6

- Option 1 Dump all or part of the disc contents by typing "D," followed by the first, then last disc track-sector numbers to be dumped (step 8).
- Option 2 Verify that the copy created by SDUMP (step 8) is correct by comparing the copy with the disc contents (step 9).
- Option 3 Load the copy (created in step 8) back onto the disc (step 10).
- Option 4 Terminate SDUMP execution (step 11).

NOTE: If magnetic tape is used, SDUMP issues a rewind command to the tape unit during initialization, before and after a verify or load operation, and rewind/standby after T for termination.

8. a. To create a copy of disc contents type... D, T_1-S_1, T_2-S_2 return linefeed

where:

T_1 = starting octal track number for the disc.

S_1 = starting octal sector number. This number may be omitted if the entire track (T_1) is to be dumped.

T_2 = ending octal track number for the disc.

S_2 = ending octal sector number. This number may be omitted if the entire track (T_2) is to be dumped.

- b. For magnetic tape storage, SDUMP dumps the requested tracks and sectors onto the magnetic tape, followed by an end-of-file mark. Return to step 7.
- c. For paper tape storage, SDUMP punches the first two of the requested disc tracks on paper tape, then halts after typing the message.....CHANGE OUTPUT TAPE, HIT RUN.

Tear off the paper tape from the punch and rewind it. Check the punch for adequate paper tape supply. Start program execution.

SDUMP dumps the next two disc tracks onto paper tape. This cycle repeats until the last disc track - sector is dumped. Return to step 7.

PROCEDURE 6

- 9. a. To verify a magnetic tape copy created by SDUMP, place the copy tape in the tape unit and ready the unit. Then type V return linefeed. One file of the magnetic tape is read and verified against the contents of the disc. When SDUMP reads the end-of-tape marker, the program types.....EOT and rewinds the magnetic tape. Return to step 7.

NOTE: SDUMP rewinds the magnetic tape before and after a V command.

- b. To verify a paper tape copy created by SDUMP, place the copy tape in the tape reader and ready the reader. Type V return linefeed. SDUMP reads the tape and verifies it against the contents of the disc. Discrepancies are reported on the teleprinter. After the tape is verified, SDUMP types.....EOT and then asks for another command.

If the dump consists of more than one tape, insert the next tape in the reader and type V return linefeed. Continue to read and verify tapes until the last tape is verified. Return to step 7.

- 10. a. To load a magnetic tape copy back onto the disc using SDUMP, place the copy tape in the tape unit and ready the unit. Type L return linefeed. SDUMP loads one file of the magnetic tape onto the disc, then types.....EOT and rewinds the magnetic tape. Return to step 7.

NOTE: SDUMP rewinds the magnetic tape before and after an L command is executed.

- b. To load a paper tape copy back onto the disc using SDUMP, place the copy tape in the tape reader and ready the reader. Type L return linefeed. SDUMP loads the entire tape contents onto the disc, then types.....EOT and asks for a command.

PROCEDURE 6

10. b. (cont.)

If the dump consists of more than one tape, place the next tape in the reader and type L return linefeed. Continue to load tapes until the last tape is loaded. Return to step 7.

11. To terminate SDUMP execution, type T return linefeed. SDUMP halts after issuing a command to rewind the magnetic tape (if magnetic tape is used).



SDUMP ERROR MESSAGES

The following messages may be printed on the teleprinter by SDUMP:

<u>Statement</u>	<u>Action</u>
STATEMENT ERROR	Retype input statement in correct format.
EOT	The end of the input tape being read has been reached; either load the next tape or go on to the next phase.
CHANGE OUTPUT TAPE, HIT RUN	Two full tracks have been dumped onto paper tape; perform the requested action.
TURN OFF DISC PROTECT, HIT RUN	Set the DISC TRACK PROTECT switch off, then press RUN.
DISC INPUT ERROR	Disc Error Diagnostic, for a Parity, Decode or Abort status after 10 retries. Input sequence repeated on restart.
DISC WRITE ABORT	Disc Error Diagnostic, for an Abort status after a write attempt. Sequence is repeated if restarted.
TRACK <i>nnn</i> (8) SECTOR <i>mmm</i> (8)	Identification information for the Disc Error Diagnostic messages described above. <i>nnn</i> is the octal track number and <i>mmm</i> is the octal sector number where the error occurred.
TAPE/DISC VERIFY ERROR	Disc and tape records do not agree. Disc record is rewritten on restart.

SDUMP ERROR MESSAGES (cont.)

<u>Statement</u>	<u>Action</u>
TAPE CHECKSUM ERROR	The checksum in the tape record does not match the sum computed by SDUMP. Current record is ignored if restarted.
MT ERROR - READ PARITY MT ERROR - EOT, RESTART	Magnetic Tape Errors. Error recovery procedures are completed by driver. Restart to retry sequence.
PARAM ERROR NON-NUMERIC or NON-OCTAL	The Disc Channel Number has been incorrectly specified. Re-enter correct number.

•