PHILCO 2000
OPERATING SYSTEM
SYS

This was the last
SYS manual –

PHILCO closed the
Computer Division
in July 1964

VERSION F3
~~NOVEMBER 1963~~
APRIL 1964

PHILCO
A SUBSIDIARY OF Ford Motor Company
COMMUNICATIONS AND ELECTRONICS DIVISION
Computer Marketing Department • Willow Grove, Pa.

TM-23

Printed in USA 11-63 2.5M-1

PREFACE

This manual provides a comprehensive description of the use and operation of the Philco 2000 Operating System (SYS), and replaces most SYS publications issued prior to October, 1962.

The reader should be familiar with basic programming on the Philco 2000 Electronic Data Processing System and with Philco's Translator-Assembler-Compiler (TAC). A knowledge of ALTAC is also helpful.

The manual includes detailed information, with examples, of the four major SYS areas — control lines, entries, service routines, and library routines.

Published in loose-leaf form, this manual permits insertion of updated pages to be issued as future changes and additions are made to SYS.

The following publications have been superseded by this manual:

TRACE 1, R&D Note 23, issued January 4, 1960.

SEGMENTATION AIDS, R&D Note 28, issued January 18, 1962.

OCTAL-QUATERNARY CORRECTIONS IN SYS, R&D Note 29, issued January 22, 1962.

SYS SERVICE ROUTINE BINDEL, R&D Note 30, issued January 22, 1962.

"FORM" ERRORS DURING SYS OPERATION, R&D Note 31 issued February 21, 1962.

LOADGEN, R&D Note 32, issued February 28, 1962.

DATA, R&D Note 33, issued February 26, 1962.

VERSION D OF THE PHILCO 2000 OPERATING SYSTEM (SYS D), R&D Note 35, issued March 12, 1962.

TACLTC (TAC Language Tape Comparison), R&D Note 36, issued March 12, 1962.

SNAPGEN, R&D Note 38, issued May 15, 1962.

VERSION D2 OF THE PHILCO 2000 OPERATING SYSTEM, R&D Note 39, issued August 10, 1962.

SYS — PHILCO 2000 OPERATING SYSTEM, Program Description (PD-1), issued January, 1961. Also includes the eight appendices of the SYS Description (issued April 26, 1961): SYSAIDE, SYSRPLC, TACSERVS, DELRPL, DATA, JOBSRCH, SNAP, and Modifications to SYS for use with TAC II and ALTAC II (the latter issued August 11, 1961).

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

UPDATED PAGES

Page Number	Date	SYS Version	Page Number	Date	SYS Version
Title Page	APRIL 1964 ~~November 1963~~	F3	3-3. 2	November 1963	F
iii	March 1963	E2	3-3. 3. 0	FEB 64 ~~November 1963~~	F1
v	November 1963	F	3-3. 3. 1	November 1963	F
vi	November 1963	F	3-3. 4	November 1963	F
vii	MARCH 1964 ~~November 1963~~	F2	3-4. 3. 1	March 1963	E2
viii	MARCH 1964 ~~November 1963~~	F2	3-5	January 1963	E
ix	FEBRUARY 1964 ~~November 1963~~	F1	3-5. 1. 1	November 1962	D3
1-5	January 1963	E	3-6. 7	March 1963	E2
1-6	January 1963	E	3-6. 8. 0	November 1962	D3
2-2. 0	January 1963	E	3-6. 8. 1	November 1962	D3
2-3. 1	January 1963	E	3-6. 8. 2	November 1962	D3
2-3. 2	January 1963	E	3-6. 8. 3	January 1963	E
2-3. 4	March 1963	E2	3-6. 11. 0	November 1962	D3
2-5	April 1963	E3	3-6. 11. 1	November 1962	D3
2-6. 0	April 1963	E3	3-6. 11. 2	November 1962	D3
2-6. 1	November 1963	F	3-6. 11. 3	November 1962	D3
3-2. 1. 1	April 1963	E3	3-7	January 1963	E
			3-7.1.1	MARCH 1964	F2
3-3. 1. 0	November 1962	D3	3-7. 2. 3	January 1963	E
3-3. 1. 1	March 1963	E2	3-8. 1. 1	FEB 1964 ~~November 1963~~	F1
3-3. 1. 3	FEB 1964 ~~March 1963~~	~~E2~~ F1	3-8. 1. 2	FEB 1964 ~~November 1963~~	F1
3-3. 1. 4	January 1963	E	3-8. 4	March 1963	E2

UPDATED PAGES (Continued)

Page Number	Date	SYS Version	Page Number	Date	SYS Version
3-8.7.0	November 1962	D3	5-9.8	FEB 64 ~~April 1963~~	~~E3~~ F1
4-3	November 1962	D3	5-9.9	July 1963	E4
4-4	January 1963	E	5-9.10	July 1963	E4
4-20.1	January 1963	E	5-9.11	July 1963	E4
4-20.2	November 1963	F	5-9.12	April 1963	E3
5-1	April 1963	E3	5-9.13	April 1963	E3
5-3.2	November 1962	D3	5-9.14	April 1963	E3
5-3.3	November 1962	D3	5-9.15	April 1963	E3
5-3.4	November 1962	D3	5-9.16	April 1963	E3
5-3.5	November 1962	D3	5-9.17	April 1963	E3
5-6.1	November 1962	D3	5-9.18	April 1963	E3
5-6.5	MARCH 1964 ~~November 1963~~	F2	5-9.19	April 1963	E3
5-6.7	MARCH 1964	F2			
5-6.10	MARCH 1964 ~~November 1963~~	F2	5-9.20	April 1963	E3
5-7.0	April 1963	E3	5-9.21	April 1963	E3
5-9.0	July 1963	E4	5-9.22	April 1963	E3
5-9.1	April 1963	E3	5-9.23	April 1963	E3
5-9.2	April 1963	E3	5-9.24	FEB. 64 ~~April 1963~~	~~E3~~ F1
5-9.3	FEB 1964 ~~July 1963~~	~~E4~~ F1	5-9.25	July 1963	E4
5-9.4	July 1963	E4	5-9.26	April 1963	E3
5-9.5	July 1963	E4	5-9.27	FEB 64 ~~April 1963~~	~~E3~~ F1
5-9.6	July 1963	E4	5-10.0	November 1962	D3
5-9.7	July 1963	E4	5-10.1	November 1962	D3

UPDATED PAGES (Continued)

Page Number	Date	SYS Version	Page Number	Date	SYS Version
5-10.2	November 1962	D3	C-10	November 1963	F
5-10.3	November 1962	D3	C-11	November 1962	F
5-10.4	FEB 1964 ~~November 1962~~	~~D3~~ F1	C-12	November 1963	F
5-10.5	FEB 1964 ~~November 1962~~	~~D3~~ F1	C-13	November 1962	F
5-10.6	FEB 1964 ~~November 1962~~	~~D3~~ F1	C-14	November 1963	F
5-10.7	FEB 1964 ~~November 1962~~	~~D3~~ F1	C-15	November 1963	F
5-10.8	deleted ~~November 1962~~	~~D3~~	C-16	November 1963	F
6-2.1	January 1963	E	C-17	November 1963	F
6-3.0	November 1962	D3	Index Tab (Appendix E)		E
A-1	November 1963	F	E-1	January 1963	E
A-2	July 1963	E4	E-2	January 1963	E
C-2	November 1963	F	E-3	January 1963	E
C-3	November 1963	F	E-4	March 1963	E3
C-4	November 1962	F	1	November 1963	F
C-5	November 1963	F	2	FEB 1964 ~~November 1963~~	F1
C-6	November 1963	F	3	November 1963	F
C-7	November 1963	F	4	November 1963	F
C-8	November 1963	F	5	November 1963	F
C-9	November 1963	F	6	November 1963	F

SECTION I

GENERAL DESCRIPTION

The Philco 2000 Operating System (SYS) is an assembly of executive routines designed to provide efficient use of computer run time by standardizing operating procedures, and by reducing the necessity for operator intervention.

The System provides automatic and rapid sequential processing of all programs which are compiled and run on a Philco 2000 computer, using a minimum of eight on-line magnetic tape units or one on-line paper tape unit and seven on-line magnetic tape units. These programs include TAC and ALTAC compilations, and all Running Program Language (RPL), Absolute Binary (ABS), and Relocatable Binary (REL) programs which are being run, tested, and/or corrected.

AN AID TO OPERATORS

At most computer installations, such operations as loading programs into memory, rewinding tapes, and creating post-mortem dumps must be performed continually. SYS relieves the operator from manually performing these duties, thus substantially reducing over-all running time.

AN AID TO PROGRAMMERS

SYS provides many specialized routines to aid the programmer in running and debugging programs. Moreover, SYS Entries, or internal subroutines, may be used to reduce such programmer tasks as repeating numerous instructions to the operator, or calling external subroutines.

AN AID TO OPERATIONS

Additional time saving for computer centers is gained as SYS can accept in succession any type of source language program (i.e., TAC or ALTAC) to be compiled and run. SYS also provides memory dumps or snapshots in the event of program failure, and automatically supplies accounting records (in conjunction with the Accounting Clock) for more efficient bookkeeping.

SYS reduces tape-handling time by permitting input programs to be stacked (placed in consecutive order) on the same tape prior to compilation or running. Similarly, object programs (those which have been compiled) and edited outputs can be stacked on their respective output tapes.

GENERAL SYS OPERATIONS

SYS is on magnetic tape during the compilation or running of a program. Its most basic operations are also contained within memory at all times to direct the operation of programs. As other sections of SYS (not within the basic operations) are required, they are read as one-block-length routines into memory from tape and are executed. These 128-word routines are called shuttle blocks, and, in general, only one such block is contained in memory at any one time. Therefore, SYS usually requires only 512 words of memory (locations zero through 777_8), leaving the remainder of memory to the programmer's use.

In addition to the magnetic tape which contains the SYS program, other tapes may be used to contain:

- the program to be run or compiled
- input data to be processed
- edited output data
- a library of standard routines and subroutines
- temporary storage of program information and data.

SYS TAPE ASSIGNMENTS

In order for SYS to operate at maximum efficiency, specific tape assignments must be used for the various programs. A brief description of these assignments appears in the table on page 1-3. A more detailed explanation of individual tape assignments is contained in Section II, page 2-1.

COMMUNICATION WITH SYS

Communication between SYS and the programmer may be made by Control Lines, SYS Entries, SYS Service Routines, and TAC Library Routines for SYS.

Control Lines

Control Lines, which specify SYS operations to be performed, are written by the programmer in standard TAC instruction format. As distinct from SYS Entries and Library Routines below, which are called from within a user's program, control lines are submitted outside the program. Each control line contains a control instruction and its parameters, and causes such actions to be performed as:

- reading, writing, or rewinding tapes
- loading programs into memory and transferring control to them
- stopping computer action or transferring control to a location in memory
- signifying the types of debugging memory dumps desired.

SYS TAPE ASSIGNMENTS

TAPE	USE DURING COMPILATION AND PROGRAM RUN TIME
0	Unassigned: may be used by a programmer for additional input or output data tapes, library tapes, TAC language program tapes, or RPL tapes.
1	SYS Program Tape: contains the Philco Operating System (SYS) program, the TAC and ALTAC compilers, and the many utility routines within SYS.
2	SYS Intermediate Tape: contains the RPL, ABS, or REL program just after compilation; contains intermediate information specified by dump and snap parameters during run time.
3	Compiler Scratch Tape: used as a working tape during TAC and ALTAC compilations and by some of the utility routines during run time.
4	User's Program Tape: used to collect binary programs (RPL, ABS, and REL) resulting from the compilation process.
5	SYS Output Tape: the tape on which most information for off-line printing and punching is placed; information written on the tape identifies jobs, provides accounting and debugging information, and contains Code-Edits and binary program cards (if any) of programs just compiled.
6	Compiler Scratch Tape: (the same as Tape 3).
7	TAC Library Tape: contains the various subroutines, macro-instructions, and generators which are available to the programmer.
8	SYS Input Tape: contains the various jobs to be compiled and/or run.
9-15	Unassigned: (the same as Tape 0).

SYS Entries

SYS Entries, which permit the programmer to use various routines stored within the basic operations of SYS, provide another means of communication. The programmer may call these routines by writing a transfer of control within his program to a desired entry. Each entry is a location within SYS which contains a transfer of control to its respective routine. The actual locations of these entries within SYS are defined by the SYS subroutine SYSDEF (page 4-1).

The entries include provisions for such operations as:

- initialization of SYS
- post-mortem dumps
- Console Typewriter type-outs
- reading, writing, and rewinding tapes
- scanning routines for the Address and Remarks field of a TAC-format card.

Service Routines

A third means of communication is provided by SYS Service Routines, which are generally called by writing a control line that specifies the designated service routine as one of its parameters. These routines are specialized programs designed to aid the programmer in performing routine functions of tape maintenance, debugging, and program monitoring.

Library Routines

Library Routines, the fourth method of communication, include special subroutines and generators applicable only to programs run under SYS control. These are called by the programmer in the same manner as any TAC subroutine or generator.

OPERATING PROCEDURES

To use SYS, it must first be loaded into memory, tapes must be mounted on their proper units, and SYS must be initialized (refer to SYS Entry 1SYSIN, page 4-3). Normally, initialization is required only once for a series of SYS jobs.

SYS Jobs

A SYS job is a combination of a program to be compiled and/or run under control of SYS, and the necessary SYS control lines.

Sources of Control Lines

Following the initialization, the source of control lines is typed on the Console Typewriter by the operator using the CONIN instruction (page 3-2.2). Such sources may include magnetic tape in either Code Mode or Image Mode format, paper tape, or the Console Typewriter.

The main function of CONIN is to change the mode of input. At the option of a particular installation, SYS can be modified so that a specific control line source can be assumed without the use of a CONIN instruction.

Running SYS

These control lines, entries, and calls for service routines, subroutines and generators submitted to SYS are executed sequentially, causing the designated operations to be performed.

SYS Output

Transmission of information from SYS to the programmer and operator is effected via type-outs on the Console Typewriter and print-outs on the High-Speed Printer. Both types of communication are concerned largely with debugging information. Information from the High-Speed Printer in particular may describe computer actions about to begin, or just completed, and may provide error indications.

MANUAL CONVENTIONS

The following conventions, definitions, and special SYS symbols are presented at this point so that the programmer may become familiar with them prior to their appearance in the manual.

MEMORY CONCEPT

In this document, the beginning of memory is referred to as Location Zero, while the high-order locations are referred to as the end of memory.

SENTINEL BLOCK

A sentinel block is defined in this manual as a 128-word block of 1024 identical alphanumeric characters (usually Z) written on tape. Only the first 120 words of the block are inspected, however, by SYS during a sentinel search. Break characters are never included in the configuration of a sentinel block produced by the System.

BREAK CHARACTER

A break character is one of four designated characters used to separate parameters within SYS control lines. It may be a comma (,), period (.), semicolon (;), or slash (/). No distinction is made between these break characters except where otherwise noted in this manual.

PROGRAM IDENTITY (ID)

A program identity (ID) is defined as 16 or fewer alphanumeric characters which are used as a unique configuration to label a program. Spaces are significant.

CODE MODE

Code Mode is that format which results from the card-to-tape conversion of cards punched in Hollerith code into Philco 2000 Code by the Punched Card Controller. If Code Mode is indicated, SYS assumes that data has been converted to Code Mode, 10 words per card, 12 cards per block. (Refer to Philco Codes, Appendix B.)

HOLLERITH-CODE MODE

Hollerith-Code Mode is the same as Code Mode.

IMAGE MODE

Image Mode is that format which results from the card-to-tape process that transcribes the twelve rows of each column of a card as 12 binary digits onto tape. Punches are transcribed as one's. If Image Mode is indicated, SYS assumes that data has been transcribed in Image Mode, 20 words per card, six cards per block. Image Mode format may be in either Hollerith-Image Mode or Binary-Image Mode.

HOLLERITH-IMAGE MODE

Hollerith-Image Mode is that format which results from Image Mode processing of cards punched in Hollerith code.

BINARY-IMAGE MODE

Binary-Image Mode is that format which results from Image Mode processing of cards punched in binary form. Binary cards may be read only in Image Mode.

DUMP

A dump is that action which results in the editing of information in designated memory locations and subsequent transmission of the information onto magnetic tape for off-line printing on the High-Speed Printer.

POST-MORTEM DUMP

A post-mortem dump is a dump which takes place after a program has been terminated. All dumps signified by the DUMP control instruction (page 3-7.1), and the SYS entries Location Zero, 1ERRDMP, and 1SUBERR (pages 4-2, 4-4, and 4-5) are post-mortem dumps.

SNAPSHOT DUMP

A snapshot dump is a dump which occurs at designated locations during the running of an object program, as specified by the SNAP control instruction (page 3-7.2). A post-mortem dump must be executed to have the snapshot dumps edited and placed on Tape 5 for printing on the High-Speed Printer.

CATASTROPHIC DUMP

A catastrophic dump is a post-mortem dump which may be taken in the event SYS is destroyed in memory. (Refer to pages 3-7 and E-4.)

SPECIAL SYS SYMBOLS

The following list presents a brief description of frequently-used symbolic locations within SYS.

1CONLIN — The first word of a ten-word memory area in which the current control line is stored in Hollerith-Code Mode. (Refer to 1NXTCRD, page 4-19.)

1CONBIT — A 48-bit indicator word which is used predominantly by SYS for internal control. (Refer to 1NXTCRD, page 4-19.)

1NTRYJA — A single memory location, the left half of which contains a jump instruction. The address is normally a location in an object program to which control is to be transferred from SYS.

1WRD1 and 1WRD2 — Two memory locations in which the parameter output of the 1SCAN and 1SCANON subroutines is stored. (Refer to pages 4-15 and 4-16.)

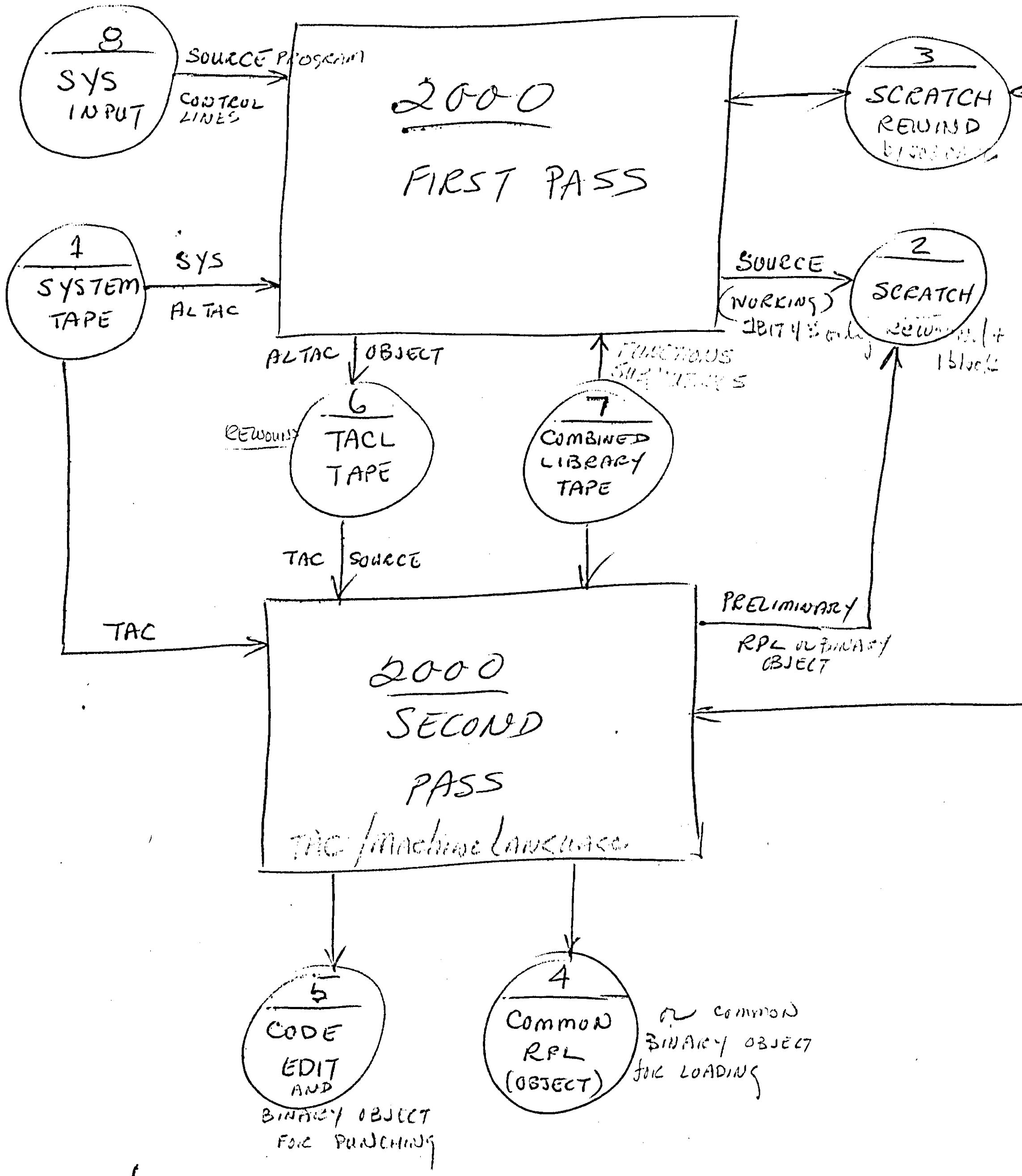
1CHARCT — A single memory location in which is stored, in the left address portion, the number of characters recognized by the 1SCAN and 1SCANON subroutines.

1IBIT1 — A single memory location which is set by the IBIT control instruction and may be interrogated by a program. (Refer to page 3-8.1.)

COMORG — (COMmon ORiGin for REL programs): A single memory location which contains the origin of the Common area in the left address portion and zero or the origin of the last REL program loaded in the right address portion. (Refer to COMMON, page 3-4.5; ORIGIN, page 3-4.4; JOB, page 3-2.1; and REL, page 3-4.3.)

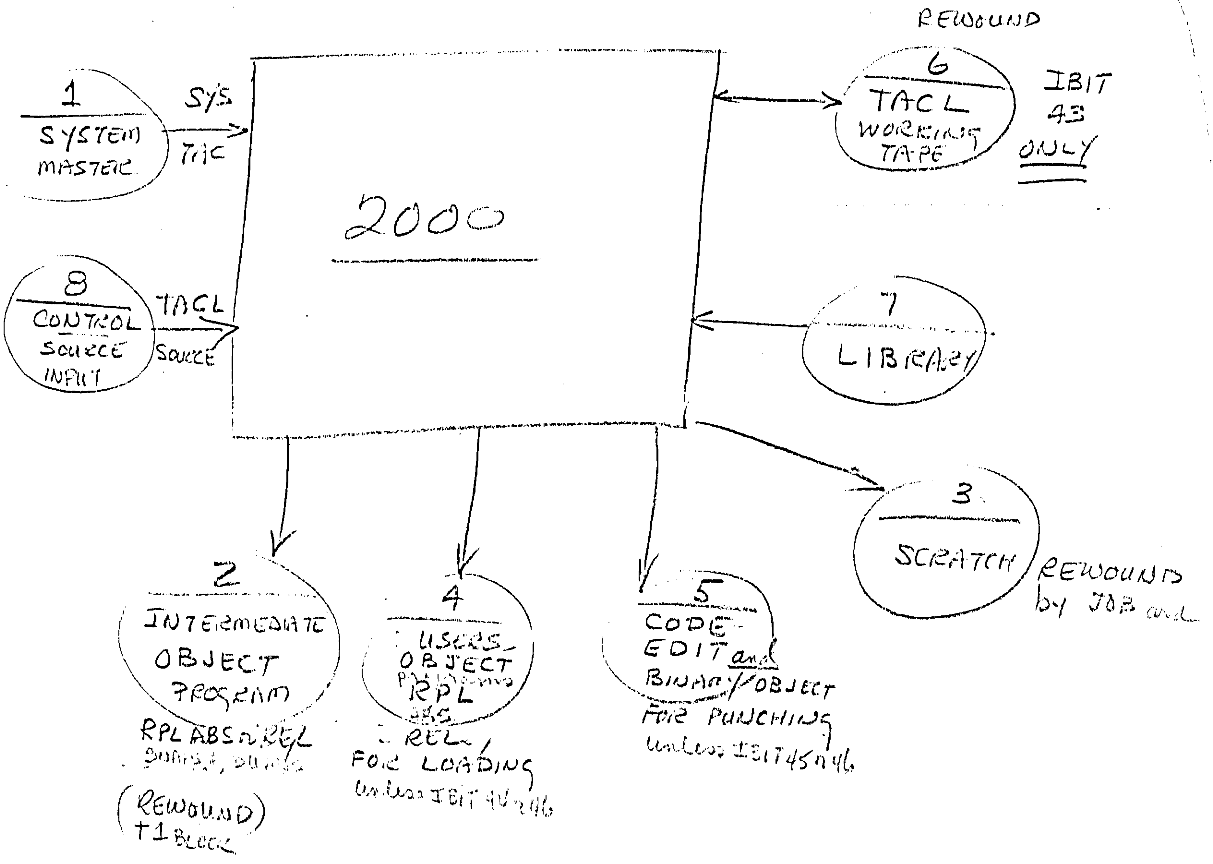
COMSIZE — (COMmon SIZE for REL programs): A single memory location which contains the size of Common in the left address portion. (Refer to COMMON, page 3-4.5; JOB, page 3-2.1; and REL, page 3-4.3.)

MASORG — (MASter ORiGin for REL programs): A single memory location which contains the origin of Common in the left address portion and zero or the origin of the master segment's final component in the right address portion. (Refer to MASTER, SEGMENT, and JOB, pages 3-5.2, 3-5.1, and 3-2.1.)



Altac ~~Fortran~~ / Fortran ~~Altac~~ Compilation with SYS Operator System

BJM



TAC II compilation with SYS

BJM

SECTION II

SYS TAPE ASSIGNMENTS AND FORMATS

Normal SYS use requires eight magnetic tape units; seven will suffice, however, when operating in paper tape or Console Typewriter modes of input. A description of the functions, contents, and data formats of these tapes is contained in this section of the manual.

Tapes 1 through 8 are described; other tapes have no significance to SYS, although their use may be directed by certain control instructions at the option of the user. These include such control instructions as REWIND, TAC, READF, READB, and WRITE, and such service routines as TACSERV, RPLC, and DATA.

TAPE 1 — SYS Program Tape

FUNCTION

The SYS Program Tape contains SYS and all associated programs and routines such as TAC, ALTAC, BINDEL, TACSERV, and RPLC. A detailed listing of the contents of this tape is contained in Appendix D.

FORMAT

The first several blocks of the tape consist of programs in a special SYS image form, while the remainder of the tape contains programs in RPL, ABS or REL form.

Image Portion

Each block of the SYS image portion is identified by its first word containing, in Bits 00-29, the Hollerith equivalent of the characters BLOCK. The remaining portion of the word is the block number in binary (scaled at T47). The remainder of each block is the machine language of the individual routines.

The first four blocks of the tape are numbered BLOCK000. The remaining blocks are numbered consecutively starting with BLOCK001.

Blocks 1 and 2 contain the most basic operations of SYS, and are read into memory (locations 0 through 377_8) by the operator at initialization time. The remaining SYS image blocks, called shuttle blocks, are one-block-length routines brought in normally one block at a time by SYS to the third block in memory (locations 400_8 through 577_8) to execute functions as required by control instructions and entries.

An exception to this rule is the relocatable loader (page 3-4.3) which occupies several blocks, four of which are read into locations above 1000_8.

A call for a SYS entry (refer to SYS Entries, page 4-1) is translated by SYS into calls for one or more blocks in succession. SYS then searches backward or forward for the proper blocks based upon their numbers. For this reason, blocks must appear on tape in their block number sequence.

RPL, ABS or REL Portion

Following the SYS image portion of the tape are several programs in RPL, ABS, or REL format. Some of these are used only by SYS, while others are service routines which may be called directly by the programmer.

ADDING PROGRAMS

Other programs, such as frequently-used production runs or special service routines, may be added to Tape 1 at installation option. The only restriction is that RPL programs added must not contain the characters BLOCK in the first 30 bits of the first word in any of the blocks placed on tape. The restriction does not apply to programs in ABS or REL card format because the first eight columns of each card contain an Identity and Sequence number in Hollerith code, although the remainder of the card is in Binary-Image Mode. The first five characters of the Identity and Sequence field is never equal to the characters BLOCK.

The last program on the tape is followed by a sentinel block of Z's which signals the end of useful information.

TAPE 2 — SYS Intermediate Tape

FUNCTION

The SYS Intermediate Tape is a scratch tape pre-empted by SYS for specific applications. The programmer should not use this tape as a program scratch tape unless he is thoroughly familiar with SYS operations and is willing to conform to the restrictions defined within this section.

FORMAT

The first block of Tape 2 is reserved for the exclusive use of SYS; it contains a copy of the last JOB control instruction executed (edited for the High-Speed Printer) and other internal SYS information. All references within this section concerning Tape 2 positioned means Tape 2 is rewound and then spaced forward one block. The remainder of the tape is used in conjunction with various control commands.

TAC

When a TAC command is executed by SYS, the following action takes place:

- Tape 2 is positioned.
- The machine language (RPL, ABS, or REL) program output is written on Tape 2 by TAC during compilation.
- When TAC returns control to SYS, two sentinel blocks of Z's are written and Tape 2 is positioned.
- If no major compilation errors were detected and/or if IBIT 47 (page 3-8.1.1) was set prior to the TAC control instruction, SYS will copy the compiled program onto the program tape, usually Tape 4, and will position Tape 2.

ALTAC

When an ALTAC command is executed by SYS, the following action takes place:

- Tape 2 is positioned.
- If IBIT 43 is set to one, the ALTAC language program will be copied onto Tape 2 for input to ALTAC. Tape 2 is then positioned.
- When the TAC compilation phase begins, Tape 2 is used in the same manner as described for TAC.

RPL, ABS and REL

If an RPL, ABS, or REL program is to be run from Tape 2, an RPL, ABS, or REL command may be given with 2,ID in the Address and Remarks field. SYS then performs the following action:

- Tape 2 is searched for the designated program.
- If the program is located, it will be read into memory and Tape 2 will be positioned.
- If it cannot be located, NOT HERE will be typed on the Console Typewriter and Tape 2 will be positioned.

DUMP and SNAP

Tape 2 is not affected at the time of interpretation of a DUMP or SNAP control instruction (pages 3-7.1 and 3-7.2, respectively). If DUMP control instructions are encountered, they will be interpreted and the parameters will be stored in a table within SYS. The resulting DUMP command words will then be processed only if a post-mortem dump (page 4-4) occurs.

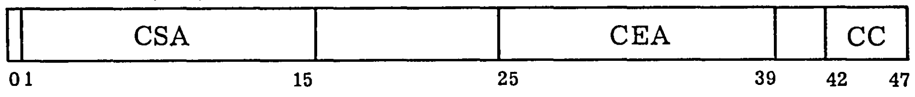
If SNAP control instructions are encountered, SYS will interpret them and plant snapshot intercepts at the designated locations in the object program. The specified areas are transcribed onto Tape 2 in Dump Data Format (page 2-3.2) when the intercepts are encountered at run time.

Snap dumps are always transcribed onto Tape 2 prior to termination of the object program, and hence are written prior to a post-mortem dump entry (1ERRDMP or 1SUBERR, pages 4-4 and 4-5). Binary information stored on Tape 2 is automatically converted for printing on the High-Speed Printer by the DUMPCON routine at the conclusion of the job. (Refer to Philco Program Description, PD-2, DUMPCON.)

Post-mortem dumps (specified by the DUMP instruction) are written onto Tape 2 after control is transferred to a SYS post-mortem dump entry (1ERRDMP or 1SUBERR, pages 4-4 and 4-5), or after initiation of the catastrophic dump process (page 3-7).

SNAP Format

Whenever a snapshot intercept is encountered, SYS writes the current console dump data (refer to the table on page 2-3.2 for format) onto Tape 2. The 13th word (Word 12 in the table) is the command word and contains the snap dump parameters — Core Starting Address (CSA), Core Ending Address (CEA) and Conversion Code (CC) -- illustrated as follows:



Immediately following the command word, SYS writes the specified dump area from memory onto Tape 2 in binary, using as many blocks as needed. As each snapshot intercept is encountered by SYS, current console dump data is written at the beginning of the next block. (Refer to the figure on page 2-3.3) Strings of identical words are deleted; only the first two equal words are written on tape, followed by a control word. Control must be transferred to a SYS post-mortem dump entry prior to processing the next JOB control instruction, or the snap dumps will not be edited and transcribed onto Tape 5.

Post-Mortem DUMP Format

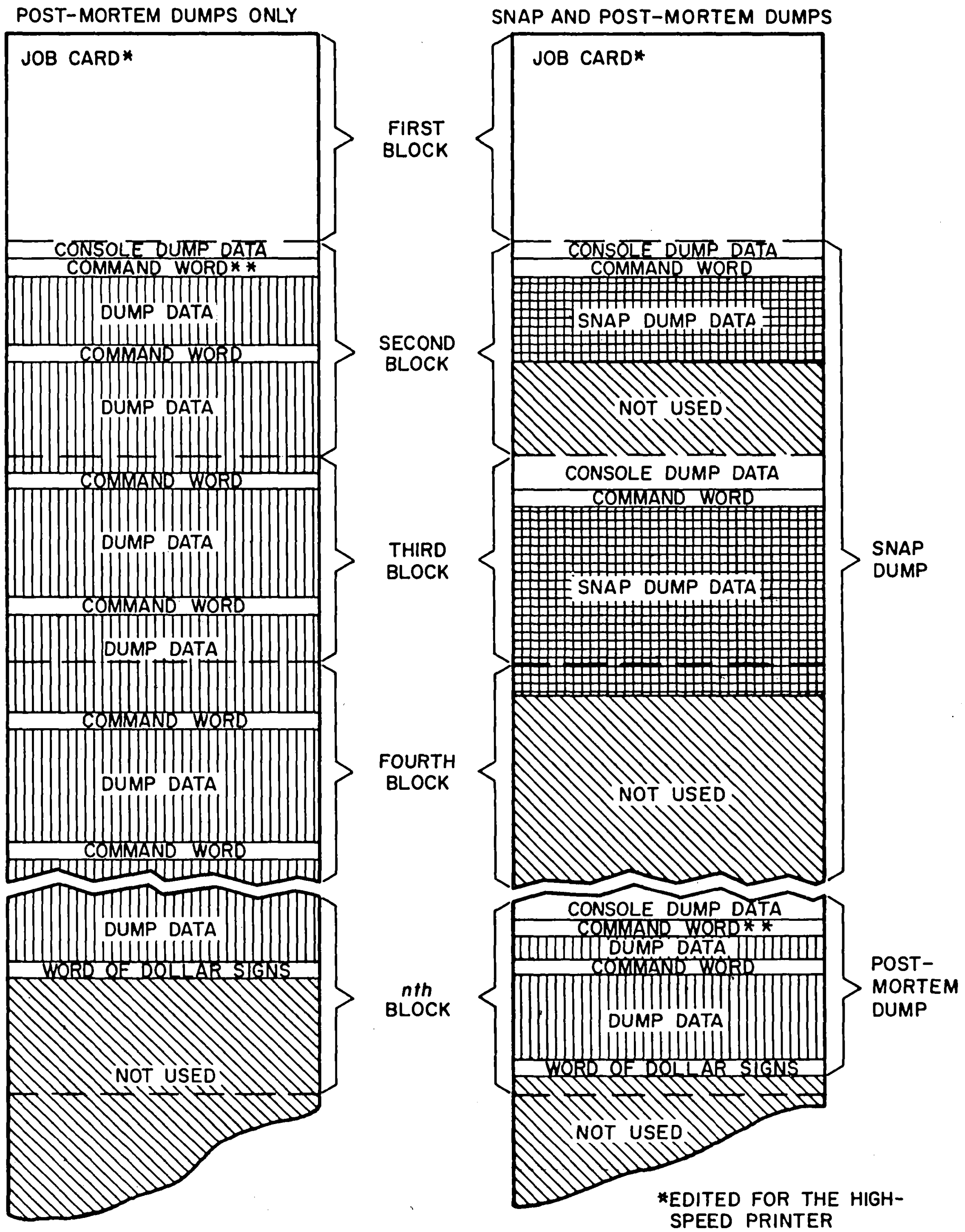
Whenever control is transferred to a post-mortem dump entry (1ERRDMP, 1SUBERR), or when the catastrophic dump is initiated, SYS writes the current console dump data at the beginning of the next available block. This will be the second block of Tape 2, if no snap dumps were specified.

DUMP DATA FORMAT

WORD	BITS	CONTENTS
0	1-16	(JA)
	24-47	24/1 means snapshot; non-24/1 means post-mortem
1	0-47	(A)
2	0-47	(Q)
3	0-47	(D)
4	0-47	(TOGGLES)
5	0-47	OVERFLOW: 48/0 means OFF; any other configuration means ON
6	0-47	ICO: 48/0 means OFF; any other configuration means ON
7	1-16	(0X)
	25-40	(1X)
8	1-16	(2X)
	25-40	(3X)
9	1-16	(4X)
	25-40	(5X)
10	1-16	(6X)
	25-40	(7X)
11	0-47	(IBIT)
12		COMMAND WORD
	1-15	Core starting address (of first word in dump area)
	18-23	Binary point in S-conversion (0-63)
	25-39	Core ending address (of last word in dump area)
	42-47	Conversion Code: A = Alphanumeric, C = Command, F = Floating-point, H = Hexadecimal, O = Octal, S = Fixed-point

The next input words are the contents of the dump area from Core Starting Address to Core Ending Address upon System entry. Strings of identical words are not copied (page 2-3.1). If necessary, these words extend beyond the second input block.

This format begins the second input block for either the snap dump or post-mortem dump. If snap dumps are not written, the format is used only once, i.e., prior to the first post-mortem dump.



*EDITED FOR THE HIGH-SPEED PRINTER

**THE FIRST POST-MORTEM DUMP COMMAND IS PRESET IN SYS'S TABLE AND PRESENTLY DUMPS LOCATIONS 60_8 THROUGH 75_8 IN ALPHANUMERIC.

Format of Tape 2 Following a Post-Mortem Dump

The 13th word (Word 12 in the table) of this block contains the first post-mortem dump command control word from SYS's table. The first dump command is preset in the table and dumps locations 60_8 through 75_8 (refer to 1CONLIN, page 4-19) in alphanumeric format.

The specified dump area is always written from memory onto Tape 2 immediately following the command word. Each succeeding command word from the table is written on Tape 2 immediately following the last word of the previous dump area, using as many blocks as necessary. (Refer to the figure on page 2-3.3.)

Upon conclusion, SYS rewinds Tape 2 and calls the DUMPCON routine. If the conversion code (Bits 42 through 47 of Word 12 in the table) is incorrect when processed by DUMPCON, a FORM *nn* error will result (see below).

FORM ERRORS

If an illegal conversion code is submitted to the DUMPCON routine, FORM *nn* is typed on the Console Typewriter, where *nn* indicates the illegal code in octal. Control is transferred to 1ENDJOB.

Either of the following conditions will cause this error to occur: (1) an illegal conversion code was submitted in an otherwise legal command word, or (2) a word of data was processed as a command word.

The latter would occur if DUMPCON did not encounter the expected format on Tape 2, as a result of one of the following errors:

- Tape 2 was moved by the object program.
- The FLEXO mode was used without a JOB control line or other initial positioning of Tape 2.
- Tape 2 was manually mispositioned during the clearing of a tape error.
- DUMPCON was manually read into memory without Tape 2 being rewound.
- Tape 2 was manually switched or repositioned after starting an object program run.
- A machine error occurred.

Because Tape 2 represents a portion of the required program interface between SYS and DUMPCON, any movement of Tape 2 other than that performed by SYS can result in a FORM *nn* error should a dump be required.

TAPE 3 — Compiler Scratch Tape

FUNCTION

Tape 3 is used as a scratch tape by the TAC and ALTAC compilers, and BINDEL and ANALYZER service routines (pages 3-3.1, 3-3.2, 5-4 and 5-3). It is rewound by SYS during execution of a JOB control instruction (page 3-2.1).

Except for the above functions, Tape 3 is available as a scratch tape for the programmer.

TAPE 4 — User's Program Tape

FUNCTION

The User's Program Tape* has the function of collecting binary programs resulting from the compilation process. This collection is cumulative although programs placed on the tape can be removed by the service routine BINDEL (page 5-4).

FORMAT

SYS places programs on Tape 4 in RPL, ABS, or REL format. The three types of programs may appear intermixed on the tape, in any order, with each program starting at the beginning of a block.

The sequence of the programs on Tape 4 is that in which they were compiled. The last program is followed by two sentinel blocks of Z's, which signify the end of useful information on the tape. The tape is rewound during the initialization of SYS to insure that programs to be compiled will be transferred to the useable portion of the tape.

Systems personnel at each installation have an option that will prevent the accumulation of programs on the User's Program Tape (Tape 4 only), from job to job. If the option is selected, the following actions will be performed at the beginning of every job:

- Tape 4 will be rewound.
- A sentinel block of Z's will be written on the tape.
- Tape 4 will be backspaced one block.

* Certain tapes other than 4 may be used as the user's program tape. (Refer to PROGTAPE, page 3-8.8.)

TAPE 5 — SYS Output Tape

FUNCTION

The SYS Output Tape is that on which most information is placed for off-line printing and punching. To use the tape, however, the programmer must adhere to the SYS standards below concerning the four modes of Data Select.

SYS writes information on this tape to identify jobs and provide accounting and dump information. In addition, the compilers produce their printed output via this tape. Following compilation, SYS transfers binary card images to this tape for off-line punching (refer to TAC and ALTAC control instructions, pages 3-3.1 and 3-3.2, respectively).

FORMAT

The format of the various types of SYS output, specified by the particular Data Select mode, is described as follows:

Standard Print

Data Select Zero is used to identify blocks to be printed on the High-Speed Printer. The standard print plugboard setting is one-to-one.

Binary Punch

Data Select 1 is used to identify blocks for binary card punching in Image Mode (20 words per card, 6 cards per block), with the Control Character SENSE/IGNORE Switch of the Punched Card Controller set to IGNORE when these cards are punched. To aid operators in separating binary decks and associating them with other output from the various jobs, SYS precedes each distinct deck with one card in which all rows of Columns 1 through 8 are punched. For the convenience of operators, any programs using the off-line punch features of Tape 5 should state via the Console Typewriter the number and type of cards to be punched.

Hollerith Punch

Data Select 2 is used to identify blocks for Hollerith card punching in Code Mode (10 words per card, 12 cards per block), with the Control Character SENSE/IGNORE Switch of the Punched Card Controller set to IGNORE when these cards are punched. As mentioned previously, any program using the off-line punch features of Tape 5 should state, via the Console Typewriter, the number and type of cards to be punched.

Accounting Cards

Data Select 3 is used to identify accounting cards placed by SYS on Tape 5 for card punching in Code Mode (10 words per card, 12 cards per block). The Control Character SENSE/IGNORE Switch of the Punched Card Controller should be set to SENSE when these cards are punched.

X-Y Plotter

Data Select 4 is used to identify blocks containing data for the X-Y Plotter.

WRAPUP FUNCTION

Whenever the Wrapup function of SYS is used by an operator during execution of a JOB control instruction (page 3-2.1), SYS writes the following onto Tape 5:

- An Accounting Card for the last executed job (if there is an Accounting Clock).
- An end-of-job block which causes a final sheet in the High-Speed Printer to be ejected when Tape 5 is printed.
- End-of-tape sentinels with unconditional stops in each of the following Data Select modes used by SYS:
  1. In Data Select 0, this statement, edited for the High-Speed Printer, appears several times:

     THIS IS THE END OF THE TAPE STOP PRINTING

     It is followed by a page-eject block which ejects a sheet from the High-Speed Printer.
  2. In Data Select 1, a block of filler characters (Octal 32) is edited for punching.
  3. In Data Select 2, ten words are edited to punch a Hollerith-Image card with the first and last rows of all columns punched.
  4. In Data Select 3, the same card as that produced by Data Select 2 is punched.
  5. In Data Select 4, the following operations for the X-Y Plotter are performed:
     a. The pen is raised and positioned 0.5 inches toward the top of the paper, and is then positioned at the left margin.
     b. The pen is lowered and a saw-toothed pattern with a 45° diagonal length of 1.12 inches is drawn from left to right.
     c. The pen is raised and positioned 1.26 inches above the saw-toothed pattern.
- A sentinel block of Z's is then written, and the tape is rewound with lockout.

TAPE 6 — Scratch Tape

FUNCTION

Tape 6 is used as a scratch tape by the BINDEL and ANALYZER service routines (pages 5-4 and 5-3), and the ALTAC compiler (page 3-3.2). Tape 6 may also be a user's program scratch tape.

In addition, the TAC compiler will use Tape 6, if IBIT 43 (refer to page 3-8.1.1) is set to one. The input program is transferred prior to compilation from Tape 8 to Tape 6 in the same mode.

Tape 6 is rewound during execution of each JOB control instruction (page 3-2.1).

TAPE 7 — TAC Library Tape

FUNCTION

The TAC Library Tape is normally Tape 7. Subroutines should be in relocatable binary format because the tape will automatically be searched by SYS if a REL . . . SUBS call is encountered (page 3-4.3).

In general, Tape 7 contains the various macro-instructions, generators, and relocatable binary subroutines which are available to the programmer.

Binary library tapes must contain a card of A's preceding the first binary subroutine and a card of Z's following the last subroutine. (Refer to the Program for Library Updating and Maintenance (PLUM) Philco Program Report No. 13.)

If no library is needed, Tape 7 may be used as a scratch tape.

TAPE 8 — SYS Input Tape

FUNCTION

The SYS Input Tape in MAGTAPE mode (refer to CONIN, page 3-2.2) contains the user's source language program and SYS control instructions in either Code Mode (10 words per card, 12 cards per block) or Image Mode (20 words per card, 6 cards per block).

SYS handles changes in format between Code and Image Modes, provided the proper CONIN line appears just prior to the change. The only restriction is that a CONIN IMAGE control instruction in Code Mode format must be an even card position in the block (such as second, fourth, and sixth).

Punched card information may be transcribed on magnetic tape in either of two modes. If transcribed in Code Mode (page 1-6), the information on cards in Hollerith code is automatically converted by the Punched Card Controller to Philco Code. If transcribed in Image Mode (page 1-6), no conversion takes place; instead, an image of the card punches, 12 bits per column, is transcribed on tape. Cards punched with binary (non-Hollerith) information must be transcribed on magnetic tape in Image Mode. If SYS control cards punched in Hollerith code are transcribed on magnetic tape in Image Mode, SYS will automatically convert them to Philco Code prior to their execution. (Refer to Philco 2000 Character Codes, Appendix B.)

SYSC CONTROL LINES

9 L	10 16 LOC	17 24 COMMAND	25 ADDRESS —PARAMETERS	RE
		JØB	NAME and COMMENTS (8char)	
		CØNIN	MEDIUM MAGTAPE } code, tape 8 MAGTAPE, CODE } MAGTAPE, IMAGE – BINARY, TAPE 8 PAPER } on line FLEXO }	
	RPL REL ABS	TAC ALTAC	SOURCE MEDIUM, UNIT#*, LIB*, UNIT*, UNIT*, NOSUBS* MAGTAPE / FLEXO / PAPER *IF OTHER THAN 8 *IF USED (MUST FOR MACROS & GENS FOR REL) *IF LIB OTHER THAN 7 * FOR ABS → SUBS when NOSUBS in BIN ARE ATTACHED AT COMPILE TIME OR REL when subs are attached at run time missing assumes ... assuming NO SUBS if missing	[illegible]
		DUMP	MODE, FROM(8), TO(8) Ø octal C command A alphanumeric F floating S fixed	
		RPL ABS	UNIT#, ID, GØ*, LOADING OFFSET(8)*	
		REL	UNIT#, ID*, GO*, LIST*, SUBS*, UNIT*, UNIT* ... UNIT#: all except 5T ID: * or 8 if program when follows more parameters GO: NØGO LIST: list the undefined symbols on 5T for printer SUBS: search tape 1 for REL subs UNIT, UNIT: searches for REL bin subs here before searching 1T	
		ORIGIN	LOC(8) = PMAX + desired starting location | defines upper limits of program for back to front REL loading.	
		COMMØN	SIZE(10), LOC(8) where LOC < ORIGIN LOC defines size of area for front to back loading	
	LOC(8)	ØCT	16 OCTAL DIGITS of CORRECTED FULL WORD OF DATA	15
[illegible]	LOC(8)	CØMMAND	QUATCODE, OCTAL V FIELD, XR NUMBER to correct a half word instruction	
L R	AT(8)	SNAP	CONV; FROM; TO; CALLSTORE(8); CONDITION Ø C A F S H; E error return; N next control line return FROM: LOC(8) or n, X TO: LOC(8) or n, X CALLSTORE: leave blank or specify some unusual free 6 word table left in [illegible] storage CONDITION: see table Pg A-28	A- [illegible] 1/2
		JMP JMPL JMPR	LOC(8) or *	BJ

IBIT	BIT, BIT, BIT - - - - - (0 → 47) 43 : TAC II source to be copied from 8T to 6T 44 : Do <u>not</u> copy binary running program from 2 to 4 45 : Do <u>not</u> copy binary punch program from 2 to 5 46 : Do <u>not</u> copy object program from 2 to 4 if <u>errors</u> 47 : Copy object program from 2 to 4 regardless of errors	
HLT REM	Remarks (STOP) (NO STOP)	
REWIND REWINDLO	UNIT nos, - - - .	
READF READB WRITE	} UNIT #, NBS, NBP, CSA(8)	
WRTSENT LØCSENT	UNIT #, 8 character sentinel word	
RPL TAPE ↕ END△DATA	1, DATA, GO COPY TO TAPE # (0, 3, 6, 7, 9 → 15 only) input data cards to be copied @ 12 x 10	
JØB RPL ↕ JOB CONIN	CHECKS 1, CHECKS, GO ENDINPUT FLEXO	} Used to check out single station card to tape conversions
JOBSRCH	JOBCARD ID, IMAGE 8 character <u>SYSC</u> reads tape 8 in image mode	
RPL	1, DELRPL, GØ ID ID ⋮	for RPL tape updating maintenance
END	or ENDALL ?	
RPLC	[illegible] in SYSRPLC from tape 1	
NEWTAPE	UNIT # (0, 3, 6, 7, 9 → [illegible] only	
COPY	INTAPE #	
COPYTIL SKIPTIL DELETE ADD CORRECT	} INTAPE #, ID	
IDCHANGE new ID words END	INTAPE #, ID old) → 16 char as many ID change as needed	
ENDALL	LIST # to list all I cards on flexo	

L	LOC	COMMAND	
		AIDE	call in SYSAIDE from Tape 1
		COPY	FROM #, TO #; NBC $
		COMPF COMPB	From #, TO #/ NBC, error listing form $ F flexo T on tape #5
		COPYCOMP	FROM #, TO #, NBCC; error form $
		REWIND	UNIT, UNIT, . . . UNIT $
		REWINDLØ	UNIT, UNIT, UNIT $
		ENDALL	$
		TACSERVS	call in SYS TACSERVS from tape 1
		REWIND	UNIT
		LOCATE	UNIT #; ID
		COPY	FROM #; TO # mandatory specifies old + new tapes to initialize TACSERV
		LOCFLAD	locate by [illegible]
		LOCSEQ	locate by card sequence nos.
		ADD REPLACE DELETE	Card ID & SEQ ; No of. [illegible] & col
		SEQ	
		NOSEQ	
		ENDPROG	
		ENDALL	
		JØB	ENDINPUT ends SYSC control & wraps up operations

B

SECTION III

CONTROL LINE FUNCTIONS

The fundamental operations of SYS, such as loading programs into memory, reading and writing on tape, and starting and stopping runs, are called into action by SYS Control Lines — each consisting of a control instruction and its parameters.

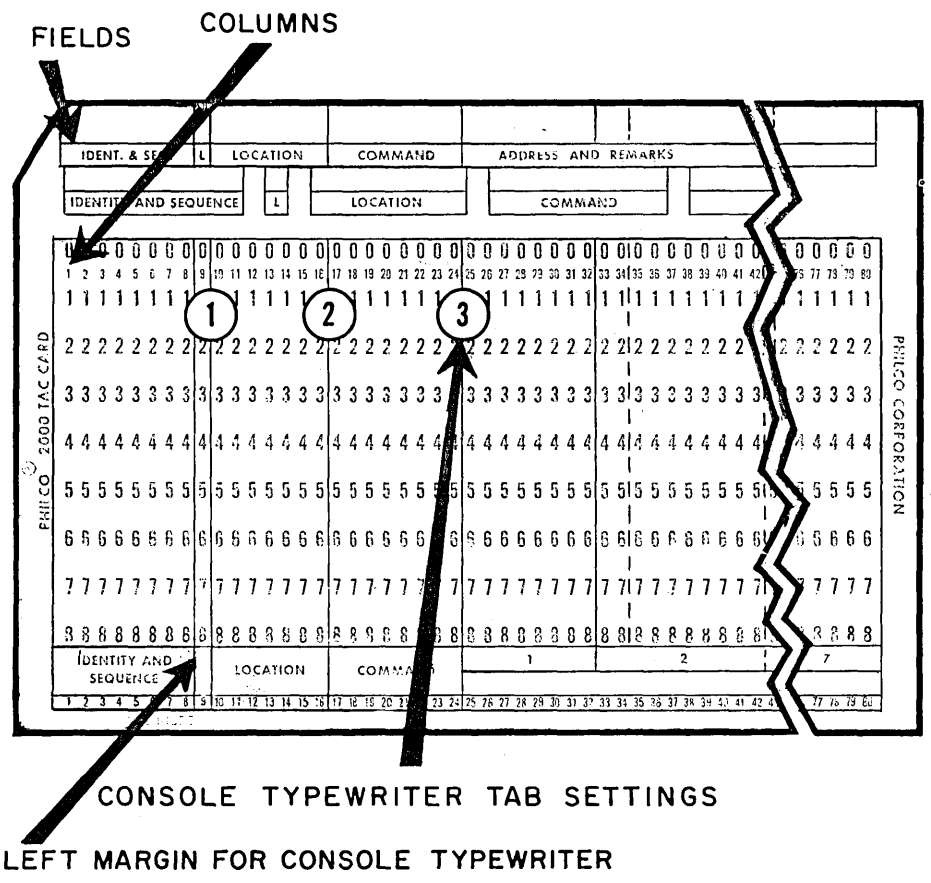
SOURCES OF CONTROL LINES

A Control Line must be written in a format similar to that for TAC instructions and may be entered into the computer from one of the three following sources:

- Cards via Magnetic Tape 8 (SYS input tape)
- Paper tape (seven-channel; on-line via the word-at-a-time channel).
- Console Typewriter (manually).

CONTROL LINE FORMAT

The format for SYS Control Lines is illustrated below, indicating the location of columns, fields, and Console Typewriter tab settings, including the left margin for the typewriter. Paper Tape format is the same as that for the Console Typewriter.



PHILCO
A SUBSIDIARY OF Ford Motor Company,
COMPUTER DIVISION

PHILCO CODING FORM

Page......of..........

Program:			Programmer:	Date:
IDENTITY AND SEQUENCE	L	LOCATION	COMMAND	ADDRESS AND REMARKS
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16	17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80
1 2 3 4 5 6 7 8	9	10 11 12 13 14 15 16	17 18 19 20 21 22 23 24	25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

TF-25

Information should be written within each field as follows:

- Label Field (Column 9). An L or an R normally placed in this field indicates whether the left or right half of an instruction or an address is to be acted upon by a control instruction. For the Console Typewriter, the single character is entered before the first tab setting following a carriage return.

- Location Field (Columns 10-16). Normally the octal address of an instruction or data to be acted upon by a control instruction is placed in this field. For the Console Typewriter, information for this field is to be entered between the first and second tabs following a carriage return.

- Command Field (Columns 17-24). The control line command is to be placed in this field. For the Console Typewriter, the field is between the second and third tabs following the carriage return.

- Address and Remarks Field (Columns 25-80). The parameters are to be written in this field in the following manner:

  1. Parameters must be separated by break characters, i.e., a comma (,), semicolon (;), period (.), or slash (/). No distinction is made between these characters, except where otherwise noted.

  2. The last parameter must be followed by a dollar sign if remarks are to follow.

  3. All parameters for a single control line <u>must</u> be contained on one SYS control card. Multiple card control lines are not permitted.

  For the Console Typewriter, information for the Address and Remarks field is to be entered following the third tab.

REMARKS

1. If any control line command is illegal or, in general, if any of its parameters are illegal, the error type-out CONT. LINE ERR.........(page 4-18) will be typed on the Console Typewriter, unless otherwise noted.

2. Upon completion of successful action of control instructions, control is generally transferred to the SYS entry 1NXTCON (page 4-7), which requests and attempts to execute the next control instruction.

3. The Philco Coding Form, illustrated on page 3-1.1, is provided for the user's convenience in writing his programs. Philco 2000 TAC cards, illustrated on page 3-1, are available for punching the information from the Coding Form.

WRITING CONTROL LINES

As indicated, the command word of a SYS control line is to appear in the Command field of a card, and the parameters are to appear in the Address and Remarks field. To simplify explanation of these lines, all capitalized words which appear within the illustration of the instruction format below are to appear on the card exactly as shown. All italicized words are to be replaced — generally by an L or R in the Label column, an address in the Location field, and parameters selected by the programmer in the Address and Remarks Field.

For example, the instruction LOCTACL (page 3-6.9) is described as:

L	Location	Command	Address and Remarks
		LOCTACL	*t,id*

The written coding for searching Tape 4 for the TAC language program PROGXRAY should appear as:

L	Location	Command	Address and Remarks
		LOCTACL	4,PROGXRAY

Although commas are used in format examples throughout the manual, any of the acceptable break characters (, ; . /) are permissible except where otherwise noted.

CONTROL LINE FUNCTIONS

SYS control lines are grouped according to function into seven classifications. These functions, their meaning, and an example of a particular control instruction within each function are listed as follows:

FUNCTION	MEANING	EXAMPLE
Initialization	Instructions used to prepare the System for compilations or runs.	JOB
Compilation	Instructions used to call in the compilers and initiate source program conversion from mnemonic coding into machine language.	TAC
Loading	Instructions used to load an object program into memory and initiate the running of the program.	REL

FUNCTION	MEANING	EXAMPLE
Segmentation	Instructions used to divide a program into smaller segments to permit loading and running of only parts of the program in memory at a given time.	SEGMENT
Tape Handling	Instructions used to read from, write onto, or rewind magnetic tapes.	READF
Debugging	Instructions used to aid the programmer in diagnosing and patching programs which may be faulty.	DUMP
Special	Instructions used to perform miscellaneous functions on the computer, such as setting a pseudo toggle register, clearing memory locations, halting an operation, or transferring control to a memory location.	IBIT

INITIALIZATION FUNCTIONS

The first set of instructions to be described concerns initialization of SYS. The JOB control instruction prepares the System for a new program, and the CONIN instruction specifies the source of control instructions.

JOB

FUNCTION

Prepares the System for a new job or series of operations and identifies the job. If an Accounting Clock* is present, the JOB instruction will also terminate accounting functions of the previous job. JOB is the highest level instruction in the SYS hierarchy because each job must begin with a JOB control instruction.

FORMAT

L	Location	Command	Address and Remarks
		JOB	*id*

PARAMETER

id 16 or fewer alphanumeric characters** which identify the job to be processed.***

ACTION

Execution of the JOB control instruction initiates the following action:

- An END JOB page for printing is edited and written on Tape 5, if an Accounting Clock is present.
- Information for an Accounting Card for the previous job is edited and written on Tape 5 with a Data Select of 3.
- The wrapup toggle is checked (refer to WRAPUP FUNCTION, Tape 5, page 2-6.1). If it is on, WRAPUP? will be typed on the Console Typewriter and the computer will halt. The operator may either (1) press the Advance Bar if a wrapup is desired (which will cause the System to wrap up Tape 5), or (2) turn the toggle off and then press the Advance Bar if no wrapup is desired.
- The JOB card is typed on the Console Typewriter.

*An Accounting Clock may or may not be present at a computer installation. If the clock is not present, any reference to Accounting Clock or Accounting Cards in this manual should be ignored.

**Only those characters which are upper case characters of the Console Typewriter should be used. (Refer to Philco 2000 Character Codes, Appendix B.)

***These characters are checked by JOBSRCH, a control instruction which causes a search for a job (page 3-8.2) and by Accounting Card routines. Spaces are ignored.

- The skip toggle (Toggle 24 at most installations) is checked. If it is on, the computer will halt. The operator may either (1) press the Advance Bar if the current job is to be skipped (which will cause SYS to search for the next job) or (2) turn the toggle off and then press the Advance Bar if the current job is to be processed.
- The date and time from the Accounting Clock are typed on the Console Typewriter (as described within the CLOCK control instruction, page 3-8.4), and are stored in 1DATE and 1DATE+1 (215_8 and 216).
- Information from the JOB card is transferred to Tape 2 and the tape is positioned immediately following the first block.
- The edited coding for the JOB page is written on Tape 5 for printing.
- All index registers except 1 and 2 are cleared.
- 1NTRYJA is initialized by placing zeroes in the Jump Address Register (page 1-6).
- The DUMP control instruction table and the IBIT control word are cleared to zeroes (pages 3-7.1 and 3-8.1, respectively).
- Tapes 3 and 6 are rewound.
- The left address portions of COMORG, COMSIZE, and MASORG are reset to 1000_8 and their right address portions are reset to zero (page 1-6).
- 1ERRDMP, 1ENDJOB, and 1NXTCON are reset (pages 4-4, 4-6 and 4-7).
- Tape 4 is reassigned as the user's program tape (refer to PROGTAPE, page 3-8.8).
- Memory is cleared from location 1000_8 to the end of memory.
- If a Systems personnel option, that prevents the accumulation of programs on Tape 4 from job to job, has been placed into effect, Tape 4 will be rewound, a sentinel block of Z's will be written on the tape, and the tape will be backspaced one block.

REMARKS

In addition to the placement of the *id* within the first 16 or fewer characters of the Address and Remarks field, the remaining 39 characters of the field may be used to identify the name of the programmer and/or provide any other descriptive information. The break character separating the *id* and other information must be either a period or a comma.

EXAMPLE

L	Location	Command	Address and Remarks
		JOB	PROGRAM XYZ, JOHN SMITH

Explanation: Execution of this instruction causes SYS to be prepared for a new job, identified as PROGRAM XYZ written by John Smith.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	JOB *id*	The job specified by *id* is ready to be executed.
2	WRAPUP?	A JOB card is read with toggle switch *n* set, where *n* is the switch used by the installation for wrap-up (usually 25).
3	*date, time*	The date and time of day from the Accounting Clock have been typed in the form: MM-DD HH-MM.T, where MM-DD is the month and day, and HH-MM.T is the hour (per 24-hour day), the minute and tenth of a minute.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	CLOCK FAILURE	The Accounting Clock was interrogated but was unavailable.

CONIN — Control Instruction Source

FUNCTION

Specifies a new source of control instructions or a new mode of operation.

FORMAT

L	Location	Command	Address and Remarks
		CONIN	*source*

PARAMETER

source One of the following sources must be selected:

MAGTAPE or MAGTAPE, CODE or CODE	Indicates that the next instruction is to be in Code Mode format on Magnetic Tape 8.
MAGTAPE, IMAGE or IMAGE	Indicates that the next instruction is to be in Image Mode on Magnetic Tape 8.
PAPER	Indicates that the next instruction is to be on paper tape.
FLEXO	Indicates that the next instruction is to be typed on the Console Typewriter.

ACTION

Whenever CONIN is executed, SYS expects to receive its next control instruction from the mode of input specified by the CONIN parameter.

REMARKS

1. Paper tape must be used on-line through the word-at-a-time channel.

2. Whenever control instructions are entered from the Console Typewriter, a carriage return character must end each instruction line. If a typing error should occur, the entire instruction may be reentered by pressing the STOP CODE key. In response to this action, SYS issues a carriage return and requests a new type-in via the white light signal.

3. A transfer of control to 1SYSIN (page 4-3) initiates SYS to the FLEXO mode of input. This may be modified at the option of an installation to any mode of input.

4. Refer to Tape 8 (page 2-9) for a description of the two input modes of information.

EXAMPLE

L	Location	Command	Address and Remarks
		CONIN	MAGTAPE

Explanation: Execution of this instruction prepares SYS to accept the next instruction from Tape 8 in Code Mode format.

COMPILATION FUNCTIONS

The second group of control instructions specifies the type of compilation to be made using SYS as an executive routine. Included are Philco's basic mnemonic language, TAC, and Philco's algebraic language, ALTAC. Both TAC and ALTAC contain parameters which indicate the type of program to be compiled (RPL, ABS, or REL) and the source of the TAC language program (magnetic tape, paper tape, or the Console Typewriter).

TAC — Translator-Assembler-Compiler

FUNCTION

Initiates a TAC compilation.

FORMAT

L	Location	Command	Address and Remarks
	format	TAC	*source,lib,subs*

PARAMETERS

format Format in which the program is to be compiled. The parameter is to be placed in the Location field, and if omitted, REL is assumed. One of the following three may be selected:

RPL Indicates that the program is to be compiled in RPL format.

ABS Indicates that the program is to be compiled in ABS format.

REL Indicates that the program is to be compiled in REL format.

source Source of the TAC language program. It is to be placed as the first parameter in the Address and Remarks field and must be selected from one of the following:

MAGTAPE Indicates that the TAC language input will be on magnetic tape (Tape 8 assumed).

MAGTAPE,*t* Indicates that the TAC language input will be on a magnetic tape other than 8, where *t* is a decimal number — 0,6,7, or 9 to 15, — representing the specified tape.

PAPER Indicates that the TAC language input program will be on paper tape.

FLEXO Indicates that the TAC language program will be entered from the Console Typewriter.

Note: An asterisk (*) may be specified as *source* to denote MAGTAPE,LIB or MAGTAPE,8,LIB as the parameters in the Address and Remarks field. No additional parameters may follow the asterisk.

lib (OPTIONAL). A parameter which may be specified if library routines are to be used by the TAC program. It is to be placed in the Address and Remarks field and may be selected from one of the following:

LIB, $t_1,\ldots,t_n$ Indicates that library routines are to be used from one or more tapes, where t_1 to t_n represent the specific tape units on which these tapes will be mounted. If LIB is written as the parameter with no tape designation specified, Tape 7 is assumed to be the library tape. If one or more tape numbers other than 7 are written following LIB, Tape 7 will not be used unless it is listed with the others.

subs (OPTIONAL). A parameter which may be specified if subroutines from the TAC library are to be compiled with the program. It is to be placed in the Address and Remarks field and must be one of the following:

SUBS	Indicates that subroutines are to be compiled.
NOSUBS	Indicates that no subroutines are to be compiled.

Note: If the parameter is omitted, RPL and ABS compilations assume SUBS; REL compilations assume NOSUBS. Generators and macro-instructions are always compiled with the program.

ACTION

Execution of the TAC control instruction initiates the following action:

- The original TAC language program on Tape 8 is copied onto Tape 6 prior to compilation if IBIT 43 is set. It remains there following compilation.
- The Accounting Clock is interrogated and the date and time are edited and stored in 1DATE and 1DATE+1 (215_8 and 216_8), as described within the CLOCK control instruction (page 3-8.4).
- The TAC language program is compiled and assembled with or without subroutines into a machine language program on Tape 2. The Code-Edit is prepared and is placed on Tape 5 for off-line printing on the High-Speed Printer using Data Select 0.
- The compiled binary program on Tape 2 is copied onto the program tape and/or Tape 5 if no IBITS were set and no serious compilation errors were detected. IBITS 44-47 are used to qualify this decision in the manner indicated in the diagram on page 3-3.1.2. Taken independently, these IBITS cause the following action:

  IBIT 47 forces transfer of programs to the program tape and/or Tape 5.

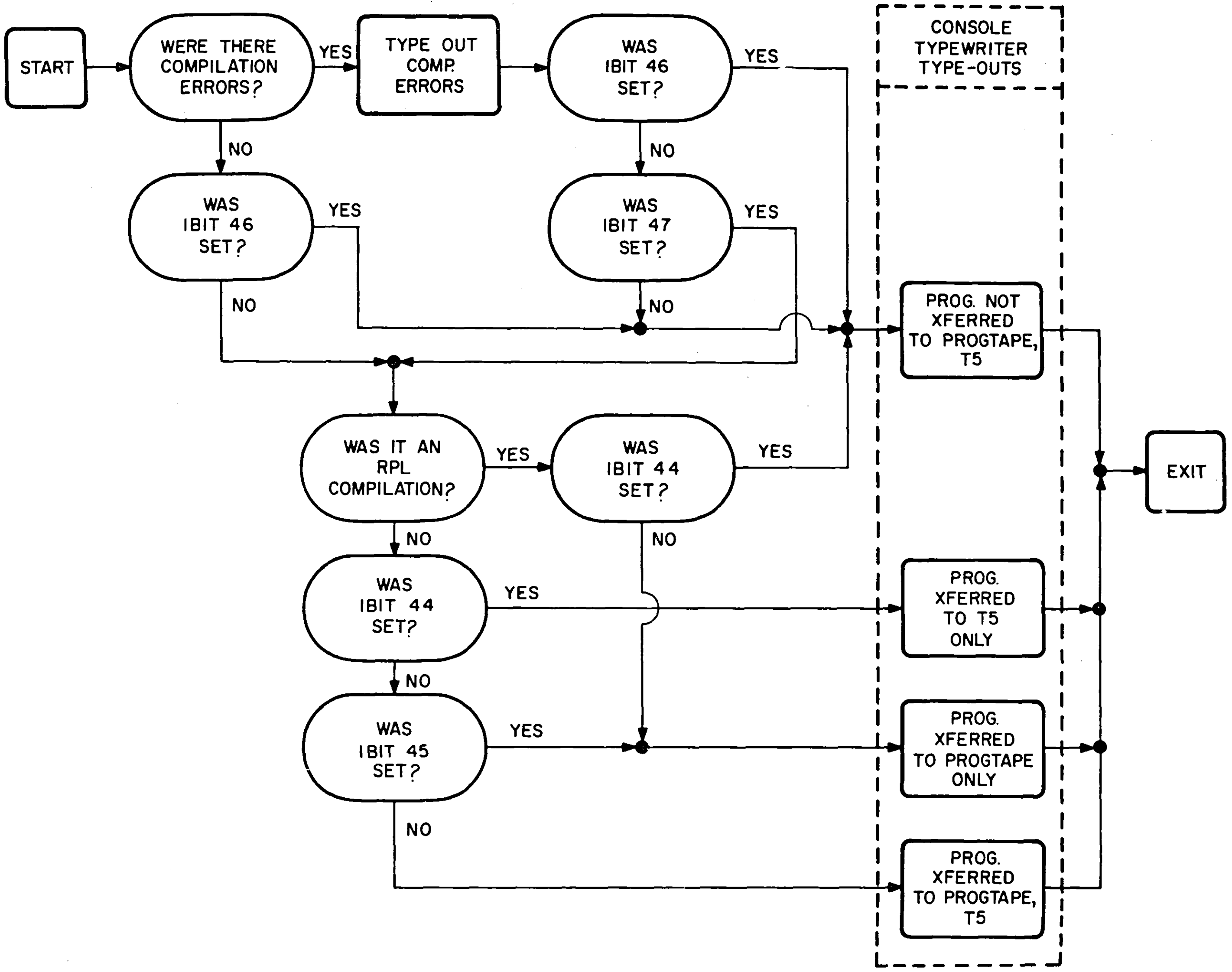
  IBIT 46 suppresses transfer of programs to the program tape and/or Tape 5.

  IBIT 45 suppresses transfer of ABS and REL card programs to Tape 5.

  IBIT 44 suppresses transfer of programs to the program tape.

REMARKS

1. If IBIT 37 is set, Philco 212 mnemonic commands will be considered legal and will be compiled into their proper bit configuration.
2. Setting IBIT 38 causes the error comment POSSIBLE F-BIT ERROR to be produced on the Code-Edit whenever a possible F-bit error is detected during a TAC compilation. (Refer to page 3-8.1.1.)



Logic Diagram for Conditional Post-Compilation Transfers

3. If IBIT 39 is set, TAC will not store the intermediate code in the higher-order portion of memory on 32,768-word memory machines. Instead, the code will be written onto Tape 3 and the memory space thereby released will be used for the symbol table. (Refer to page 3-8.1.1.)
4. Setting IBIT 41 prior to the execution of the TAC command specifies that input is to be in Binary-Image Mode on a tape other than 8.
5. If IBIT 42 is set, input will be in Hollerith-Image Mode on a tape other than 8 (refer to IBIT, page 3-8.1).

IBIT 0 is restricted to system maintenance use

6. If a program to be compiled is on Tape 8, it must immediately follow the TAC control instruction. In addition, if IBIT 43 is set, the first card following the TAC command must be an I card. The I card must have an I in Column 9 and the Location field must be blank.

EXAMPLE ONE

This example illustrates an RPL TAC call using the standard library tape.

L	Location	Command	Address and Remarks
	RPL	TAC	MAGTAPE,6,LIB

Explanation: Execution of this instruction causes the TAC language program on Tape 6 to be compiled into RPL format. Library routines are located on Tape 7. SUBS is assumed.

EXAMPLE TWO

This example illustrates a REL TAC call using more than one library tape.

L	Location	Command	Address and Remarks
		TAC	MAGTAPE,LIB,9,10,11,SUBS

Explanation: Execution of this instruction causes the TAC language program on Tape 8 to be compiled into REL format. Library subroutines are located on Tapes 9, 10, and 11 only. Subroutines are to be compiled with the program.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	PROG. XFERRED TO PROGTAPE, T5	The compiled program was transferred to the program tape and the information for binary cards was transferred to Tape 5.
2	PROG. NOT XFERRED TO PROGTAPE, T5	The compiled program was not transferred to the program tape and information for binary cards was not transferred to Tape 5 because of compilation errors or IBIT settings.

CONTINUED

CONSOLE TYPEWRITER NORMAL TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
3	PROG. XFERRED TO PROGTAPE ONLY	The compiled program was transferred to the program tape but information for binary cards was not transferred to Tape 5.
4	PROG. XFERRED TO T5 ONLY	Information for binary cards of a compiled program was transferred to Tape 5 but the program was not transferred to the program tape.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	ID MISSING	The first card of a TAC language program transferred from Tape 8 to Tape 6 was not an I card. Control is transferred to 1ENDJOB (page 4-6).
2	COMP. ERRORS	One or more serious compilation errors were detected by TAC.
3	TAPE *t* FAULTY	A non-recoverable error was encountered on Tape *t*. Further action is dependent upon the operator.
4	TAPE *t* IN LOCAL	Tape *t* is in local operation. The operator should place Tape *t* in automatic mode and press the Advance Bar to continue.
5	TAPE *t* ROCKED 5	A parity or sprocket error persisted during five successive retries. The operator may press the Advance Bar to continue retrying.
6	TAPE *t* WR RING	There is no Write Ring on Tape *t*. To continue, the operator may insert a Write Ring and press the Advance Bar.

ALTAC — Algebraic Translator into TAC

FUNCTION

Initiates an ALTAC compilation.

FORMAT

L	Location	Command	Address and Remarks
	format	ALTAC	*source,lib,subs*

REMARKS

The parameters, action, and type-outs are the same as those for TAC control instruction (page 3-3.1), except for the following:

1. ALTAC statements from the input tape are converted into TAC language and placed on Tape 6. Following compilation of the TAC language into machine language, the program is placed on Tape 2. Transfers from Tape 2 to the program tape and/or Tape 5 take place the same as for TAC compilations.
2. If IBIT 43 is set to one prior to an ALTAC compilation, the ALTAC statements will be copied from the input tape to Tape 2 and the program will then be compiled from Tape 2.
3. Tape 6 may not be used as a source input tape. If it is specified as such, ILLEGAL INPUT TAPE NO. will be typed on the Console Typewriter and control will be transferred to 1ERRDMP.

EXAMPLE

L	Location	Command	Address and Remarks
		ALTAC	MAGTAPE,LIB,9,10

Explanation: Execution of this instruction causes the ALTAC language program on Tape 8 to be compiled into REL format. Library routines are located on Tapes 9 and 10 only.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	EALTAC	The ALTAC compiler detected one or more source language errors that would cause generation of an incorrect object program. The TAC assembly phase is bypassed, and control is returned to SYS via 1NXTCON.
2	ILLEGAL INPUT TAPE NO.	An attempt was made to use Tape 6 as an input tape. Control is transferred to 1ERRDMP.

COBOL — Common Business-Oriented Language Compiler

FUNCTION

Initiates a COBOL compilation

FORMAT

L	Location	Command	Address and Remarks
	format	COBOL	*source,lib,subs*

REMARKS

The parameters, action, and type-outs are the same as those for the TAC control instruction (page 3-3.1) except that COBOL statements from the input tape are converted to TAC language and placed on Tape 6. Following compilation of the TAC language into machine language, the program is placed on Tape 2. Transfers from Tape 2 to the program tape and/or Tape 5 take place the same as for TAC compilations.

IBIT SETTINGS

34 A one in IBIT 34 indicates that the COBOL compiler will interpret the source language (from the COBOL library and input deck) as follows:

COLUMNS 73-80 will be ignored

The 8-4 punch will be interpreted as the quote (") character rather than the semicolon (;) character

35 ~~A one in IBIT 35 indicates that additional information will be supplied to the output tape (Tape 5) for printing to aid in debugging the compiler.~~ During a COBOL compilation, IBIT 35 is reserved for COBOL maintenance use

36 A one in IBIT 36 indicates that the COBOL compiler will generate a TAC program, but will bypass the TAC assembly phase. TAC COMPILATION BYPASSED will be typed on the Console Typewriter (refer to Normal Type-Out 1, below).

40 A one in IBIT 40 indicates that all references to the COBOL library will be ignored.

SPECIAL TAPE

The COBOL library may be mounted on Tape 9, instead of being combined with the regular TAC library on Tape 7. If Tape 9 is used for the COBOL library, CONBITS 19 must be set to 1. The setting of this CONBIT is an *installation option* and must be set at the time the installation system tape is generated.

Tape 10 may be used as a scratch tape instead of Tape 4, if CONBITS 20 is set to one. The setting of this CONBIT is also an *installation option* and must be set at the time the installation tape is generated.

EXAMPLE

L	Location	Command	Address and Remarks
	REL	COBOL	MAGTAPE,LIB

Explanation: Execution of this instruction causes the COBOL program on Tape 8 to be compiled into REL format.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	TAC COMPILATION BY-PASSED	IBIT 36 is set to one. The COBOL compiler produced a TAC program, but bypassed the TAC compilation phase. The TAC program is on Tape 6. Control is returned to SYS via 1NXTCON.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	ILLEGAL INPUT TAPE NO.	One of the following occurred: • An attempt was made to use Tape 6 as an input tape. • An attempt was made to use Tape 9 as an input tape *and* CONBITS 19 is set to 1. • An attempt was made to use Tape 10 as an input tape *and* CONBITS 20 is set to 1. Control is returned to SYS via 1ERRDMP.
2	NO LIB	The COBOL library, required by the source program, is not mounted on the correct tape.
3	ECOBOL	The COBOL compiler detected one or more source language errors that would cause generation of an incorrect object program. The TAC assembly phase is bypassed, and control is returned to SYS via 1NXTCON.
4	MACHERR	A machine or compiler error was detected. The computer halts following this type-out and the operator must manually return control to SYS.
5	*id* MISSING	One of the compiler phases was not correctly identified, thus indicating a faulty system tape.
6	BAD SEC WORD	One of the compiler phases cannot be loaded, thus indicating a faulty system tape.

REPORT — Report Generator Compiler

FUNCTION

Initiates a Report Generator Compilation

FORMAT

L	Location	Command	Address and Remarks
	format	REPORT	*source,lib,subs*

REMARKS

The parameters, action, and type-outs are the same as those for the TAC control instruction (page 3-3.1) except that Report Generator statements from the input tape are converted to TAC language and placed on Tape 6. Following compilation of the TAC language to machine language, the program is placed on Tape 2. Transfers from Tape 2 to the program tape and/or Tape 5 take place the same as for TAC compilations.

SPECIAL TAPE

Tape 10 may be used as a scratch tape instead of Tape 4, if CONBITS 20 is set to one. The setting of this CONBIT is an *installation option* and must be set at the time the installation system tape is generated.

EXAMPLE

```
L  Location  Command  Address and Remarks

   ABS       REPORT   MAGTAPE,LIB,12
```

Explanation: Execution of this instruction causes the Report Generator program on Tape 8 to be compiled into ABS format. Library routines are located on Tape 12 only.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	ILLEGAL INPUT TAPE NO.	An attempt was made to use Tape 10 as an input tape *and* CONBITS 20 is set to 1. Control is returned to SYS via 1ERRDMP.

LOADING FUNCTIONS

The loading and initiating of program runs is the function of the third group of control instructions. Three instructions in this category, RPL, ABS, and REL, specify the format of the particular programs which are to be loaded. Two instructions, ORIGIN and COMMON, affect loading of REL programs and are used respectively to set the end memory location boundary for a program and to specify Common storage areas.

RPL and ABS programs are loaded in a forward sequence from the starting location of the program near the beginning of memory, extending toward the end of memory. REL programs are also loaded in a forward sequence, but normally into the end of memory, so that the last word of the program is loaded into the last word of memory. Common storage for REL programs normally begins (as arbitrarily assigned by SYS) at location 1000_8 and may extend to any length required within the space available.

The figure on page 3-4.0.1 depicts the location of the three types of programs loaded within memory.

ONE-WORD SEARCH

A one-word-identity search may be used to locate a program with an identity of more than eight characters. If this method is used, the *id* must be terminated by a break character because trailing spaces are significant.

For example, a program with the identity:

PROGAΔΔΔDATEΔ1ΔΔ

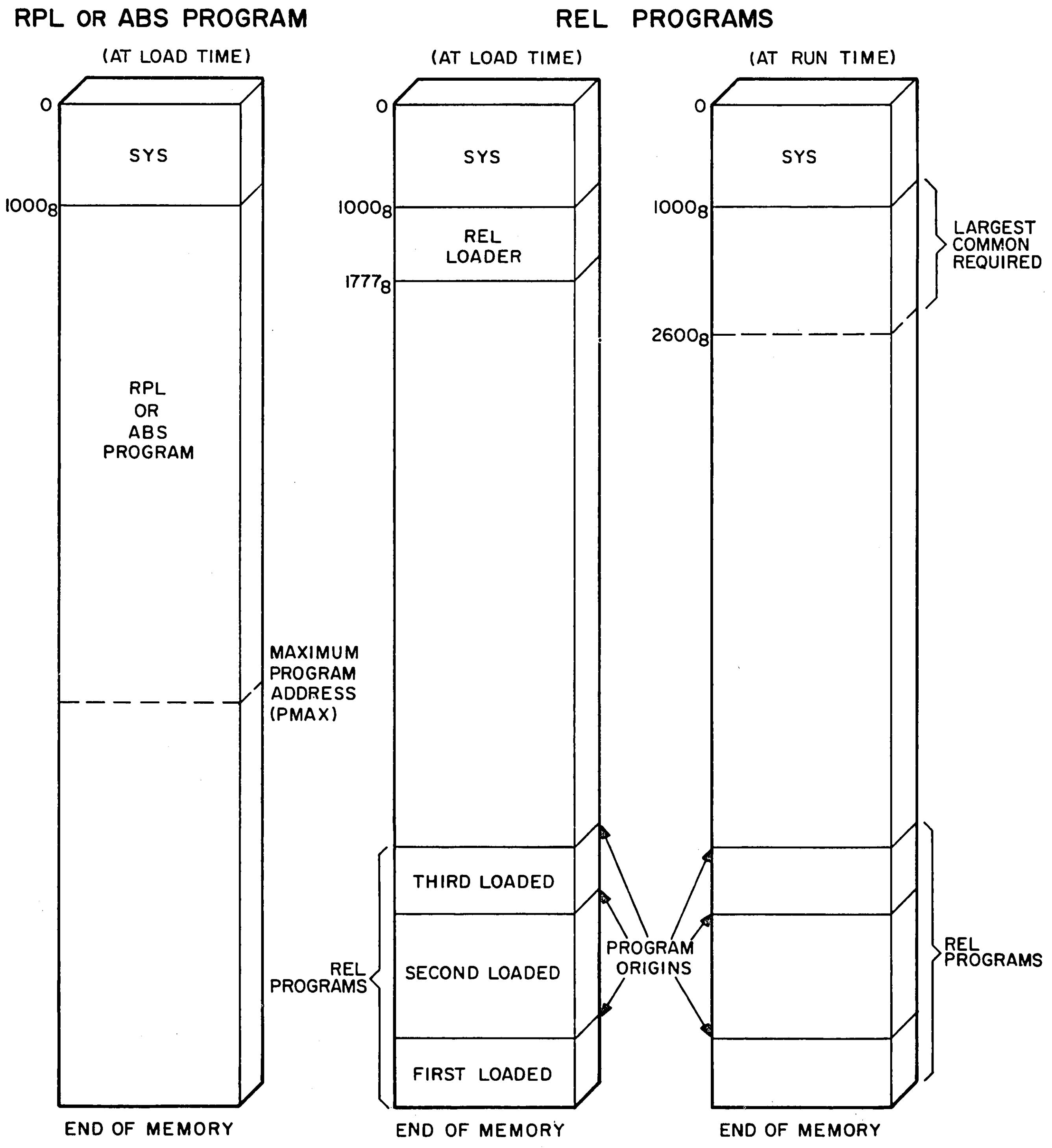
appears in RPL format on Tape 4 and has a unique first word of identity. The program may then be loaded by the control instruction:

L	Location	Command	Address and Remarks
		RPL	4,PROGA$

The same control instruction without the dollar sign:

L	Location	Command	Address and Remarks
		RPL	4,PROGA

results in the type-out NOT HERE on the Console Typewriter. The omission of a break character indicates that the second word of the ID is all spaces and no match can occur.



Allocation of Memory for Program Loading

RPL — RPL Loader Call

FUNCTION

Locates and loads an RPL program.

FORMAT

L	Location	Command	Address and Remarks
		RPL	*t,id,go,addr*

PARAMETERS

t A decimal number, 0 through 4 or 6 through 15, which designates the tape containing the RPL program to be loaded. Tape 8 is excluded if input is via magnetic tape, and if used, a Control Line Error will result.

id 16 or fewer alphanumeric characters including spaces which designate the program identity. If the program is not located, Error Type-Out 1 (page 3-4.1.2) will occur. If the *id* is eight characters or fewer, only the first eight characters of each RPL program on Tape *t* will be compared with the *id.* (Refer to ONE-WORD SEARCH, page 3-4.)

go (OPTIONAL). Indicates immediate or delayed running of a program. If GO is written, control is to be transferred to the program's starting address immediately following successful completion of the loading process. If the parameter is omitted, SYS saves the starting address and executes the next control instruction. A subsequent SYS "JMP *" instruction may be used to transfer control to the program's starting address (refer to JMP, page 3-8.7).

addr (OPTIONAL). An octal number to be added to the normal loading addresses of the program being called. The combined effective address may not be less than 1000_8; otherwise, an illegal load address will result. If program TRACK would normally be loaded into memory location 1000_8, the following instruction would cause it to be loaded into memory starting at location 1050_8:

L	Location	Command	Address and Remarks
		RPL	4,TRACK,,50

ACTION

Upon execution of the RPL control instruction, the following action takes place:

- Tape *t* is searched forward for the RPL program designated by *id* until the *id* or a sentinel block of Z's is found. If the sentinel block is encountered first, the tape is searched backward until the *id* or the beginning-of-tape sensing strip is located. If the *id* is not found, Error Type-Out 1 (page 3-4.1.2) will occur and control will be transferred to 1ERRDMP (page 4-4).

- If Tape *t* is the System tape (Tape 1), the search is made forward only. If *t* is Tape 2 and *id* is not found, the tape will be positioned by being rewound and spaced forward one block before the type-out takes place.
- When the designated program is located, it is loaded into memory.
- If GO is not specified, READY will be typed on the Console Typewriter and control will be transferred to 1NXTCON (page 4-7).
- If GO is specified, control will be transferred to the program's starting address.

EXAMPLE ONE

This example illustrates the use of an RPL call with the program to be executed immediately.

```
L  Location  Command  Address and Remarks

              RPL     4,SOLOMON23,GO
```

Explanation: Execution of this instruction causes RPL program SOLOMON23 to be loaded from Tape 4 and control to be transferred immediately to its starting address.

EXAMPLE TWO

This example illustrates the use of an RPL call with delayed execution of the program.

```
L  Location  Command  Address and Remarks

              RPL     4,SOLOMON23
               •
               •      (Other SYS control instructions)
               •
              JMP     *
```

Explanation: Execution of these instructions causes RPL program SOLOMON23 to be loaded from Tape 4. When the SYS "JMP *" control instruction is encountered, control will be transferred to the program's starting address.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	READY	A program has just been loaded and is ready for execution.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	*id* NOT HERE	The program specified by *id* cannot be found on the designated tape. Control is transferred to 1ERRDMP (page 4-4).
2	BAD SECTION WORD	The length of the RPL section is stated incorrectly. Control is transferred to 1ERRDMP.
3	ILLEGAL LOAD ADDRESS	An attempt was made to load a program into memory occupied by SYS. Control is transferred to 1ERRDMP.

ABS — ABS Loader Call

FUNCTION

Locates and loads an ABS program.

FORMAT

L	Location	Command	Address and Remarks
		ABS	*t,id,go,addr*

REMARKS

The parameters and action are the same as those for the RPL control instruction (page 3-4.1) except for the following:

- The program designated by *id* is presumed to be in ABS format.

- An asterisk or 8 may be used as the first parameter in the ABS control instruction, if a binary absolute deck of the program to be loaded follows the control instruction. Input must be via magnetic tape in Image Mode. In this case, *id* is optional.

EXAMPLE

L	Location	Command	Address and Remarks
		ABS	*,,GO

Explanation: Execution of this instruction causes the deck of binary cards which follows the instruction on Tape 8 to be loaded into memory. After the program is loaded, control is transferred to the program's starting address. The second parameter, which must be indicated by an extra comma, is ignored.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	READY	A program has just been loaded and is ready for execution.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	*id* NOT HERE	The program specified by *id* cannot be found on the designated tape.
2	CHECKSUM ERROR *data*	A discrepancy in checksums was found by the SYS loader, where *data* indicates Columns 1-8 of the illegal card.
3	ILLEGAL CARD *data*	A card unacceptable to the SYS loader has been encountered, where *data* indicates Columns 1-8 of the illegal card.
4	ILLEGAL CSA *data*	An attempt was made to load a program into an illegal memory address, where *data* indicates Columns 1-8 of the illegal card.

Control will be transferred to 1ERRDMP (page 4-4), after the occurrence of any of the above error type-outs.

REL — REL Loader Call

FUNCTION

Locates and loads a REL program.

FORMAT

L	Location	Command	Address and Remarks
		REL	t_1,*id*,*go*,*list*,*subs*,t_2,...,t_n

PARAMETERS

t_1 The tape containing the REL program to be loaded, designated by one of the following:

- A decimal number, 0 through 4 or 6 through 15. Tape 8 is excluded if input is via magnetic tape, Code Mode.
- An asterisk (*), which is equivalent to 8. Input then must be via magnetic tape Image Mode, and a deck of REL program cards must follow the instruction.
- A double asterisk (**), which indicates that only subroutines are to be loaded. If (**) is specified, the SUBS parameter must be used and the *id* parameter must be omitted.

id 16 or fewer alphanumeric characters which designate the program identity. Spaces are significant. If the *id* is eight characters or fewer, only the first eight characters of each REL program ID on Tape *t* will be compared with the *id*. (Refer to ONE-WORD SEARCH, page 3-4.)

(OPTIONAL). The *id* parameter may be omitted if t_1 is an asterisk (*) or 8 and input is via magnetic tape Image Mode. The *id* parameter must not be used if a double asterisk (**) is specified as t_1.

go (OPTIONAL). Indicates immediate or delayed running of a program. If GO is written, control will be transferred to the program's starting address immediately following successful completion of the loading process. If the parameter is omitted, SYS will save the starting address and will execute the next control instruction. A subsequent SYS "JMP *" control instruction may be used to transfer control to the program's starting address (refer to JMP, page 3-8.7).

list (OPTIONAL). The parameter, written as LIST, indicates that all program SYMBOUTS and REFOUTS which have been placed in a NAME/SYMBOL list up to and including this loading are to be edited for printing and written on Tape 5, immediately following the successful completion of the loading process. If the specified program is not loaded successfully, the NAME/SYMBOL list, if any, will be processed as above regardless of whether or not LIST was specified.

subs (OPTIONAL). The parameter, written as SUBS, indicates that the binary library tape(s) will be searched for required subroutines to be loaded if there are any undefined symbols remaining at the completion of the loading process.

This parameter is required if a double asterisk is specified as t_1.

$t_2,\ldots,t_n$ (OPTIONAL). One or more decimal numbers — 0, 2, 3, 4, and 6 through 15 — which designate the tape(s) containing binary subroutines that may be used by the program just loaded. Tape 8 is illegal if input is via magnetic tape. Tape 7 is always assumed and should not be specified.

If 7 is specified, it must be written last in the list of library tapes. Tapes are searched in the order listed in the parameter.

This parameter, if used, must be preceded by the SUBS parameter or a Control Line Error will result.

ACTION

The action is the same as that for the RPL control instruction (page 3-4.1), except for the following:

- The program designated by *id* is presumed to be in REL format.
- If undefined symbols exist, and the SUBS parameter is specified, the library tape(s) will be searched* for those subroutines which define these symbols. After being located, the particular subroutines will be loaded. If undefined symbols still exist following loading of the designated subroutines, and GO is specified, Error Type-Out 10 will occur on the Console Typewriter and the NAME/SYMBOL list will be edited and placed on Tape 5 for printing on the High-Speed Printer in Data Select 0. Control will then be transferred to 1ERRDMP (page 4-4).
- If there are undefined symbols, the SUBS parameter is not specified, and GO is specified, the same error procedure as above will occur.
- If a symbol is defined more than once, it will not be considered an error unless the symbol is referenced. The only exception is the symbolic location LOADSETS.AVAILMEM, which may be defined only once.
- The unique symbol LOADSETS.AVAILMEM has a special definition. If it appears first as a referenced symbol, it is immediately defined as being the location one less than the origin of the program just loaded. This origin is then reduced by one to preserve the location. If this symbol appears first as a defined symbol, that definition is given to it.
- If location LOADSETS.AVAILMEM appears on the NAME/SYMBOL list, its left and right address portions respectively will contain, after loading, the sum of Common origin and the size of Common, and the address of the smallest program origin greater than zero. (Refer to SEGMENT and MASTER, pages 3-5.1 and 3-5.2.)

*Binary library tapes must contain a card of A's preceding the first binary subroutine and a card of Z's following the last subroutine.

- If the LIST parameter is specified, the NAME/SYMBOL list will be edited and placed on Tape 5 for off-line printing.

  Upon completion of a successful loading, COMORG (page 1-6) and COMSIZE (page 1-6) will have been updated. COMORG will contain the origin of Common in its left address portion and the origin of the program or subroutine last loaded in its right address portion. COMSIZE will contain the largest size of Common used to this point, or 1000_8 (whichever is larger) in its left address portion.

REMARKS

For making corrections to relocatable binary programs, refer to REMARK 2 of OCT (page 3-7.3).

EXAMPLE ONE

This example illustrates the loading of a single REL program.

```
L  Location  Command  Address and Remarks

             REL      4,TRAJ,GO,LIST,SUBS,0,9
```

Explanation: Execution of this instruction causes the REL program, TRAJ, to be loaded into the end of memory from Tape 4. Subroutines required to satisfy undefined program symbols are to be loaded first from Tapes 0 and 9, and then, if necessary, from Tape 7. After the NAME/SYMBOL list is edited and placed on Tape 5 for off-line printing, control is transferred to the program's starting address.

EXAMPLE TWO

This example illustrates the joint loading of two REL programs.

```
L  Location  Command  Address and Remarks

             REL      4,DIFF1$
             REL      4,DIFF2,GO,LIST,SUBS
```

Explanation: Execution of these instructions causes the REL program, DIFF1, to be loaded from Tape 4 into the end of memory and a second REL program, DIFF2, to be loaded into memory adjoining DIFF1 (refer to the figure on page 3-4.0.1). Subroutines required by either program to satisfy undefined symbols are to be loaded from Tape 7. When both programs and all subroutines are loaded, the NAME/SYMBOL list is edited and placed on Tape 5 for off-line printing. Control is then transferred to the starting address of DIFF2.

EXAMPLE THREE

This example illustrates the method of making corrections to a REL program at the time the program is loaded.

```
L  Location  Command  Address and Remarks

             REL      4,JOE
   245P      ALPHA    PROGRAMA
             REL      **,,GO,LIST,SUBS
```

Explanation: Execution of these instructions causes program JOE to be loaded without subroutines from Tape 4. The alphanumeric correction will then be placed into location 245_8, relative to program origin. The second REL call is then used to load the subroutines, cause the NAME/SYMBOL list to be edited and placed on Tape 5 for off-line printing, and control to be transferred to the starting address of program JOE. If SUBS had been specified with the first REL call, the change would have been incorrectly made relative to the origin of the last subroutine loaded. (Refer to REMARK 2 of OCT, page 3-7.3.)

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	READY	A program has just been loaded and is ready for execution.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	CHECKSUM ERROR *data*	A discrepancy in checksums was found by the SYS loader.
2	EXCEEDS MEMORY *data*	The program to be loaded (with subroutines) exceeds memory capacity.
3	ILLEGAL CARD *data*	A card unacceptable to the SYS loader has been encountered.
4	ILLEGAL CSA *data*	An attempt was made to load a program into an illegal memory address.
5	ILLEGAL 1ST CARD *data*	The first card of a program to be loaded was not a PMAX or TUG absolute card.
6	ILLEGAL MODIFIER *data*	An unacceptable address modifier has been encountered by the SYS loader.
7	SYMBOL REDEFINED *data*	A program name symbol has been redefined at load time.
8	ILLEGAL SUBLIB TAPE NO.	Tape 1, 5, or 8 (if magnetic tape input) was specified as a binary library tape in a REL control instruction.
9	TOO MANY SYMBOLS *data*	The NAME/SYMBOL list of the relocatable loader has been filled to capacity.

CONTINUED

CONSOLE TYPEWRITER ERROR TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
10	UNDEFINED SYMBOL	The program just loaded contains an undefined symbol, and GO is specified.
11	*id* NOT HERE	The program designated by *id* cannot be found on the specified tape.

Control will be transferred to 1ERRDMP (page 4-4) after the occurrence of any of the above error type-outs. The type-out *data* indicates Columns 1-8 of the illegal card.

ORIGIN — Set the Relocatable Binary Program Loading Origin

FUNCTION

Specifies the higher-order limit in memory for relocatable binary programs which are to be loaded.

FORMAT

L	Location	Command	Address and Remarks
		ORIGIN	*n*

PARAMETER

n An octal number which indicates the higher-order memory limit for the next REL program to be loaded.

ACTION

When the ORIGIN control instruction is executed, the right address portion of COMORG (page 1-6) is set to the value of *n*.

REMARKS

1. The address *n*, specified by the ORIGIN card, will be the end of the next REL program loaded.
2. The value in the right address portion of COMORG can be changed by the execution of (1) another ORIGIN instruction, (2) a REL instruction, and (3) a JOB instruction. (Refer to JOB, page 2-2.1.)

EXAMPLE

Assume that a relocatable binary program with a PMAX of 777_8 (program length) is to be loaded.

L	Location	Command	Address and Remarks
		ORIGIN	16777
		REL	......

Explanation: Execution of this instruction causes the right address portion of COMORG to be preset to 16777_8. During the loading, the program's PMAX (777_8) is subtracted from the right address portion of COMORG (16777_8) to obtain the program's origin — 16000_8. The right address portion of COMORG is then replaced with the new value.

COMMON — Common Storage Area

FUNCTION

Specifies the length and core starting address of the Common storage area to be used by relocatable binary programs.

FORMAT

L	Location	Command	Address and Remarks
		COMMON	*$length_{10}$,$start_8$*

PARAMETERS

$length_{10}$ A decimal number which indicates the number of locations to be set aside for Common storage.

$start_8$ An octal number equal to or greater than 1000 which indicates the starting address of the Common storage area. If a value less than 1000_8 is specified, a Control Line Error will occur.

ACTION

When the COMMON control instruction is executed, the following action takes place:

- The value of *$start_8$* is stored as the left address portion of COMORG (page 1-6).
- If the value of *$start_8$* plus *$length_{10}$* is greater than 2000_8, *$length_{10}$* is stored in the left address portion of COMSIZE (page 1-6).
- If the value of *$start_8$* plus *$length_{10}$* is less than 2000_8, (2000_8 - *$start_8$*) is stored in the left address of COMSIZE. The value of *$length_{10}$* is then ignored, so that the relocatable loader (which occupies locations 1000_8 to 1777_8) is automatically protected by enforcement of the rule that Common and program areas must not overlap.

EXAMPLE

Assume that a relocatable binary program is to be loaded and run.

```
L  Location  Command  Address and Remarks

              COMMON   596,1200
```

Explanation: Execution of this instruction causes the Common storage area to be preset to 596_{10} locations, beginning at location 1200_8.

PHILCO CORPORATION

A SUBSIDIARY OF Ford Motor Company

COMPUTER DIVISION • 3900 Welsh Road, Willow Grove, Pa. • Phone OLdfield 9-7700

February 5, 1963

PSG 63-8

```
T      TMD      1/1T23$
T      DORMS    M/63$
```

put before each SEGRPL macro when loading segments when read tape 8

To: Philco 2000 Installations

Subject: Error in Segment Loader

. . . .

An error has been discovered in the special loader for segments (restricted RPL format). Temporarily, in order to correct this error, bit 23 of SYS.1CONBIT (M/63) should be set to 1 prior to each SEGRPL call. This bit should also be set if the programmer is using the internal loader by jumping to SYS.1INTLD (M/22) with bit 41 of the Q register set to 1.

As an example, if the previous coding were

```
       :
       SEGRPL        12,PROGA,GO
       :
```

the coding should now read

```
       :
       TMD           1/1T23
       DORMS         SYS.1CONBIT
       SEGRPL        12,PROGA,GO
       :
```

It will not be necessary to change programs containing the extra coding when a permanent correction is distributed.

Oscar Boyajian
OSCAR BOYAJIAN
Manager
Program Services

OB:vg

SEGMENTATION FUNCTIONS

Two SYS control instructions, SEGMENT and MASTER, provide the facilities for dividing a relocatable binary program, too large to be run with available memory, into smaller segments. Each of these segments may then be contained individually in the memory area.

SEGMENTATION PROCESS

This process, known as segmentation, generates segments by writing one or more program components (which are loaded as a single unit into memory) onto magnetic tape in a restricted RPL format (See below).

The segmentation process takes place in three phases: first, the compilation of components as individual smaller programs in REL format; second, the generation of segments in RPL format; and third, the loading and running of the segmented program in RPL format. The SYS control instructions SEGMENT and MASTER are used in the second phase.

When segments are used in a running program, they may be independent of one another, or they may contain cross-references. If they are independent, they may be generated one at a time during the second phase of the segmentation process. If any contain cross-references, the referencing must be done either via the Common area of memory or via a Master segment which is contained in memory during the generation of all segments.

The SEGMENT instruction generates segments by writing the components just loaded onto magnetic tape. The MASTER control instruction is used to indicate that the component(s) just loaded are to form a Master segment, and are to remain in memory during the generation of succeeding segments.

LOADING SEGMENTED PROGRAMS

Program segments may be loaded by the SYS internal loader, which may be called by either of the following methods:

- Via the SYS Generator LOADGEN (page 6-2), which creates the necessary coding in TAC language to execute the equivalent of an RPL, ABS, or REL call.
- Via the SYS entry 1INTLD (page 4-20), which provides a transfer of control to the internal loader during the running of the program.

RESTRICTED RPL FORMAT

Restricted RPL format differs from normal RPL format in that there are no half-words within a segment or its sections; that is, each segment or section always starts in the left-half of a word and ends in the right-half of a word. The maximum size for any section to be written in restricted RPL format is $16,383_{10}$ memory locations. Any greater memory area is automatically divided into the correct number of sections.

SEGMENT — Define A Program Segment

FUNCTION

Causes program components loaded into memory in REL format to be written as a segment onto magnetic tape in restricted RPL format (refer to page 3-5).

FORMAT

L	Location	Command	Address and Remarks
		SEGMENT	*t,id,master*

PARAMETERS

t A decimal number — 0, 2, 3, or 6 through 15 — that specifies the tape on which the segment is to be written. If any number other than the above is specified, a Control Line Error will result.

id Eight or fewer alphanumeric characters which indicate the identity of this particular segment. Spaces are significant.

master (OPTIONAL). A parameter written as MASTER which enables a Master segment (page 3-5.2) to be written on tape.

ACTION

Execution of a SEGMENT control instruction initiates the following actions:

- Examines the NAME/SYMBOL list associated with the particular segment of the program to determine if there are any undefined symbols. Undefined symbols cause Error Type-Out 2 (page 3-5.1.2) to be typed on the Console Typewriter.

- Searches for a sentinel block of Z's on Tape *t* if IBIT 0 is set to one prior to the execution of the SEGMENT control instruction, and positions the tape at the start of the block. If IBIT 0 is not set, no search is made and writing will begin wherever the tape is positioned. (Refer to IBIT 0, page 3-8.1.1.)

- Writes that portion of memory between the first non-zero word after the end of Common or 1777_8, whichever is larger, and MASORG (page 3-5.2), if a Master segment has been defined, onto Tape *t* in restricted RPL format. Normal Type-Outs 1 and 2 (page 3-5.1.2) are typed on the Console Typewriter. If a Master segment has not been defined, that portion of memory between the first non-zero word following Common (or 1777_8) and the end of memory will be written onto Tape *t*.

- Writes a sentinel block of Z's onto Tape *t* following the segment, and positions the tape at the beginning of this block. If another segment is to follow, it will be written over the sentinel block and a new sentinel block will be written following the new segment. This procedure is repeated for all segments, including the Master segment, thus insuring the presence of a sentinel block after the final segment only.

- Clears to fixed-point zeroes that portion of memory occupied by the segment after the segment is written onto Tape *t*.

- Clears the NAME/SYMBOL list of all names and symbols defined by the particular segment, excluding the Master segment list, if used.

- Writes onto Tape *t* in RPL format that area between MASORG and the end of memory, if the *master* parameter is used. In addition, it clears the memory area and the NAME/SYMBOL list used by the Master segment.

REMARKS

If a symbol is defined more than once it will not be considered an error unless the symbol is referenced. The only exception is the symbolic location LOADSETS.AVAILMEM (page 3-4.3.1). If this location is defined in the Master segment, it may not be defined in any of the related segments. It may be defined once in every segment if it is not defined in the Master segment.

RESTRICTIONS

The master parameter should be used only once per program after all other segments are written. If used before all segments are loaded, Error Type-Out 2 (page 3-5.1.2) will probably result.

EXAMPLE ONE

This example illustrates the use of SEGMENT without a master parameter. Assume that the first segment (SEG1) of a large program consists of components PROG1 and PROG2, and that these components are in REL format on Tape 4.

```
L  Location  Command   Address and Remarks

                REWIND    6
                REL       4,PROG1
                REL       4,PROG2,,LIST,SUBS
                SEGMENT   6,SEG1
```

Explanation: Execution of these instructions causes the two components to be loaded and written in RPL format onto Tape 6 as SEG1.

EXAMPLE TWO

This example illustrates the use of SEGMENT with the master parameter. Assume that EXEC is a Master segment. APROG and BPROG constitute one segment (SEGA) of a program and CPROG and DPROG form another segment (SEGB).

```
L  Location  Command   Address and Remarks

              REWIND    6
              REL       4,EXEC,,LIST,SUBS
              MASTER
              REL       4,APROG
              REL       4,BPROG,,LIST,SUBS
              SEGMENT   6,SEGA
              REL       4,CPROG
              REL       4,DPROG,,LIST,SUBS
              SEGMENT   6,SEGB
              SEGMENT   6,EXEC,MASTER
```

Explanation: Execution of these instructions causes the three segments to be written in RPL format onto Tape 6.

If a "JMP *" control instruction is written in place of the final SEGMENT instruction (SEGMENT 6,EXEC,MASTER), the overall program will be run, but the Master segment will not be saved on tape.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	COMMON *n*	The program(s) just segmented contain *n* words of Common storage.
2	*id* RPL *n* BLOCKS	A segment has just been written on tape in RPL format, identified as *id* and containing *n* blocks.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	NO MASTER	A master parameter is present in a SEGMENT control instruction, but no MASTER control instruction had been encountered. Control is transferred to 1XCONER (page 4-18).
2	UNDEFINED SYMBOL	The program being segmented has an undefined symbol. Control is transferred to 1ENDJOB (page 4-6).

MASTER — Define a Master Segment

FUNCTION

Indicates to SYS that (1) a segmenting process using a Master segment is about to take place, and (2) that a Master segment consisting of one or more REL components has been loaded and is to remain in memory during the forthcoming loading and writing of the various segments. The Master segment remains in memory during segmentation so that symbolic cross-references may be made with succeeding segments.

FORMAT

L	Location	Command	Address and Remarks
		MASTER	

PARAMETERS

None

ACTION

Execution of a MASTER control instruction initiates the following actions:

- Assigns the origin of the Master segment's final component loaded to the right address portion of MASORG (MASter ORiGin). The Master segment, therefore, extends from MASORG to the end of memory. The maximum length of succeeding segments is that area remaining between the end of any Common storage area or Location 1777_8, whichever is larger, and MASORG.
- Stores the Master segment's starting address in XSYS.MASJMP, a location which remains within SYS in memory at all times. This address is taken from the last relocatable END card of a Master component to be loaded that contains a starting address. If no component contains a starting address, XSYS.MASJMP will be zero-filled, preventing the use of a SYS ''JMP *'' control instruction for transfer of control (refer to REMARKS below).

REMARKS

1. If a MASTER control instruction is given, control may be transferred to a Master segment still in memory by the use of a ''JMP *'' control instruction, upon completion of the writing of an entire segmented program.
2. Refer to REMARKS of SEGMENT (page 3-5.1.1).

EXAMPLE

Assume that the Master segment consists of two components in relocatable binary format with ID's of EXEC1 and EXEC2, both contained on Tape 4.

```
L  Location  Command  Address and Remarks

             JOB      XRAY
             REL      4,EXEC1
             REL      4,EXEC2,,LIST,SUBS
             MASTER
```

Explanation: Execution of these instructions causes the two components to be loaded into memory, forming a Master segment and retaining the segment in memory until all other segments are loaded and written onto tape. MASORG contains the origin of the last component of the Master segment loaded, while XSYS.MASJMP contains one of the following:

- The starting address of EXEC1, if EXEC2 has no starting address.
- The starting address of EXEC2, if it contains a starting address. EXEC2 takes precedence because it is the last to be loaded.
- Zeroes, if neither EXEC1 nor EXEC2 contain a starting address.

Note: TAC library subroutines never contain starting addresses.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	MASTER ORIGIN *addr*	A Master segment has been loaded into memory and its origin has been stored in MASORG as an octal *addr*.

TAPE HANDLING FUNCTIONS

The fifth group of control instructions is concerned specifically with magnetic tape operations. These operations include rewinding tapes, reading tapes forward and backward, writing on tapes, writing sentinels, and locating sentinels.

REWIND — Rewind Magnetic Tape

FUNCTION

Rewinds magnetic tapes.

FORMAT

L	Location	Command	Address and Remarks
		REWIND	*t,...,t*

PARAMETERS

t One or more decimal numbers, 0 through 15, which indicate the tape(s) to be rewound.

ACTION

When the REWIND control instruction is executed, the tape(s) indicated by *t,...,t* are rewound.

REMARKS

If the tape is unavailable, the computer will loop continuously on the REWIND order until the tape becomes available or the operator intervenes.

EXAMPLE

L	Location	Command	Address and Remarks
		REWIND	3,9

Explanation: Execution of this instruction causes Tapes 3 and 9 to be rewound.

REWINDLO — Rewind With Lockout

FUNCTION

Rewinds magnetic tapes with lockout.

FORMAT

L	Location	Command	Address and Remarks
		REWINDLO	*t*,...,*t*

PARAMETERS

t One or more decimal numbers, 0 through 15, which indicate the tape(s) to be rewound with lockout.

ACTION

When the REWINDLO control instruction is executed, the tape(s) indicated by *t*,...,*t* are rewound with lockout.

REMARKS

If the tape is unavailable, the computer will loop continuously on the REWINDLO order until the tape becomes available or the operator intervenes.

EXAMPLE

L	Location	Command	Address and Remarks
		REWINDLO	9,11,12,13

Explanation: Execution of this instruction causes Tapes 9,11,12, and 13 to be rewound with lockout.

READF — Read Magnetic Tape Forward

FUNCTION

Spaces forward and/or reads blocks forward from a specified magnetic tape into memory.

FORMAT

L	Location	Command	Address and Remarks
		READF	*t,nbs,nbp,addr*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be spaced and/or read forward.

nbs (OPTIONAL). A decimal number which indicates the number of blocks to be spaced forward. If *nbp* below is omitted, *nbs* must be specified.

nbp (OPTIONAL). A decimal number which indicates the number of blocks to be processed forward. If *nbs* is omitted, *nbp* must be specified.

addr (OPTIONAL). An octal number which indicates the starting memory location into which the data on the tape is to be read. If *nbp* is omitted or is zero. *addr* may be omitted.

ACTION

When the READF control instruction is executed, tape *t* is spaced forward *nbs* number of blocks, and *nbp* number of blocks are read into memory beginning at location *addr*.

REMARKS

Checks for completion and fault routines are incorporated within the READF, READB, and WRITE control instructions. If a non-recoverable error occurs, the computer will halt and any further action will be determined by the operator or operating system. If the Advance Bar is pressed, control will be transferred to 1SYSIN (page 4-3).

EXAMPLE ONE

This example illustrates the use of a space-read command.

L	Location	Command	Address and Remarks
		READF	6,2,5,1500

Explanation: Execution of this instruction causes Tape 6 to be spaced forward two blocks, and the next five blocks to be read into memory starting at location 1500_8.

EXAMPLE TWO

This example illustrates the use of a read-only command.

L	Location	Command	Address and Remarks
		READF	11,,36,2500

Explanation: Execution of this instruction causes 36 blocks of Tape 11 to be read into memory starting at location 2500_8. No spacing of blocks is to take place prior to the read.

EXAMPLE THREE

This example illustrates the use of a space-only command.

L	Location	Command	Address and Remarks
		READF	12,160

Explanation: Execution of this instruction causes Tape 12 to be spaced forward 160 blocks. No reading is to take place.

READB — Read Magnetic Tape Backward

FUNCTION

Spaces backward and/or reads blocks backward from a specified magnetic tape into memory.

FORMAT

L	Location	Command	Address and Remarks
		READB	*t,nbs,nbp,addr*

REMARKS

The parameters, action, and remarks concerning READB are the same as those for the READF instruction (page 3-6.3), except that spacing and/or reading is performed backward on the tape.

EXAMPLE

L	Location	Command	Address and Remarks
		READB	9,5,10,3200

Explanation: Execution of this instruction causes Tape 9 to be spaced backward 5 blocks, and the next 10 blocks to be read backward into memory starting at location 3200_8.

WRITE — Write on Magnetic Tape

FUNCTION

Spaces forward and/or writes a specified number of blocks from memory onto an output magnetic tape.

FORMAT

L	Location	Command	Address and Remarks
		WRITE	*t,nbs,nbp,addr*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be spaced forward and/or written upon.

nbs (OPTIONAL). A decimal number which indicates the number of blocks to be spaced forward. If *nbp* below is omitted, *nbs* must be specified.

nbp (OPTIONAL). A decimal number which indicates the number of blocks to be written. If *nbs* is omitted, *npb* must be specified.

addr (OPTIONAL). An octal number which indicates the starting memory location from which information is to be written on tape. If *nbp* is omitted or is zero, *addr* may be omitted.

REMARKS

The remarks and action for the WRITE control instruction are the same as those for the READF instruction (page 3-6.3), except that writing onto tape rather than reading from tape is performed.

EXAMPLE

L	Location	Command	Address and Remarks
		WRITE	9,3,6,1750

Explanation: Execution of this instruction causes Tape 9 to be spaced forward three blocks, and the next six blocks to be written starting from location 1750_8.

WRTSENT — Write Sentinel on Magnetic Tape

FUNCTION

Writes a sentinel block of 128 words of a designated configuration onto a specified tape.

FORMAT

L	Location	Command	Address and Remarks
		WRTSENT	*t,sent*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be written upon.

sent Eight or fewer alphanumeric characters which compose the sentinel to be used as each of the 128 words of the sentinel block. All characters except break characters (, . ; /) are acceptable. Spaces are ignored. If fewer than eight characters are used, the sentinel will be right-justified and left-filled with zeroes. If more than eight characters are used, a Control Line Error (page 4-18) will result.

ACTION

When the WRTSENT control instruction is executed, a sentinel block of 128 words of *sent* is written onto Tape *t*. Tape *t* remains positioned following the sentinel.

EXAMPLE ONE

This example illustrates the use of a WRTSENT command with an eight-character sentinel.

L	Location	Command	Address and Remarks
		WRTSENT	6,ZZZZZZZZ

Explanation: Execution of this instruction causes a sentinel block containing 128 words of Z's to be written onto Tape 6.

EXAMPLE TWO

This example illustrates the use of a WRTSENT command with a sentinel of fewer than eight characters.

L	Location	Command	Address and Remarks
		WRTSENT	9,XXX

Explanation: Execution of this instruction causes a sentinel block containing 128 words of 00000XXX to be written onto Tape 9.

LOCSENT — Locate a Sentinel on Magnetic Tape

FUNCTION

Searches forward on a tape for a sentinel block which contains a designated sentinel as each of the first 120 words.

FORMAT

L	Location	Command	Address and Remarks
		LOCSENT	*t,sent*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be searched.

sent Eight or fewer alphanumeric characters, excluding break characters, that compose the specific sentinel word. Spaces are ignored. If fewer than eight characters are used, the sentinel will be right-justified and left-filled with zeroes.

ACTION

Execution of the LOCSENT control instruction initiates the following action:

- Tape *t* is searched forward for a sentinel block with the parameter *sent* as each of its first 120 words.
- When the designated sentinel block is found, tape *t* is positioned immediately past the sentinel block.
- If the sentinel is not found, the computer will encounter a serious tape error. Any action taken depends upon the operator. If the Advance Bar is pressed, control will be transferred to 1SYSIN (page 4-3).

EXAMPLE

L	Location	Command	Address and Remarks
		LOCSENT	6,ZZZZZZZZ

Explanation: Execution of this instruction causes Tape 6 to be searched forward for a sentinel block which contains all Z's in the first 120 words. After the sentinel block is located, the tape is positioned immediately past the block. If the sentinel block is not located, the computer will halt (as explained previously in ACTION).

WRTRPL — Write On Magnetic Tape In RPL Format

FUNCTION

Writes words between designated memory locations onto magnetic tape in restricted RPL format (refer to page 3-5) and assigns a program identity to the data written as a new RPL.

FORMAT

L	Location	Command	Address and Remarks
		WRTRPL	*t,id,start,end,jmp,addr, locsent, wrtsent*$

PARAMETERS

t A decimal number — 0 through 15 excluding 1 and 5 — which indicates the tape on which the RPL is to be written. If any number other than the above is specified, a Control Line Error (page 4-18) will be indicated. Tape 8 is illegal if input is via magnetic tape.

id Sixteen or fewer alphanumeric characters which indicate the identity of the new RPL. Spaces are significant. If fewer than 16 characters are specified, the ID will be left-justified and right-filled with spaces (Δ).

start An octal number which indicates the first memory location from which information is to be written on tape.

end An octal number which indicates the last memory location from which information is to be written on tape.

jmp (OPTIONAL). The starting address of the new RPL. One of the following may be selected:

1. An octal address.
2. An asterisk (*), which indicates that the address stored in 1NTRYJA (location 60_8) will be the starting address. 1NTRYJA normally contains the address of a location in the current object program to which control is to be transferred from SYS. (Refer to page 3-7.1.1.)

Note: If *jmp* is omitted, the starting address will be that address indicated by *start.*

addr (OPTIONAL). An octal number which is added to the *start* parameter to designate the normal loading address of the new RPL program. The number is also added to the address specified by the *jmp* parameter. The addition of the values takes place modulo machine size. The RPL will still be written beginning at the address specified by *start.*

locsent (OPTIONAL). Eight or fewer alphanumeric characters (excluding break characters and spaces) which designate the first 120 words of a sentinel block to be located. If *locsent* is omitted, a sentinel block of Z's is assumed.

wrtsent (OPTIONAL). Eight or fewer alphanumeric characters (excluding break characters and spaces) which designate the first 120 words of a sentinel block to be written following the new RPL. If *wrtsent* is omitted, a sentinel block of Z's is assumed.

Note: If fewer than eight characters are specified for either *locsent* or *wrtsent,* the sentinel word will be right-justified and left-filled with zeroes. Spaces are ignored.

PRESET REQUIREMENTS

If Tape *t* is to be searched for a sentinel block before the new RPL is written, one of the following options may be selected:

1. The search will be made for a sentinel block containing the *locsent* configuration in its first 120 words if IBIT 0 is zero and the *locsent* parameter is specified.
2. The search will be made for a sentinel block containing all Z's in its first 120 words if IBIT 0 is set to one — regardless of whether the *locsent* parameter is present or omitted.
3. No sentinel search will take place if both IBIT 0 is zero and *locsent* is omitted.

ACTION

When the WRTRPL control instruction is initiated, the following action takes place:

- A sentinel search may or may not take place according to Preset Requirements, described above.
- If the designated sentinel is not located, the sentinel search will continue until a non-recoverable tape error is encountered or the program is halted by an operator.
- If the designated sentinel is found, the tape will be positioned at the beginning of the sentinel before writing takes place. If a sentinel search is not specified, writing will take place wherever the tape is positioned.
- All information between memory locations *start* and *end* will be written on Tape *t* in restricted RPL format with a program identity of *id.*
- Two sentinel blocks, designated by *wrtsent,* are written on the output tape following the new RPL.
- Tape *t* is backspaced two blocks.

- ID *id* RPL BLX *n* ON TAPE *t* is typed as two lines on the Console Typewriter, where *id* is the program identity, *n* is the total decimal number of blocks written, and *t* is the tape on which the RPL is written.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	ID *id* RPL BLX *n* ON TAPE *t*	An RPL program designated by *id*, containing *n* decimal blocks, has been written onto Tape *t*.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	CONT. LINE ERR.........	One of the following errors was detected in the parameters of a WRTRPL control instruction: • A *t*, *id*, *start*, or *end* parameter was not given. • An illegal tape number (1 or 5) was given. • Too many characters appear in *id*. • The parameter *start* is less than 1000_8. • The parameter *end* is less than *start*. • The address in *start*, *end*, *jmp*, or *addr* is not an octal value. • The address in *start*, *end*, *jmp*, or *addr* is larger than memory size. Control has been transferred to 1XCONER, which caused the Console Typewriter to type the above statement, and a post-mortem dump to be given. (Refer to pages 4-18 and 4-4.)

EXAMPLE ONE

This example illustrates the use of WRTRPL without an offset address.

```
L Location Command  Address and Remarks

                  WRTRPL  9,GOODΔBYE,1200,2000,,,XRAY,BAKER$
```

Explanation: Execution of this instruction causes the contents of memory locations from 1200_8 to 2000_8 to be written in restricted RPL format onto Tape 9. The program identity of the new RPL is GOODΔBYE. The address specified by *start*, 1200_8, is to be used as the starting address for the RPL, as the *jmp* parameter is omitted. Assuming that IBIT 0 was zero prior to the execution of the WRTRPL instruction, Tape 9 is first searched for a sentinel block containing 0000XRAY in its first 120 words prior to writing the RPL. Two sentinel blocks, each containing 000BAKER in their first 120 words, are written following the new RPL on Tape 9. (If IBIT 0 had been set to one in this example, the search would have been made for a sentinel block of Z's – regardless of the configuration within *locsent*.)

EXAMPLE TWO

This example illustrates the use of WRTRPL with an offset address.

```
L Location Command  Address and Remarks

                  WRTRPL  11,HELLO,10000,15000,12000,325$
```

Explanation: Execution of this instruction causes the contents of memory locations from 10000_8 to 15000_8 to be written in restricted RPL format onto Tape 11. The program identity of the new RPL is HELLO. The address specified by *jmp*, plus the offset value specified by *addr* indicates that the starting address of the new RPL is to be 12325_8. The value within *addr* is also added to the starting address of each section written by WRTRPL. No *locsent* parameter is specified, and assuming that IBIT 0 had been set to one, a sentinel block of Z's will be located prior to writing the RPL. No *wrtsent* parameter is given, so two sentinel blocks of Z's will be written following the new RPL.

LOCTACL — Locate a TAC Language Program on Magnetic Tape

FUNCTION

Searches forward on a tape for a specified TAC language program.

FORMAT

L	Location	Command	Address and Remarks
		LOCTACL	*t,id*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be searched.

id Sixteen or fewer alphanumeric characters which designate the program identity. Spaces are significant. If the *id* is eight characters or fewer, only the first eight characters of each TACL program identity will be compared with the *id* during the search. (Refer to ONE-WORD SEARCH, page 3-4.)

ACTION

Execution of the LOCTACL control instruction initiates the following action:

- Tape *t* is searched forward until either the designated TACL program is located or a sentinel block of Z's is encountered. If the sentinel block is encountered prior to the *id*, Error Type-Out 1 (page 3-6.9.1) will occur, and Tape *t* will be rewound.
- After the designated TACL program is located, the tape is positioned immediately prior to the first block of the program.

REMARKS

A TACL program on tape is defined as being in Code Mode, 10 words per card, 12 cards per block, and must begin with an I card as the first card of the first block. The I card must have an I in Column 9 and the Location field must be blank.

EXAMPLE

Assume that Tape 9 contains the following TACL programs:

```
PARTΔ1
PARTΔ2ΔΔ040162
PARTΔ3ΔΔ041062
```

```
L  Location  Command   Address and Remarks

             LOCTACL   9,PARTΔ3ΔΔ041062
             LOCTACL   9,PARTΔ3$
```

Explanation: Execution of either of these instructions causes the third program on the tape to be located.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	*id* NOT HERE	A block of Z's was encountered prior to a program designated by *id*. Control is transferred to 1XCONER (page 4-18).

LOCRPL — Locate an RPL Program on Magnetic Tape

FUNCTION

Searches forward on a tape for a specified RPL program.

FORMAT

L	Location	Command	Address and Remarks
		LOCRPL	*t,id*

PARAMETERS

t A decimal number, 0 through 15, which indicates the tape to be searched.

id Sixteen or fewer alphanumeric characters which designate the program identity. Spaces are significant. If the *id* is eight characters or fewer, only the first eight characters of each RPL program identity will be compared with the *id* during the search. (Refer to ONE-WORD SEARCH, page 3-4.)

ACTION

The LOCRPL control instruction causes the following action to take place:

- Tape *t* is searched forward until either the designated RPL program is located or a sentinel block of Z's is encountered. If the sentinel block is encountered prior to the *id*, Error Type-Out 1 (page 3-6.10.1) will occur, and Tape *t* will be rewound.
- After the designated RPL program is located, the tape will be positioned immediately prior to the first block of the program.

EXAMPLE

Assume that Tape 9 contains the following RPL programs:

PARTΔ1

PARTΔ2ΔΔ040162

PARTΔ3ΔΔ041062

L	Location	Command	Address and Remarks
		LOCRPL	9,PARTΔ3ΔΔ041062
		LOCRPL	9,PARTΔ3$

Explanation: Execution of <u>either</u> of these instructions causes the third program on the tape to be located.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	*id* NOT HERE	A block of Z's was encountered prior to the program designated by *id*. Control is transferred to 1XCONER (page 4-18).

WRTABS – Write on Magnetic Tape in ABS Format

FUNCTION

Writes words between designated memory locations onto magnetic tape in TUG absolute binary card (ABS) format and assigns a program identity to the data written as the new ABS program.

L	Location	Command	Address and Remarks
z		WRTABS	*t, id, start, end, jmp, addr, locsent, wrtsent* $

PARAMETERS

z (OPTIONAL). The character – Z – which indicates that every word (including fixed-point zeroes) in the program will be written onto tape *t*. If this parameter is omitted, leading words of zeroes will be omitted from each absolute binary card produced. If trailing words of zeroes are present, they will be included to complete a card.

t A decimal number – 0 through 15 excluding 1 – which indicates the tape on which the data in ABS card format is to be written. If any number other than the above is specified, a Control Line Error (page 4-18) will be indicated. Tape 8 is illegal if input is via magnetic tape.

id Sixteen or fewer alphanumeric characters which indicate the identity of the new ABS program. Spaces are significant. The first four characters are used to provide an ID in Hollerith code for Columns 1-4 of the ABS card output. If fewer than 16 characters are specified, the configuration within *id* will be left-justified and right-filled with spaces (Δ).

start An octal number which indicates the first memory location from which information is to be written on tape.

end An octal number which indicates the last memory location from which information is to be written on tape.

jmp (OPTIONAL). The starting address of the new ABS program. One of the following may be selected:

1. An octal address.
2. An asterisk (*), which indicates that the address stored in 1NTRYJA (location 60_8) will be the starting address. 1NTRYJA normally contains the address of a location in the current object program to which control is to be transferred from SYS. (Refer to page 3-7.1.1.)

Note: If *jmp* is omitted, the starting address will be that address indicated by *start*.

addr (OPTIONAL). An octal number which is added to the *start* parameter to designate the normal loading address of the new ABS program. The number is also added to the address specified by the *jmp* parameter. The addition of the values takes place modulo machine size. The ABS program will still be written beginning at the address specified by *start.*

locsent (OPTIONAL). Eight or fewer alphanumeric characters (excluding break characters and spaces) which designate the first 120 words of a sentinel block to be located. If *locsent* is omitted, a sentinel block of Z's is assumed.

wrtsent (OPTIONAL). Eight or fewer alphanumeric characters (excluding break characters and spaces) which designate the first 120 words of a sentinel block to be written following the new ABS program. If *wrtsent* is omitted, a sentinel block of Z's is assumed.

Note: If less than eight characters are specified for either *locsent* or *wrtsent,* the sentinel word will be right-justified and left-filled with zeroes. Spaces are ignored.

PRESET REQUIREMENTS

The preset requirements for WRTABS are the same as that for WRTRPL (page 3-6.8), except that no sentinel search will be made if tape *t* is 5. In this case, the block is written wherever the tape is positioned.

ACTION

When the WRTABS control instruction is initiated, the following action* takes place:

- ** A sentinel search may or may not take place according to Preset Requirements (described above).
- ** If a designated sentinel is not located, the sentinel search will continue until a non-recoverable tape error is encountered or the program is halted by an operator.
- ** If the designated sentinel is found, the tape will be positioned at the beginning of the sentinel before writing takes place.
- If no sentinel search is designated, writing will take place wherever the tape is positioned.
- All information between memory locations *start* and *end* will be written onto Tape *t* in TUG absolute binary card (ABS) format. If Z is written as the *z* parameter, words of zeroes will also be included in the output.
- Two sentinel blocks, designated by *wrtsent,* are written on the output tape following the new ABS program.
- Tape *t* is backspaced two blocks.
- ID *id* ABS CDS *n* ON TAPE *t* is typed as two lines on the Console Typewriter, where *id* is the program identity, *n* is the decimal number of cards of output, and *t* is the tape on which the ABS program is written.

*The action designated by** will not take place if *t* is 5.

REMARKS

1. The first card of the output deck written onto Tape *t* is a dummy PMAX card. The other cards contain the first four characters of the program identification parameter, *id,* in Columns 1-4, and a sequence number in Columns 5-8, in Hollerith code.
2. If *t* is 5, the cards will be edited for normal binary punching in Data Select 1.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	ID *id* ABS CDS *n* ON TAPE *t*	An ABS program designated *id,* containing *n* decimal cards, has been written onto Tape *t.*

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	CONT. LINE ERR.........	One of the following errors was detected in the parameters of a WRTABS control instruction: • A *t, id, start,* or *end* parameter was not given. • An illegal tape number (1) was given. • Too many characters appear in *id.* • The parameter *start* is less than 1000_8. • The parameter *end* is less than *start.* • The address in *start, end, jmp,* or *addr* is not an octal value. • The address in *start, end, jmp,* or *addr* is larger than memory size. Control has been transferred 1XCONER which caused the Console Typewriter to type the above statement and a post-mortem dump to be given. (Refer to pages 4-18 and 4-4.)

EXAMPLE ONE

This example illustrates the use of WRTABS without an offset address.

```
L  Location  Command  Address and Remarks

                      WRTABS   12,JACKSON,2000,3000$
```

Explanation: Execution of this instruction causes the contents of all memory locations from 2000_8 to 3000_8, except words of zeroes, to be written in TUG absolute binary format onto Tape 12. The program identity of the new ABS program is JACKSON. The address specified by *start,* 2000_8, is to be used as the starting address for the ABS program, as the *jmp* parameter is omitted. No *locsent* parameter is specified, and assuming that IBIT 0 had been set to one, a sentinel block of Z's is to be located prior to writing the RPL. No *wrtsent* parameter is given, so two sentinel blocks of Z's are written following the new RPL.

EXAMPLE TWO

This example illustrates the use of WRTABS with an offset address.

```
L Location Command Address and Remarks

Z          WRTABS  15,ABLE,1000,5000,2000,600,BETA,ZETA$
```

Explanation: Execution of this instruction causes the contents of all memory locations from 1000_8 to 5000_8, including words of zeroes, to be written in TUG absolute binary format onto Tape 15. The program identity of the new ABS program is ABLE. The address specified by *jmp*, plus the offset value specified by *addr* indicates that the starting address of the new ABS program is to be 2600_8. The value within *addr* is also added to the *start* address of each card written by WRTABS onto the output tape. Assuming that IBIT 0 was zero prior to the execution of the WRTABS instruction, Tape 15 is first searched for a sentinel block containing 0000BETA in its first 120 words prior to writing the ABS program. Two sentinel blocks, each containing 0000ZETA in their first 120 words, are written following the new ABS program on Tape 15. (If IBIT 0 had been set to one in this example, the search would have been made for a sentinel of Z's — regardless of the configuration within *locsent.*)

EXAMPLE THREE

This example illustrates the use of WRTABS with Tape 5 as the output tape.

```
L Location Command Address and Remarks

Z          WRTABS  5,GOODΔCHEER,6000,12000$
```

Explanation: Execution of this instruction causes the contents of all memory locations from 6000_8 to 12000_8, including words of zeroes, to be written in TUG absolute binary format onto Tape 5. The program identity of the new ABS program is GOODΔCHEER. The address specified by *start*, 6000_8, is to be used as the starting address for the ABS program, as the *jmp* parameter is omitted. Neither *locsent* nor *wrtsent* is specified, as no sentinels are permitted on Tape 5. If they were specified, they would be ignored. Tape 5 is the normal output tape for the System. WRTABS transfers binary card images to this tape for off-line punching in Data Select 1, Image Mode (20 words per card, six cards per block), with the Control Character SENSE/IGNORE Switch of the Punched Card Controller set to IGNORE.

DEBUGGING FUNCTIONS

Various control instructions within the Philco Operating System aid the programmer during the debugging of his programs. DUMP and SNAP instructions provide two methods of obtaining information stored within various parts of memory via the High-Speed Printer. Three other instructions, OCT, CMD, and ALPHA, permit replacement of a word in memory in octal, quaternary/octal, and alphanumeric representation, respectively.

POST-MORTEM DUMP

The post-mortem dumping process of the DUMP control instruction takes place in three stages. The first, or pre-dump stage, which takes place whenever a DUMP control instruction is encountered by SYS, stores information concerning the dump request within a dump table in memory.

The second, or transfer stage, copies binary data from the memory locations specified by the dump table onto Tape 2. The third, or conversion phase, automatically edits the data into the specified format and transfers the edited information to Tape 5.

The latter two stages will take place <u>only</u> if a post-mortem dump is initiated. (Refer to Location Zero, 1ERRDMP, and 1SUBERR, pages 4-2, 4-4, and 4-5.)

SNAP DUMP

The dumping process for the SNAP control instruction differs from that of the DUMP instruction only in that unedited data from selected locations is transferred to Tape 2 during the running of the program rather than after the initiation of a post-mortem dump. If no post-mortem dump takes place, the unedited data is neither edited nor transferred to Tape 5.

CATASTROPHIC DUMP

In the event SYS is destroyed in memory, the operator may call the SYS catastrophic dump (CLOBDMP), which provides a post-mortem dump and recovers any snapshots that may have been prestored on Tape 2.

In addition to the standard first four output lines of a post-mortem dump (page 3-7.1.1), the catastrophic dump provides dumps of locations 0 through 777_8 in Command, Octal and Alphanumeric formats, followed by dumps of locations 1000_8 through 77777_8 in the same three formats.

For a description of catastrophic dump operating procedures, refer to page E-4.

DUMP — Dump Memory

FUNCTION

Prepares SYS to write information from (1) specified areas of memory, (2) all addressable registers, and (3) settings of console switches onto magnetic tape in a designated format edited for the High-Speed Printer. The writing takes place only in conjunction with a post-mortem dump, such as a programmed or manually-executed transfer of control to 1ERRDMP (page 4-4).

FORMAT

L	Location	Command	Address and Remarks
		DUMP	*format,start,end*

PARAMETERS

format An alphabetic character that indicates the format in which information within the specified memory locations is to be dumped. The format must be selected from one of the following:

A	Alphanumeric conversion
C	Command conversion
F	Floating-point conversion
H	Hexadecimal conversion
O	Octal conversion
S	Fixed-point conversion

start An octal number which indicates the starting location of the memory area to be dumped. Non-octal locations are not accepted. If they are used, Error Type-Out 1 will occur, and the DUMP instruction containing the error will be ignored.

end An octal number which indicates the last location of memory to be dumped.

Note: If any parameter is omitted, or if *end* is less than *start*, Error Type-Out 1 (page 3-7.1.4) will occur and the DUMP instruction containing the error will be ignored.

ACTION

For a description of the action which takes place when a DUMP control instruction is executed, refer to the introduction to Debugging (page 3-7) and to the dumping process described in Section II of the manual (page 2-3.1).

REMARKS

1. A post-mortem dump occurs whenever control is transferred to 1ERRDMP. If no DUMP control instruction was given prior to this transfer of control, data in all addressable registers and console switches and in memory locations 60_8 through 75_8 is edited for the High-Speed Printer (refer to Output Format, below).

2. DUMP control instructions should normally appear prior to the instruction which initiates the running of a program. Information specified by the parameters is saved by SYS until a new JOB instruction is encountered.

3. A maximum of ten DUMP instructions may be specified with each JOB instruction. If more than ten are requested, Error Type-Out 2 (page 3-7.1.4) will occur and any DUMP instructions over ten will be ignored.

OUTPUT FORMAT

Every dump output, as converted and edited by DUMPCON (a SYS routine which is called automatically during the dumping process), includes a minimum of seven lines of data on the High-Speed Printer. Their contents are as follows:

Line 1

Includes the word JOB, the program ID, the word DUMPS, the present version of SYS being used, and, if an Accounting Clock is present, the time and date as specified by the CLOCK control instruction (page 3-8.4).

Line 2

Represents the A, Q, D and Toggle Register contents in octal format.

Line 3

Represents Index Registers 0 through 7 in octal format, with the letter C added for each register which is set to count. In the 212 ... a letter Y is added for each register which has its Y bit set to one. either or both bits are zero spaces will be substituted.

Line 4

Represents the JA Register in octal format (including an L or R to indicate left or right), the Overflow and ICO Indicators as ON or OFF, the Breakpoint Indicator as HLT/JMP or IGNORE, and the IBIT word (61_8) in octal format.

Lines 5, 6, and 7

Represent the alphanumeric conversion of data contained within memory locations 60_8 through 75_8, used by SYS in the following manner:

Location 60_8 – 1NTRYJA. A single memory word, the left half of which contains a jump instruction and an address. The address should be either a location in an object program to which control is to be transferred from SYS, or the beginning address for a "JMP *" instruction.

Location 61_8 – 1IBIT1. A single memory word which is set by the IBIT control instruction and may be interrogated by a program (page 3-8.1).

MARCH 1964

CONVERSION	WORDS PER LINE	SPACES BETWEEN WORDS	EXAMPLES	COMMENTS
A:Alphanumeric	8	6	(X2)=Y ? 00??00?0	Question marks denote replacement of Octal 32 or 77. A question mark placed two spaces to the left of a configuration indicates that a replacement has occurred.
C:Command	4	3	RPTNA 00777 ;AMS 0001,1 TIJL 0012,2;TJMR 02531 3333 7777,7;3333 00000	Commands are printed in their mnemonic form; illegal commands are printed in quaternary form. Two repeat options are always given; X indicates an illegal 01 pattern. Addresses are given in octal; a comma indicates an S-bit of one; the digit following the comma is the index register designation.
F:Floating-Point	6	2	-.1000000000 E001 .9999999999E-002 .3750000000 E000 -.3125000000E-001	E indicates times 10 to the power. All numbers are decimal conversions except the following which are converted as octal: -2^{2047}, unnormalized numbers, all numbers less than 5 times -2^{2047} in magnitude. No sign indicates a positive number.
H:Hexadecimal	8	2	7FFF80000000 80007FFFFFFF	A through F denotes binary 1010 through 1111, respectively.
O:Octal	6	2	37777600 00000000 40000177 77777777	Octal representation is given. Half-words are separated by a space.
S:Fixed-Point	6	2	.000700000000000 -.500000000000000	Numbers are given in decimal representation. The word is assumed to be a fraction with the binary point at B0. The binary number -1 appears in octal, as 40000000 00000000. No sign indicates a positive number.

Dump Conversion Format

```
                 JOB      S2101,RSH,ALTAC3                                                   DUMPS    SYSDX60P 10-03 09-59.5
```

CONSOLE DUMP

```
A-REG  00000003 00001040        Q-REG  77777777 00000177        D-REG  00253040 00252640        TOGGLES  00000000 00000000
0X  00000       1X  00310       2X  00806       3X  00065C      4X  75605       5X  00000       6X  00021       7X  00800
JA  00526L            OVERFLOW  OFF             ICO  OFF              BREAKPOINT  HLT/JMP                 IBIT  00000000 00000000
00060        0EH-0≡●-       00000000       00000000       0%0.031;                                        JMP            2
00070
```

ALPHANUMERIC

```
77726        *00A"¬0-       ";Y-">Q-       (1H0,4F6       .1)            ",(-0003       00000000       0010"(D0       00 00000
77736        ":-00000     ? "?+00000       ":-00000       00400000       00+00000       00D00000       0020">-0     ? "?+00000
77746        00+00000       00D00000       HOIOI004       FT;T;T06       C#IOI007       D;T;T;04       FIOIOI04     ? ?;T;T;04
77756        D 000006       +6IOI005       A(T;T≡07       C;T;T≡05       +T;T;T02       FC;T;T05       HOIOI004       D;T;T;04
77766        D 000006       C;T;T≡05       FT;T;T06       FIOIOI04       +6IOI005       +T;T;T02       C#IOI007     ? ?;T;T;04
77776        A(T;T≡07       FC;T;T05
```

COMMAND

```
77726    TMA    0000,3;JMPL   77240     JMPL   76316 ;JMPL   77732     HLTL   0026,7;SKF    6105,6     TDXL   66027 ;TDXL   1414,4
77732    JMPL   77677 ;NOPL   00000     HLTL   00000 ;HLTL   00000     AM     00000 ;HLTL   77705      HLTL   00014 ;HLTL   00000
77736    HLTL   77750 ;HLTL   00000     HLTL   77764 ;HLTL   00000     HLTL   77750 ;HLTL   00000      HLTL   00001 ;HLTL   00000
77742    HLTL   00004 ;HLTL   00000     HLTL   00005 ;HLTL   00000     HLTR   00000 ;HLTL   77730      HLTL   77764 ;HLTL   00000
77746    HLTL   00004 ;HLTL   00000     HLTL   00005 ;HLTL   00000     MMAR   61146 ;TIO    63140      TXDLC  55463 ;SKC    31460
77752    MMAR   47346 ;TCXZ   63140     FAQA   50314 ;TIO    6300,4    SRQ    54631 ;TIO    4620,1     FAQA   64314 ;TIO    6300,4
77756    HLTL   51400 ;SKC    00000     MMAR   40146 ;TCM    63140     FAQA   43714 ;TCXZ   6320,4     FAQA   46314 ;TCM    6320,4
77762    TXDLC  41463 ;ICOZ   31460     TXDLC  54463 ;TCM    31460     MMAR   61146 ;TIO    63140      FAQA   50314 ;TIO    6300,4
77766    HLTL   51400 ;SKC    00000     FAQA   46314 ;TCM    6320,4    TXDLC  55463 ;SKC    31460      SRQ    54631 ;TIO    4620,1
77772    MMAR   40146 ;TCM    63140     TXDLC  41463 ;ICOZ   31460     MMAR   47346 ;TCXZ   63140      FAQA   64314 ;TIO    6300,4
77776    FAQA   43714 ;TCXZ   6320,4    TXDLC  54463 ;TCM    31460
```

FLOATING-POINT

```
77726  -.2684345602 E010   .1737437435E-452   74013000 73042606   .1322215186E-293   .7984161377 E001   00000000 00000000
77734  00000100 37742400   00006000 00000000   .9992675781 E000   .9996337890 E000   .9992675781 E000   00000400 00000000
77742  00002000 00000000   00002400 00000000   00000200 37754000   .9996337890 E000   00002000 00000000   00002400 00000000
77750   .1230000000 E002   .4560000000 E002   .7889999999 E002   .1010000000 E002   .1120000000 E002   .1310000000 E002
77756   .4150000000 E002   .1610000000 E002   .7180000000 E002   .1920000000 E002   .2100000000 E001   .2230000000 E002
77764   .1230000000 E002   -.1010000000 E002  .4150000000 E002   .1920000000 E002   .4560000000 E002   .1120000000 E002
77772   .1610000000 E002   .2100000000 E001   .7889999999 E002   .1310000000 E002   .7180000000 E002   .2230000000 E002
```

HEXADECIMAL

```
77726  B000117EA020   7CCE207FDA20   F01600EC4586   6C1730C30C30   7FBF20000003   000000000000   0000407FC500   000C00000000
77736  7FE800000000   7FF400000000   7FE800000000   000100000000   000400000000   000500000000   0000807FD800   7FF400000000
77746  000400000000   000500000000   626666666004   5B3333333006   4EE666666007   50CCCCCCC004   599999999004   68CCCCCCC004
77756  530000000006   406666666005   47CCCCCCD007   4CCCCCCCD005   433333333002   593333333005   626666666004   50CCCCCCC004
77766  530000000006   4CCCCCCCD005   5B3333333006   599999999004   406666666005   433333333002   4EE666666007   68CCCCCCC004
77776  47CCCCCCD007   593333333005
```

OCTAL

```
77726  54000021 37520040   37147040 37755040   74013000 73042606   33013460 60606060   37737440 00000003   00000000 00000000
77734  00000100 37742400   00006000 00000000   37764000 00000000   37772000 00000000   37764000 00000000   00000400 00000000
77742  00002000 00000000   00002400 00000000   00000200 37754000   37772000 00000000   00002000 00000000   00002400 00000000
77750  30463146 31460004   26631463 14630006   23563146 31460007   24146314 63140004   26314631 46310004   32146314 63140004
77756  24600000 00000006   20063146 31460005   21746314 63150007   23146314 63150005   20631463 14630002   26231463 14630005
77764  30463146 31460004   24146314 63140004   24600000 00000006   23146314 63150005   26631463 14630006   26314631 46310004
77772  20063146 31460005   20631463 14630002   23563146 31460007   32146314 63140004   21746314 63150007   26231463 14630005
```

FIXED-POINT

```
77726   .624997914477489    .975040495326766    .124328503258723    .844457717168893    .998020172119161    .000000000000000
77734   .000007688891855    .000366210937500    .999267578125000    .999633789062500    .999267578125000    .000030517578125
77742   .000122070312500    .000152587890625    .000015318320947    .999633789062500    .000122070312500    .000152587890625
77750   .768749999988386    .712499999994221    .616406249988408    .631249999976745    .699999999982566    .818749999976745
77756   .648437500000042    .503124999988393    .560937500005870    .600000000005856    .524999999994193    .696874999994214
77764   .768749999988386    .631249999976745    .648437500000042    .600000000005856    .712499999994221    .699999999982566
77772   .503124999988393    .524999999994193    .616406249988408    .818749999976745    .560937500005870    .696874999994214

   END OF DUMPS
```

Output From a Post-Mortem Dump

Location 62_8 — 1IBIT2. A single memory word which serves as a pseudo Toggle Register for SYS use only.

Location 63_8 — 1CONBIT. A 48-bit word which is used predominantly by SYS. (Refer to 1NXTCRD, page 4-19.)

Locations 64_8 - 75_8 — 1CONLIN. The last control instruction executed by SYS or the last instruction delivered to a program using the 1NXTCRD entry (page 4-19). (Under some circumstances, SYS uses these locations for temporary storage of other data.)

Memory Dump Lines

Each printed line in a memory dump, similar in format to Lines 5, 6, and 7 described previously, begins with the five-digit octal address of the first word represented on the line. An asterisk to the right of this address indicates the deletion of a string of words equal to the last word of the preceding line.

The table on page 3-7.1.2 outlines the six types of memory dump conversions. The figure on page 3-7.1.3 depicts the six types of dump conversions within a post-mortem dump output.

EXAMPLE

L	Location	Command	Address and Remarks
		DUMP	O,1000,1700

Explanation: Execution of this instruction causes the dump routine to be preset to execute a dump from locations 1000_8 through 1700_8 in octal format.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	DUMP ERR	An illegal dump parameter was encountered by SYS. The dump card will be ignored.
2	DUMP TABLE FULL	More than ten DUMP instructions were given within one job. All dump calls over ten are ignored.

SNAP — Snapshot of Memory

FUNCTION

Prepares SYS to write information from (1) selective locations in memory, (2) all addressable registers, and (3) settings of console switches* onto magnetic tape, whenever designated conditions occur during the running of a program. The information is converted to a designated format and edited for the High-Speed Printer only when a post-mortem dump occurs, such as a programmed or manually-executed transfer of control to 1ERRDMP (page 4-4).

FORMAT

L	Location	Command	Address and Remarks
p	*addr*	SNAP	*format;start;end;table;cond*

PARAMETERS

SNAP parameters are written as follows: *p* appears in the Label column, *addr* appears in the Location field, and the remainder appears in the Address and Remarks field, separated by semicolons.

p An alphabetic character, L or R, which indicates whether the left or right instruction of a particular location may initiate the snap routine. The snap occurs prior to the execution of the instruction. If nothing is written in the Label column, L will be assumed.

addr An octal address of an instruction in the running program, which, if encountered, may initiate the snap routine. If nothing is written in the Location field, Error Type-Out 1 (page 3-7.2.5) will occur, and the next instruction will be executed.

format An alphabetic character (except E or N) which indicates the format in which information within the specified memory locations is to be dumped. E and N specify a return to SYS. One of the following eight selections must be designated:

A	Alphanumeric conversion
C	Command conversion
F	Floating-point conversion
H	Hexadecimal conversion

* Toggle and Breakpoint switch values reflect the time the post-mortem dump is executed, not each time a SNAP instruction dump is executed.

O	Octal conversion
S	Fixed-point conversion
E	Specifies transfer of control to SYS to perform a post-mortem dump and a search for the next job.
N	Specifies transfer of control to SYS to execute the next SYS control instruction.

If *format* is either E or N, no additional parameters are recognized. If an illegal character or nothing is specified as *format,* Error Type-Out 1 (page 3-7.2.5) will occur, and the next control instruction will be executed.

start An octal address which indicates the first memory location to be dumped. If an index register is specified, it must be separated from the address by a comma, and the character X may not be used to designate the register. For example, 2347,1 is acceptable; 2347,1X is not. Eight index registers are assumed; index register designations 8 and 0 are considered to be the same.

end An octal address which indicates the last memory location to be dumped. The same rules for *start* also apply to *end.*

table (OPTIONAL). An octal number which indicates the location of a snap table. The snap table consists of six words which contain information required to perform the snapshot dump, the call for the snapshot routine, the instruction whose appearance initiates the snap routine, and an exit to the object program. The table is stored in the six memory locations immediately preceding the address specified in *table.* If no address is specified, the snap table will be stored in memory immediately preceding the last snap table. If no address had been specified for any snap table, the first table will be stored at the end of memory. It is the programmer's responsibility to select the appropriate locations and provide sufficient memory space for all his snapshot tables.

cond (OPTIONAL). Specifies a condition which must be satisfied before a snapshot dump can be executed. Any one of the following conditions may be designated:

- (Condition 1). The contents of either the A Register, the Q Register, or a memory location are zero, non-zero, positive, or negative.
- (Condition 2). The contents of an index register are either equal to, not equal to, greater than, or less than a specified value.
- (Condition 3). The Mth to the Nth time the specified instruction is to be executed.
- (Unconditional Snap). The snapshot dump is to take place every time the particular instruction is to be executed.

CONDITION 1—To set the condition based upon the status of the A or Q Register or memory location, the *cond* parameter must be selected from one of the following 12 combinations:

A=0	Q=0	*m*=0
A>0	Q>0	*m*>0
A<0	Q<0	*m*<0
A#0	Q#0	*m*#0

Where	A	is the A Register
	Q	is the Q Register
	m	is an octal memory location which may be index register-modified
	=0	is zero
	>0	is positive
	<0	is negative
	#0	is non-zero

Examples:

A=0	(If the contents of the A Register are zero)
12345>0	(If the contents of octal address 12345 are positive)
0,5#0	(If the contents of index register-modified address 0,5 are non-zero)

CONDITION 2—To set the condition based upon the status of an index register, the *cond* parameter must be selected from one of the following four combinations:

*n*X=, *v*   *n*X>, *v*   *n*X<, *v*   *n*X#, *v*

Where	*n*X	is an index register, *n* being a number from 0 through 7; 8 is assumed to be 0
	v	is an octal value

Examples:

7X=,3456	(If the contents of Index Register 7 are equal to the octal value 3456)
0X<,1573	(If the contents of Index Register 0 are less than the octal value 1573)

CONDITION 3—To set the condition based on the Mth to Nth time an instruction is to be executed, any combination of the following may be written:

m/n	Where *m* is the Mth time, and *n* is the Nth time, expressed by decimal numbers

Example:

10/91	(If it is the 10th time through the 91st time this instruction is to be executed)

UNCONDITIONAL SNAP—If the *cond* parameter is omitted, a snap dump will take place each time the instruction is to be executed.

ACTION

For a description of the action which takes place when a SNAP control instruction is executed, refer to the introdution to Debugging (page 3-7) and to the dumping process described in Section II of the manual (page 2-3.1).

REMARKS

1. The program to be run must be loaded into memory prior to the appearance of the SNAP control instruction(s). (SNAP calls may be compiled with a program through the use of the SNAPGEN SYS generator, described on page 6-3.)
2. If the same address is specified in the Location fields of two or more SNAP instructions, the SNAP instructions which specify that address are performed in the reverse order from which they were submitted. SNAP instructions which specify different addresses need not be sequenced by address.
3. When control is returned from the snapshot call to the object program, all registers except JA are restored.
4. There is no limit to the number of SNAP instructions which can be given with each job, except that sufficient memory space should be available for snap tables (six locations per call).
5. The instruction specified in the Location field of a SNAP instruction may not be a TIJ, TJM, TIO, SKC, or RPT. In addition, this instruction may not be under the influence of an RPT, TIO, or SKC instruction, nor may it be program-modified (e.g., by EIS or TJM). If one of these conditions is in effect, (1) the return from the snapshot call may be in the wrong location in the object program, (2) a TJM may give unexpected results, or (3) control may be transferred to Location Zero.
6. When a return to SYS is specified by E or an N as the first parameter in the Address and Remarks field, the instruction at the location specified by the Label and Location fields is not performed.
7. To obtain an edited SNAP dump on Tape 5, (1) control must be transferred to 1ERRDMP, or (2) a SNAP instruction with an E as the first parameter must be executed.

OUTPUT FORMAT

Every snap output, as converted and edited by DUMPCON (a SYS routine which is called automatically during the dumping process), includes a minimum of five lines of data on the High-Speed Printer. Their contents are as follows:

Line 1

Includes the word JOB, the program ID, the word DUMPS, the present version of SYS being used, and, if an Accounting Clock is present, the time and date as specified by the CLOCK control instruction (page 3-8.4).

Line 2

Represents the A, Q, D and Toggle Register settings in octal format.

Line 3

Represents Index Registers 0 through 7 in octal format, with the letter C added for each register which is set to count.

Line 4

Represents the JA Register in octal format (including an L or R to indicate left or right), the Overflow and ICO Indicators as ON or OFF, the Breakpoint Indicator as HLT/JMP or IGNORE, and the IBIT word (61_8) in octal format.

Remaining Lines

Line 5 and any additional lines contain the requested information in the specified format beginning with a five-digit octal address of the first word represented on the line. The table on page 3-7.1.2 outlines the six types of memory dump conversions.

NOTE: Only the first SNAP dump of an over-all job to be printed on the High-Speed Printer carries Line 1 of SNAP output, while all others contain Lines 2 through 5 and any additional lines which may be required.

EXAMPLE ONE

This example illustrates the use of a conditional snap dump.

```
L  Location   Command   Address and Remarks

              JOB       XYZ
              RPL       4,ALPHA
L  1526       SNAP      O;0,3;12,3;;A#0
              JMP       *
```

Explanation: Execution of these instructions causes program ALPHA to be loaded and run. An octal dump of locations 0,3 through 12,3 will be performed preceding the instruction at 1526L, if the contents of the A Register are non-zero. The SNAP table is situated in the last six memory locations at the end of memory.

EXAMPLE TWO

This example illustrates the use of an unconditional snap dump.

L	Location	Command	Address and Remarks
	1443	SNAP	F;2405;2417

Explanation: Execution of this instruction causes an unconditional floating-point dump of locations 2405_8 through 2417_8 to be performed preceding the instruction at 1443L (L is assumed).

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	SNAP ERR. AT = *p* *addr*	Illegal parameter(s) are contained in a SNAP control instruction, where *p* is the label and *addr* is the octal address of the instruction which initiated the snap routine. The SNAP call containing the illegal parameter is ignored.

OCT — Replace Word(s) in Octal Format

FUNCTION

Replaces the contents of one or more consecutive memory locations with words in octal format.

FORMAT

L	Location	Command	Address and Remarks
	addr x	OCT	$word_1, \ldots, word_n$

PARAMETERS

addr An octal number which indicates the location of the first word to be changed in a program.

x (OPTIONAL). An alphabetic character, P or C, which indicates that *addr* is relative to the program origin or Common.

P Indicates that the effective address will be the sum of *addr* and the origin of the last REL program loaded.

C Indicates that the effective address will be the sum of *addr* and the first location of Common.

$word_1$ 16 or fewer octal digits which indicate the bit configuration to be inserted into the first location to be changed. If fewer than 16 characters are specified, the remainder of the word is right-filled with zeros. If more than 16 characters or a non-octal configuration is given, a Control Line Error will result.

$,word_n$ (OPTIONAL). One or more additional parameters in the same format as $word_1$ to be inserted into subsequent consecutive locations.

REMARKS

1. The number of locations which can be corrected by one OCT instruction is limited to the number of consecutive octal configurations which can be contained in the Address and Remarks field of a card.

2. If corrections are to be made to relocatable programs, the following information should be noted:

   The relocatable loader is stored in part of Common (1000_8 to 2000_8). The symbol list may be stored in the TEMPORARY/ASTOR area of the program, but if there is symbol overflow, an additional part of Common is also used for storage. For these reasons, corrections to the Common area should be made only when the loader, and possibly symbols, need not be saved. Corrections to the TEMPORARY/ASTOR area should be made only when symbols need not be saved.

EXAMPLE

```
L  Location  Command  Address and Remarks

   100C      OCT      1111111111111111,22222,333333
   200P      OCT      4444,666666666,555
   1025      OCT      777,44
```

Explanation: Execution of these instructions will replace the contents of the memory locations listed below with the following information (expressed in TAC format):

Location	New Contents
CSA of Common + 100_8	16/001
CSA of Common + 101_8	5/010
CSA of Common + 102_8	6/011
Program Origin + 200_8	4/100
Program Origin + 201_8	9/110
Program Origin + 202_8	3/101
1025_8	3/111
1026_8	2/100

CMD (or COMMAND) — Replace Program Instruction(s) in Command Format

FUNCTION

Replaces the contents of one or more consecutive half-words in memory with instructions in command format.

FORMAT

L	Location	Command	Address and Remarks
p	*addr x*	CMD	*instr*$_1$,*addr*$_1$*x*...;*instr*$_n$,*addr*$_n$*x*

PARAMETERS

p Position of the half-word within a memory location which indicates whether the left or right instruction is the first to be changed. The *p* is to be written in the Label column and may be one of the following:

L (or blank)	Specifies the left instruction
R	Specifies the right instruction

addr An octal address written in the Location field which indicates the location of the first word to be corrected in a program.

x (OPTIONAL). An alphabetic character, P or C, which indicates that *addr* is relative to the program origin or Common.

P	Indicates that the effective address will be the sum of *addr* and the origin of the last REL program loaded.
C	Indicates that the effective address will be the sum of *addr* and the first location of Common.

instr$_1$,*addr*$_1$*x* The new instruction and its address, separated by a comma, to be inserted into the first half-word in memory to be changed.

instr$_1$	A quaternary representation of the Philco 2000 command to be inserted.
addr$_1$*x*	The octal representation of the address field of the new command, where x is an optional C or P for relative corrections as described above.

;*instr*$_n$,*addr*$_n$*x* (OPTIONAL). One or more additional parameters in the same format as that for instr$_1$,addr$_1$, to be inserted into subsequent consecutive half-words in memory. Parameters within the CMD control line must be separated by semicolons.

ACTION

When the CMD control instruction is executed, one or more half-words in memory, starting in the *p* half of *addr,* are replaced by one or more half-words designated by *instr*$_1$,*addr*$_1$;...*instr*$_n$, *addr*$_n$.

REMARKS

1. CMD and COMMAND may be used interchangeably as the call for this control instruction. CMD is preferred.
2. If either *instr* and/or *addr* is omitted from a parameter, the corresponding field(s) of the half-word will be filled with zeroes.
3. The latter portion of the parameter, *addr,* may contain an index register-modified address, if the address and the single digit (0 through 7) designating the index register are separated by a comma. Thus the entire parameter may contain two commas, one between the instruction and address and the second between the address and the index register number.
4. The number of instructions which can be replaced by one CMD instruction is limited to the number of consecutive half-words which can be contained within the Address and Remarks field of a card.
5. Only quaternary characters (for the instruction) and octal digits (for its address) may be used in the Address and Remarks field; otherwise, a Control Line Error (page 4-18) will occur.
6. Refer to OCT (page 3-7.3), REMARK 2, concerning relocatable corrections.

EXAMPLE ONE

This example illustrates the use of an absolute address correction.

L	Location	Command	Address and Remarks
R	2120	CMD	0300,0,5;0100,17677

Explanation: Execution of this instruction causes two half-words contained in memory locations 2120R and 2121L, to be replaced by the binary representation for TDXL 0,5 and CM 17677, respectively.

EXAMPLE TWO

This example illustrates the use of a relative address correction.

L	Location	Command	Address and Remarks
		REL	4,CORRECT$
R	210P	CMD	0202,14P;0102,30C

Explanation: Execution of the first instruction above causes program CORRECT to be loaded without subroutines. If SUBS were specified, the origin used for the corrections would be that of the last subroutine loaded rather than that of the program CORRECT. Execution of the second instruction causes the two command corrections to be made, starting at the right half of the word at 210_8 relative to the program origin. The first command operand address is relative to the program origin; the second to Common.

The results of the CMD instructions (expressed in TAC format) will be:

Location	New Contents
Program Origin + 210_8 (right half)	JNOL Program Origin $+14_8$
Program Origin + 211_8 (left half)	TMQ CSA of Common $+30_8$

ALPHA — Replace Word(s) in Alphanumeric Format

FUNCTION

Replaces the contents of one or more consecutive memory locations with words in alphanumeric format.

FORMAT

L	Location	Command	Address and Remarks
n	*addr x*	ALPHA	*replacement*

PARAMETERS

n A decimal number, one through seven, which specifies the number of words to be replaced. If the parameter is omitted, the replacement of one word is assumed. A value equal to zero or greater than seven will result in a Control Line Error (page 4-18).

addr An octal number which indicates the location of the first word to be changed in a program.

x (OPTIONAL). An alphabetic character, P or C, which indicates that *addr* is relative to the program origin or Common.

P Indicates that the effective address will be the sum of *addr* and the origin of the last REL program loaded.

C Indicates that the effective address will be the sum of *addr* and the first location of Common.

replacement Seven or fewer groups of eight alphanumeric characters each, which replace the current contents of the specified consecutive memory locations. Spaces are significant. All Philco characters are permitted (refer to Appendix B).

ACTION

When the ALPHA control instruction is executed, *n* number of words in memory, starting at location *addr,* are replaced by *n* eight-character configurations designated by *replacement.*

REMARKS

Refer to REMARK 2 in OCT (page 3-7.3) concerning relocatable corrections.

EXAMPLE

L	Location	Command	Address and Remarks
3	205P	ALPHA	ONE△△IS$TWO△△IS$THREE$$$
2	1200	ALPHA	ABSOLUTEEXAMPLE$

Explanation: Execution of these instructions causes the contents of the memory locations listed below to be replaced with the following information (expressed in TAC format):

Location	New Contents
Program Origin $+205_8$	W/ONE△△IS$
Program Origin $+206_8$	W/TWO△△IS$
Program Origin $+207_8$	W/THREE$$$
1200_8	W/ABSOLUTE
1201_8	W/EXAMPLE$

SPECIAL FUNCTIONS

The final category of SYS control instructions includes those which perform miscellaneous computer operations. Among these operations are setting a pseudo Toggle Register, searching for a specified job, clearing memory, and halting the computer.

IBIT — Indicator Bit

FUNCTION

Changes the bit configuration in the Indicator Word (memory location 61_8), which serves as a pseudo Toggle Register.

FORMAT

L	Location	Command	Address and Remarks
		IBIT	*n,n,n,...n*

PARAMETERS

n One or more decimal numbers, 0 through 47, which designate the specific bit or bits to be reversed (i. e., to change a zero to a one, or a one to a zero). If any number greater than 47 is specified, a Control Line Error will result.

ACTION AS A TOGGLE REGISTER

- Used in place of a Toggle Register, the IBIT instruction can generate a designated bit configuration in the Indicator Word. The configuration can then be transferred to any register or location for use by the programmer.
- Each number placed in the Address and Remarks field as a parameter reverses its particular bit in the Indicator Word.
- Any number greater than 47 is illegal and results in a Control Line Error (page 4-18).
- The Indicator Word is cleared whenever a new JOB instruction is executed. (Refer to page 3-2.1.)
- Bits in the Indicator Word may be sensed by the ALTAC IF SENSE BIT statement (refer to the Philco ALTAC Manual).

ACTION DURING TAC OR ALTAC COMPILATION

- One use of the control word (location 61_8) as a toggle register is made by SYS during a TAC or ALTAC compilation to alter normal program transfer from one tape to another, or to specify the type of language input.
- TAC compilations which have no errors normally transfer RPL programs from Tape 2 to the program tape (normally Tape 4) and ABS and REL programs from Tape 2 to the program tape and Tape 5. If compilation errors are present, programs may not be transferred to the program tape and/or Tape 5.

To inhibit any of these actions, override various restrictions, or specify language input, the following bit locations within the Indicator Word may be set at the programmer's discretion (bits must be set prior to the execution of the SYS control instruction TAC or ALTAC in the job):

37 A one in IBIT 37 indicates that Philco 212 mnemonic commands will be compiled into their proper bit configuration; otherwise, such commands will be considered command faults.

38 A one in IBIT 38 indicates that POSSIBLE F-BIT ERROR will appear on the Code-Edit whenever a possible F-Bit error is detected during a TAC compilation.

39 A one in IBIT 39 indicates that TAC will *not* store the intermediate code in the higher-order portion of memory on 32.768-word memory machines. Instead, the code will be written onto Tape 3 and the memory space thereby released will be used for the symbol table.

41 A one in IBIT 41 indicates that the input will be in Binary-Image Mode on a magnetic tape other than 8.

42 A one in IBIT 42 indicates that the input will be in Hollerith-Image Mode on a magnetic tape other than 8.

43 A one in IBIT 43 indicates, that the TAC language input will be copied from Tape 8 in the same mode to Tape 6. It also indicates that ALTAC language input will be copied from Tape 8 in the same mode to Tape 2.

44 A one in IBIT 44 indicates that a compiled program will not be transferred to the program tape, even if no serious compilation errors are detected.

45 A one in IBIT 45 indicates that a compiled program will not be transferred to Tape 5 for off-line punching even if no serious compilation errors are detected.

46 A one in IBIT 46 indicates that a compiled program will not be transferred to the program tape or Tape 5, even if no serious compilation errors are detected.

47 A one in IBIT 47 indicates that any or all compiled program transfers will take place, subject to the settings of IBITS 44 and 45 regardless of serious compilation errors.

REMARKS

1. IBITS 46 and 47 take precedence over IBIT settings 44 and 45 during a TAC compilation.

2. Refer to the figure on page 3-3.1.2 for a logic diagram of compilation transfers.

3. IBIT 0 is restricted to system maintenance use for all compiler calls (TAC, ALTAC, COBOL, REPORT)

ACTION DURING COBOL COMPILATION

The COBOL compiler uses the IBIT control word (location 61_8) as a toggle register for special input or output conditions. If any of the following IBIT's are set to one prior to a COBOL compilation, the action indicated will take place.

34 A one in IBIT 34 indicates that special COBOL input format is to be used during compilation (Columns 73-80 are ignored and the 8-4 punch is to be interpreted as a quote ('') character).

35 ~~A one in IBIT 35 indicates that the COBOL compiler will produce as output special information to aid in debugging the compiler.~~ During COBOL compilation, IBIT 35 is used for COBOL maintenance use.

36 A one in IBIT 36 indicates that the TAC phase of the compilation is to be bypassed.

~~40 A one in IBIT 40 indicates that the COBOL library is to be ignored.~~

In addition to these IBIT's, those of TAC (page 3-8.1.0) also apply to COBOL compilations.

USE OF IBIT 0 TO SPECIFY SENTINEL SEARCH

IBIT 0 may be used with the SEGMENT, WRTRPL, and WRTABS control instructions to specify a particular type of sentinel search to be made prior to writing programs in memory onto an output tape. (Refer to pages 3-5.1, 3-6.8, and 3-6.11, respectively.)

EXAMPLE ONE

This example illustrates the use of the IBIT command as a Toggle Register.

L	Location	Command	Address and Remarks
		IBIT	0,1,6,7,12,13,18,19

Explanation: Execution of this instruction causes an octal bit configuration of 6060606000000000 to be inserted into location 61_8 (assuming the word originally contained all zeros).

EXAMPLE TWO

This example illustrates the use of the IBIT command for a TAC compilation.

L	Location	Command	Address and Remarks
		IBIT	45

Explanation: Execution of this instruction causes IBIT 45 to be set to one, assuming it was previously a zero. If a TAC compilation follows, any binary cards produced will not be transferred to Tape 5.

EXAMPLE THREE

This example illustrates that each time a number is specified the bit is reversed.

L	Location	Command	Address and Remarks
		IBIT	27,27

Explanation: Execution of this instruction causes no change to the Indicator Word; each appearance of a number reverses the bit.

JOBSRCH — Search For A Specified Job

FUNCTION

Searches the SYS Input Tape (Tape 8) for a specified JOB card (refer to page 3-2.1) and initiates immediate execution of that job.

FORMAT

L	Location	Command	Address and Remarks
		JOBSRCH	*id,image*

PARAMETERS

id 16 or fewer alphanumeric characters which correspond to the first parameter of the JOB card to be located. Spaces are ignored.

image (OPTIONAL). This parameter, if used, is written as IMAGE, which signifies that SYS input is in Image Mode. This parameter must be used if input is in Image Mode. If omitted, Code Mode is assumed.

ACTION

The JOBSRCH control instruction performs the following actions:

- Tape 8 is searched forward only for the particular JOB card specified by the *id* parameter of the JOBSRCH control instruction.
- When the designated JOB card is encountered, FOUND JOB is typed on the Console Typewriter and the job is executed.
- If an ENDINPUT JOB card (see REQUIREMENTS below) is encountered prior to the designated JOB card, CANNOT FIND JOB *id* will be typed on the Console Typewriter. Control will then be transferred to 1SYSIN (page 4-3), which will request further information from the operator. Any action taken at this point depends upon the operator or the particular computer installation.

REQUIREMENTS

For an installation to use JOBSRCH, input must be read into memory via the SYS input tape (Tape 8). Moreover, the input tape must be terminated by a JOB card with the characters ENDINPUT as the first parameter. (In normal SYS practice, the input tape is terminated by the ENDINPUT JOB card.)

EXAMPLE

```
L  Location  Command    Address and Remarks

             JOBSRCH    PROGRAMJJ, IMAGE
```

Explanation: Execution of this instruction causes the job PROGRAMJJ to be located on the SYS input tape. After the job is located, FOUND JOB is typed on the Console Typewriter. PROGRAMJJ is then immediately executed. If the job is not found, CANNOT FIND JOB PROGRAMJJ will be typed on the Console Typewriter, and control will be transferred to 1SYSIN (page 4-3).

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	FOUND JOB	Indicates that the specified JOB card has been located. The name of the job is then typed on the Console Typewriter and the job is initiated.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	CANNOT FIND JOB *id*	Indicates that a JOB card with ENDINPUT as its first parameter was encountered prior to the requested JOB card specified by *id*. Control is transferred to 1SYSIN.

CLEAR — Clear Memory

FUNCTION

Clears specified memory locations to fixed-point zeroes (48 zero bits per word).

FORMAT

L	Location	Command	Address and Remarks
		CLEAR	*start,end*

PARAMETERS

start A number in octal notation, 1000 or greater, which indicates the beginning location of an area in memory to be cleared. If the value is less than 1000_8, a Control Line Error will result. If *start* is omitted (indicated by a break character in its place), the beginning location will automatically be set to 1000_8.

end A number in octal notation which indicates the last location of an area in memory to be cleared. If the value is greater than memory capacity of the particular computer, a Control Line Error will result. If *end* is omitted, the last location will automatically be set to the end of memory.

Notes:

1. If both parameters are omitted, all memory from location 1000_8 to the end of memory will be cleared.
2. If *start* has a value greater than *end*, or if non-octal characters are specified, a Control Line Error will result.

ACTION

When the CLEAR control instruction is executed, all memory locations between *start* and *end* inclusive are cleared to fixed-point zeroes.

EXAMPLE ONE

This example illustrates the clearing of specified memory locations.

L	Location	Command	Address and Remarks
		CLEAR	1400,2600

Explanation: Execution of this instruction causes the bit configuration in all words from locations 1400_8 to 2600_8 to be replaced by zeroes.

EXAMPLE TWO

This example illustrates the clearing of memory from location 1000_8 to a specified memory location.

L	Location	Command	Address and Remarks
		CLEAR	,6000

Explanation: Execution of this instruction causes the bit configuration in all words from locations 1000_8 to 6000_8 inclusive to be replaced by zeroes.

EXAMPLE THREE

This example illustrates the clearing of all memory.

L	Location	Command	Address and Remarks
		CLEAR	

Explanation: Execution of this instruction causes the bit configuration in all words from location 1000_8 to the end of memory to be replaced by zeroes.

CLOCK — Type Accounting Clock Date and Time on Console Typewriter

FUNCTION

Causes the date and time from the Accounting Clock to be typed on the Console Typewriter.

FORMAT

L	Location	Command	Address and Remarks
		CLOCK	

PARAMETERS

None

ACTION

When the CLOCK control instruction is executed, the time and date as indicated by the Accounting Clock are typed on the Console Typewriter (refer to Normal Type-Out 1, below) in the form MM-DD HH-MM.T. The time and date are stored in 1DATE and 1DATE+1 (215_8 and 216_8).

REMARKS

1. If the CLOCK instruction is given and a computer has no Accounting Clock, the instruction will be ignored.
2. If the Accounting Clock is unavailable when a CLOCK command is given, Error Type-Out 1 (below) will occur and the next control instruction will be executed.
3. Whenever 1DATE and 1DATE+1 are interrogated, the values will reflect the time at which the most recent CLOCK, JOB, TAC, or ALTAC control instruction was executed. If no clock is present in the system, the contents of the two words will be meaningless.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	*date,time*	The CLOCK instruction was encountered and the date and time are typed via the Accounting Clock in the form: MM-DD HH-MM.T, where MM-DD is the month and day, and HH-MM.T is the hour (per 24-hour day), the minute, and tenth of a minute.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	CLOCK FAILURE	The Accounting Clock was interrogated but was unavailable. The CLOCK instruction is ignored.

OUTPUT EXAMPLE

The Console Typewriter type-out, 02-14 15-55.6, indicates that the time on the Accounting Clock when the CLOCK control instruction was encountered was February 14 at 55.6 minutes past 3 p.m.

HLT — Halt Computer Operation

FUNCTION

Relays messages to the operator via the Console Typewriter and stops the computer, to permit a manual operation.

FORMAT

L	Location	Command	Address and Remarks
		HLT	*message*

PARAMETER

message Fifty-six or fewer alphanumeric characters (including spaces) written in the Address and Remarks field, that compose a message to the operator. The characters must be acceptable to the Console Typewriter. Upper case mode is assumed (refer to Philco 2000 Character Codes, Appendix B).

ACTION

When the HLT control instruction is executed, *message* is typed on the Console Typewriter. Upon completion of the type-out, the computer halts, enabling the operator to perform the manual action specified by *message*.

REMARKS

1. If the message is longer than the Address and Remarks field (56 characters), a REM control instruction must also be used. (Refer to page 3-8.6.)

2. After completing any manual operation required, the operator should press the Advance Bar to cause SYS to read and execute the next control instruction.

3. Whenever the computer stops via a HLT instruction, all one's are displayed in the A, Q, and D Registers, indicating to the operator that this is a System halt.

EXAMPLE

L	Location	Command	Address and Remarks
		HLT	PUT REEL 1-1036 ON T9

Explanation: Execution of this instruction causes HLT PUT REEL 1-1036 ON T9 to be typed on the Console Typewriter and computer action to be halted. Operation may be continued by pressing the Advance Bar.

REM — Relay Remarks To An Operator

FUNCTION

Relays messages to the operator via the Console Typewriter without stopping computer action.

FORMAT

L	Location	Command	Address and Remarks
		REM	*message*

PARAMETER

message Fifty-six or fewer alphanumeric characters (including spaces) written in the Address and Remarks field, that compose a message to the operator. The characters must be acceptable to the Console Typewriter. Upper case mode is assumed (refer to Philco 2000 Character Codes, Appendix B).

ACTION

When the REM control instruction is executed, *message* is typed on the Console Typewriter without stopping computer action.

REMARKS

1. If the message is longer than 56 characters, as many REM commands may be used as necessary.
2. The REM instruction may also be used in conjunction with the HLT command, if the HLT message is longer than 56 characters. As many REM commands may be used as necessary. The final portion of the message, however, must appear with the HLT instruction (refer to HLT, page 3-8.5).

EXAMPLE

L	Location	Command	Address and Remarks
		REM	THIS IS A TEST PROGRAM

Explanation: Execution of this instruction causes REM THIS IS A TEST PROGRAM to be typed on the Console Typewriter without halting computer action.

JMP — Transfer Control To A Memory Location

FUNCTION

Transfers control to a specified address.

FORMAT

L	Location	Command	Address and Remarks
		jmp	*addr*

PARAMETERS

jmp A JMP command which indicates the portion of the word to which control is to be transferred. One of the following must be selected:

JMP	Indicates the left half of the word
JMPL	Indicates the left half of the word
JMPR	Indicates the right half of the word

addr An octal number which indicates the address to which control is to be transferred, or an asterisk (*). If an asterisk is used, the jump address will be either of the following:

- The starting address of the last program loaded if there has been no intervening transfer of control to any of the following SYS entries: 1SYSIN, 1ERRDMP, 1SUBERR, 1ENDJOB, 1NXTCON, or Location Zero (refer to pages 4-2 through 4-7.)
- The location specified by the JA Register upon the last transfer of control to any of the SYS entries listed above.

Note: If the specified address is greater than machine size, a Control Line Error (page 4-18) will be indicated.

REMARKS

JMP, JMPL, and JMPR will all have the same action, if * is used as the parameter.

EXAMPLE ONE

This example illustrates the use of a transfer to an absolute address.

L	Location	Command	Address and Remarks
		JMP	3240

Explanation: Execution of this instruction causes control to be transferred to the left half of location 3240_8.

EXAMPLE TWO

This example illustrates the use of the JMP command in conjunction with a loader call.

```
L  Location  Command  Address and Remarks

             JOB      GEORGE
             REL      4,BETA
              •
              •       (Other SYS control instructions)
              •
             JMP      *
```

Explanation: Execution of the above instructions causes program BETA to be loaded, and control to be transferred at a later point to the program's starting address. (Refer to REL, page 3-4.3.)

PROGTAPE — Program Tape

FUNCTION

Assigns a designated tape as the user's program tape. (Tape 4 is assumed to be the normal program tape. Refer to page 2-5.)

FORMAT

L	Location	Command	Address and Remarks
		PROGTAPE	*t*

PARAMETER

t A decimal number — 0,3,4,6, or 9 through 15 — which indicates the tape onto which programs are to be transferred from Tape 2. If any number other than those indicated above is used, a Control Line Error will result.

PRESET REQUIREMENTS

The PROGTAPE control instruction, if used, should appear prior to a TAC or ALTAC SYS control instruction in a user's job.

ACTION

When the PROGTAPE instruction is executed, PROGTAPE = *t* is typed on the Console Typewriter. All subsequent programs compiled within the current job will then be transferred to Tape *t* following compilation, subject to the same IBIT settings for Tape 4 (page 3-3.1.1).

REMARKS

1. Tape 4 is assumed to be the user's program tape for all jobs, unless reassigned by the PROGTAPE instruction.

2. Any tape designated as the program tape must contain a block of Z's, which indicates the location for the storage of the next RPL. This requirement is the same as that for the standard program tape (Tape 4).

3. A new PROGTAPE assignment may be changed only by another PROGTAPE instruction or a JOB instruction.

4. The JOB control instruction (page 3-2.1) always resets the program tape to Tape 4.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE OUT	EXPLANATION
1	PROGTAPE=*t*	A PROGTAPE instruction has been executed and all subsequent programs within the currect job will be transferred to Tape *t* following compilation, subject to normal IBIT settings.

SECTION IV
SYS ENTRIES

SYS Entries permit the programmer to make use of specialized routines used by SYS and located within the Philco Operating System. The entries are locations within SYS which contain a transfer of control to a specific routine. Some routines, at the conclusion of their operation, return control to the object program, while others transfer control to another SYS entry.

The programmer may use these routines by writing a transfer of control to the desired SYS entry. This transfer may be made to an absolute memory location or a symbolic address as designated herein. It is recommended that symbolic addresses be used.

SYSDEF

To provide a centralized source of absolute definitions of SYS symbolic locations, the subroutine SYSDEF is available for the TAC library tape. In an RPL or ABS program, SYS locations should be referred to symbolically (under the name SYS), and the subroutine SYSDEF should be called at compilation time using the TAC control instruction SUBR.

In a REL program, both symbolic references as well as REFOUTS should be used for those SYS symbols used, so that they may be defined at load time by a binary version of the subroutine. This permits the absolute locations to be changed according to future requirements without causing any current programs to be rewritten.

As an example, a transfer of control to the 1NXTCON entry should be written as:

L	Location	Command	Address and Remarks
		JMP	SYS.1NXTCON$

With such a transfer of control to a SYS location in an RPL or ABS program, the following instruction should also then appear:

L	Location	Command	Address and Remarks
		SUBR	SYSDEF$

With such a transfer of control to a SYS location in a REL program, the following instruction should also be included:

L	Location	Command	Address and Remarks
		REFOUT	SYS.1NXTCON$

For a complete list of the symbols defined by SYSDEF, refer to the SYSDEF subroutine description, SSYS-4.

LOCATION ZERO — Exponent Fault Exit

FUNCTION

Initiates SYS post-mortem dumping for a floating-point exponent fault.

SYMBOLIC ADDRESS

None

ABSOLUTE ADDRESS

Memory Location Zero

PREDOMINANT USER

The programmer

ACTION

- The contents of the Jump Address Register are saved.
- Control is transferred to the SYS post-mortem dump routine, the same as that used by 1ERRDMP (refer to page 4-4). All actions listed for 1ERRDMP, including the type-out, apply to Location Zero.

EXIT FROM ROUTINE

Following completion of the post-mortem dump routine, control is transferred to 1ENDJOB (page 4-6).

REMARKS

1. An exponent fault occurs whenever a floating-point operation generates an exponent which cannot be entirely contained in the exponent field.
2. If the programmer wishes, he may replace the automatic post-mortem dump entrance in Location Zero with an entrance to his own routine. Location Zero will be reset to the standard SYS function by execution of either a 1SYSIN function (page 4-3) or a JOB control instruction (page 3-2.1).

1SYSIN — Initialize the Philco Operating System

FUNCTION

Prepares SYS, after it is loaded into memory by an operator, for a new series of jobs.

SYMBOLIC ADDRESS

SYS.1SYSIN

ABSOLUTE ADDRESS

Memory Location 1

PREDOMINANT USER

The operator

PRESET REQUIREMENTS

If Toggle Switch 47 is set to ON prior to execution of 1SYSIN, the System will be preset to accept input in FLEXO mode (i.e., from the Console Typewriter). (Refer to CONIN, page 3-2.2.)

ACTION

- Location Zero (page 4-2) and 1SUBERR (page 4-5) are reset to standard SYS functions.
- SYS is initialized either to the FLEXO mode of input (see Preset Requirements, above) or to the standard mode of input selected by the installation (page 3-2.2). The version of SYS being used is typed on the Console Typewriter (see below).
- 1NTRYJA (the left address portion of location 60_8) is cleared to zero. (Refer to page 1-6.)
- The IBIT word (location 61_8) is cleared to zero. (Refer to page 3-8.1.)
- The dump table for the DUMP control instruction is cleared to fixed-point zeros. (Refer to page 3-7.1.)
- SYS is preset to bypass all control instructions from magnetic tape or paper tape input until a JOB control instruction is encountered.

EXIT FROM ROUTINE

Control is transferred to 1NXTCON (page 4-7) without destroying the address in 1NTRYJA.

REMARKS

1SYSIN should be used only by an operator.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	SYS *n* VERSION	SYS has been initialized, where *n* is the version of SYS being used.

1ERRDMP — Error Dump Routine

FUNCTION

Initiates SYS post-mortem dumping.

SYMBOLIC ADDRESS

SYS.1ERRDMP

ABSOLUTE ADDRESS

Memory Location 2

PREDOMINANT USER

The programmer and/or operator

ACTION

Whenever control is transferred to the SYS post-mortem dump routine, either via 1ERRDMP, Location Zero (page 4-2), or 1SUBERR (page 4-5), the following action takes place:

- The contents of all addressable registers are saved.
- A type-out occurs on the Console Typewriter, signifying that a post-mortem dump is to be performed. A zero, 2, or 3 within the type-out indicates the memory location to which control was transferred.
- The memory areas specified by all DUMP control instructions are transferred to Tape 2.
- If no DUMP or SNAP control instructions are specified, the ERRDMP routine will provide, in edited form for the High-Speed Printer, data in all addressable registers and console switches, and in memory locations 60_8 through 75_8. (Refer to DUMP Output Format, page 3-7.1.1, for a comprehensive description of output information.)

EXIT FROM ROUTINE

Control is transferred to 1ENDJOB (page 4-6), which initiates a search for the next job.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1 2 3	ERRDMP 0 ERRDMP 2 ERRDMP 3	A programmed transfer of control to Location Zero, 1ERRDMP, or 1SUBERR has been executed, or a program malfunction was detected (either by the program or operator) and control was transferred to Location Zero, 1ERRDMP, or 1SUBERR.

1SUBERR — Subroutine Error Exit

FUNCTION

Initiates SYS post-mortem dumping.

SYMBOLIC ADDRESS

1SUBERR*

ABSOLUTE ADDRESS

Memory Location 3

PREDOMINANT USER

Philco 2000 Subroutines and/or the programmer

ACTION

- The contents of the JA Register are saved.
- Control is transferred to the same SYS post-mortem dump routine as that used by 1ERRDMP (refer to page 4-4). All actions, including the type-out, listed for 1ERRDMP apply to 1SUBERR.

EXIT FROM ROUTINE

Following completion of the post-mortem dump routine, control is transferred to 1ENDJOB (page 4-6).

REMARKS

1. If an error occurs, only those subroutines which contain a standard TAC subroutine error exit will automatically transfer control to 1SUBERR.
2. If the programmer wishes, he may replace the automatic post-mortem dump entrance in 1SUBERR with an entrance to his own routine. 1SUBERR is reset to the standard SYS function by execution of either a 1SYSIN entry (page 4-3) or a JOB control instruction (page 3-2.1).
3. The above substitution feature is the only difference between 1SUBERR and 1ERRDMP.

* A symbol assigned by TAC at compilation time.

1ENDJOB — End of Job Routine

FUNCTION

Terminates a job.

SYMBOLIC ADDRESS

SYS.1ENDJOB

ABSOLUTE ADDRESS

Memory Location 4

PREDOMINANT USER

The operator and/or SYS

ACTION

- 1NTRYJA (the left address portion of location 60_8) is cleared to zero. (Refer to page 1-6.)
- Location Zero (page 4-2) and 1SUBERR (page 4-5) are reset to standard SYS functions.
- A type-out signifying job termination is typed on the Console Typewriter.

EXIT FROM ROUTINE

A search for the next JOB control instruction (page 3-2.1) is initiated.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	END OF JOB	A job has been terminated.

1NXTCON — Next Control Instruction Routine

FUNCTION	Obtains and executes the next control instruction.
SYMBOLIC ADDRESS	SYS.1NXTCON
ABSOLUTE ADDRESS	Memory Location 5
PREDOMINANT USER	SYS and/or the programmer

ACTION

- The contents of the JA Register are saved.
- The next control instruction is obtained from the ten memory locations starting at 1CONLIN via the 1NXTCRD routine (page 4-19).
- When the instruction is obtained, the command field is interpreted.

EXIT FROM ROUTINE

If the control instruction is legal, it will be executed. Illegal instructions result in a Control Line Error (refer to 1XCONER, page 4-18), causing control to be transferred to 1ERRDMP (page 4-4). The illegal control instruction remains in 1CONLIN.

1TYPOUT — Console Typewriter Type-Out

FUNCTION

Types alphanumeric characters from memory on the Console Typewriter.

SYMBOLIC ADDRESS

SYS.1TYPOUT

ABSOLUTE ADDRESS

Memory Location 6

PREDOMINANT USER

SYS and/or the programmer

PRESET REQUIREMENTS

Before control is transferred to 1TYPOUT, the A Register should be loaded in the form:

C/HLT, x; C/HLT, n

where x is the address of the first of n words to be typed on the Console Typewriter.

ACTION

- The contents of the JA Register are saved.
- The contents of n number of words beginning at address x, as specified by the A Register, are typed on the Console Typewriter.

EXIT FROM ROUTINE

Control is returned to the next sequential instruction in the object program.

REMARKS

1. It is assumed that information to be typed will be acceptable Philco 2000 Console Typewriter characters. Because 1TYPOUT does not return the Console Typewriter to Upper Case Mode, the programmer should insure that the typewriter is left in Upper Case. (Refer to Philco 2000 Character Codes, Appendix B.)
2. Trailing full words of spaces are not transmitted to the Console Typewriter.
3. The maximum number of characters per line of type-out is limited to the length of the Console Typewriter line (approximately 70 characters). Any number of lines may be typed using the carriage return character (octal 32) when appropriate.

EXAMPLE

Assume that symbolic memory locations Store1 and STORE1+1 contain the characters TOTALΔIS and Δ2054ΔΔΔ respectively.

```
L  Location  Command   Address and Remarks

             TMA       C/HLT,STORE1;C/HLT,2
             JMP       SYS.1TYPOUT
```

Explanation: Execution of these instructions causes TOTAL IS 2054 to be typed on the Console Typewriter.

1MAGRED — Read From Magnetic Tape

FUNCTION

Spaces or reads forward or backward one block from magnetic tape into memory.

SYMBOLIC ADDRESS

SYS.1MAGRED

ABSOLUTE ADDRESS

Memory Location 7

PREDOMINANT USER

SYS and/or the programmer

PRESET REQUIREMENTS

Before control is transferred to 1MAGRED, the A Register must contain a one-block read or space order, and the left address portion of the Q Register must contain the starting memory location for the order.

ACTION

- The contents of the JA Register are saved.
- One block from the tape specified by the A Register is spaced or read backward or forward into memory, starting at the location specified in the Q Register. Checks for completion and fault checks (for reading only) are included within the subroutine.

EXIT FROM ROUTINE

Upon completion of the tape order, control is returned to the next sequential instruction in the object program (except as indicated in REMARKS, below).

REMARKS

1. Two types of errors may occur during execution of a 1MAGRED, 1MAGWRT, or 1MAGREW subroutine. The first, or retry error, may be a parity or sprocket error, indicated by a HLT 5 bit configuration on the operator's console. The subroutine is automatically retried five times with each depression of the Advance Bar by the operator. The second, or non-recoverable error, is caused by a situation such as an S1, S2, or Beginning or End Tape error, indicated by a HLT 77777.
2. These three subroutines have no return to the object program nor error exit to 1SUBERR should a non-recoverable error occur. Any action following the error halt depends upon the operator or operating procedures within a particular installation. After a non-recoverable error occurs, pressing the Advance Bar transfers control to 1SYSIN (page 4-3).

3. If the tape is in local control or is being rewound, the input-output order will not be accepted and the program will loop until it is stopped manually, the tape is made available, or the rewind is completed.

EXAMPLE

One block from Tape 9 is to be read into memory beginning at location 2600_8.

```
L  Location  Command  Address and Remarks

             TMA      N/0T15;N/9T23;N/1T39;H/91T47
             TMQ      O/2600T15
             JMP      SYS.1MAGRED
```

Explanation: Execution of these instructions causes the proper input-output order to be loaded into the A Register, the memory address to be loaded into the Q Register, and control to be transferred to the 1MAGRED subroutine. Upon completion of the tape order, control is returned to the next sequential instruction in the object program.

1MAGWRT — Write Onto Magnetic Tape

FUNCTION

Spaces or writes one block from memory onto magnetic tape.

SYMBOLIC ADDRESS

SYS.1MAGWRT

ABSOLUTE ADDRESS

Memory Location 10_8

PREDOMINANT USER

SYS and/or the programmer

PRESET REQUIREMENTS

Before control is transferred to 1MAGWRT, the A Register must contain the one-block write or space order to be executed, and the left address portion of the Q Register must contain the starting memory location for the order.

ACTION

- The contents of the JA Register are saved.
- 128 words of memory, starting at the location specified in the Q Register, are written onto the tape specified by the A Register, or the tape is spaced one block, depending upon the order.

EXIT FROM ROUTINE

Upon completion of the tape order, control is returned to the next sequential instruction in the object program.

REMARKS

All remarks for 1MAGWRT are the same as those for 1MAGRED (page 4-9) except that checks for completion and fault checks are for writing only. A No Write Ring Error is treated the same as a tape in local control. (Refer to REMARK 3, 1MAGRED, page 4-9.1.)

EXAMPLE

One hundred and twenty-eight words of memory beginning at location 4102_8 are to be written onto Tape 12.

```
L   Location   Command   Address and Remarks
               TMA       N/0T15;N/12T23;N/1T39;H/19T47
               TMQ       O/4102T15
               JMP       SYS.1MAGWRT
```

Explanation: Execution of these instructions causes the proper input-output order to be loaded into the A Register, the memory address to be loaded into the Q Register, and control to be transferred to the 1MAGWRT subroutine. Upon completion of the tape order, control is returned to the next sequential instruction in the object program.

1MAGREW — Rewind Magnetic Tape

FUNCTION	Rewinds a magnetic tape.
SYMBOLIC ADDRESS	SYS.1MAGREW
ABSOLUTE ADDRESS	Memory Location 11_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	Before control is transferred to 1MAGREW, the A Register must contain the rewind order to be executed.

ACTION

- The contents of the JA Register are saved.
- The tape specified in the A Register is rewound.

EXIT FROM ROUTINE

Upon completion of the tape order, control is returned to the next sequential instruction in the object program.

REMARKS

1. Both rewind and rewind with lockout orders are acceptable.
2. If the tape is in local control, the input-output order will not be accepted and the program will loop until it is stopped manually or the tape is made available.

EXAMPLE

Assume that Tape 10 is to be rewound.

L	Location	Command	Address and Remarks
		TMA	N/10T23;H/8AT47
		JMP	SYS.1MAGREW

Explanation: Execution of these instructions causes the proper input-output order to be loaded into the A Register and control to be transferred to the 1MAGREW subroutine. Upon completion of the tape order, control is returned to the next sequential instruction in the object program.

1BLKIN — Obtain a SYS Shuttle Block

FUNCTION

Reads a SYS shuttle block into memory and transfers control to the block. (Refer to Image Portion, Tape 1, page 2-2.)

SYMBOLIC ADDRESS

SYS.1BLKIN

ABSOLUTE ADDRESS

Memory Location 12_8

PREDOMINANT USER

SYS

PRESET REQUIREMENTS

Whenever SYS requires a shuttle block from the System tape to be read into memory, it executes the following command:

C/JMP,SYS.1BLKIN;N/*n*T47

where *n* is the number of the specified block as defined in the Image Portion of Tape 1 (page 2-2).

ACTION

- The contents of the JA Register are saved in the right address portion of 1NTRYJA (location 60_8, page 1-6).
- 1BLKIN checks location 400_8 to determine whether the designated shuttle block is in memory.
- If the block is not in memory, the subroutine searches for and reads the appropriate block (specified by *n* in the C/JMP command) into memory, beginning at location 400_8.

EXIT FROM ROUTINE

When the read is completed, control is transferred to the left instruction in location 401_8.

REMARKS

The 1BLKIN subroutine uses the 1MAGRED subroutine (page 4-9) for the one-block read.

1GETBL — Get Blanks During Scanning

FUNCTION	Presets the 1SCANON subroutine (page 4-16) to recognize (not suppress) spaces in the parameters contained in the Address and Remarks field of a control instruction.
SYMBOLIC ADDRESS	SYS.1GETBL
ABSOLUTE ADDRESS	Memory Location 13_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	None
ACTION	o The contents of the JA Register are saved. o 1SCANON is set to recognize spaces.
EXIT FROM ROUTINE	Control is returned to the next sequential instruction in the object program.
REMARKS	1GETBL has no effect on the 1SCAN subroutine (page 4-15) because 1SCAN automatically resets itself to suppress spaces each time it is initiated.
EXAMPLE	(Refer to 1SCANON, page 4-16.1, for an example using the four scan subroutines.)

1IGBL — Ignore Blanks During Scanning

FUNCTION	Presets the 1SCANON subroutine (page 4-16) to suppress spaces in the parameters contained in the Address and Remarks field of a control instruction.
SYMBOLIC ADDRESS	SYS.1IGBL
ABSOLUTE ADDRESS	Memory Location 14_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	None
ACTION	• The contents of the JA Register are saved. • 1SCANON is set to suppress spaces.
EXIT FROM ROUTINE	Control is returned to the next sequential instruction in the object program.
REMARKS	1IGBL has no effect on the 1SCAN subroutine (page 4-15) because 1SCAN automatically resets itself to suppress spaces each time it is initiated.
EXAMPLE	(Refer to 1SCANON, page 4-16.1, for an example using the four scan subroutines.)

1SCAN – Scan the First Parameter of a Control Instruction

FUNCTION	Initializes the SYS scanning subroutine and stores the first parameter contained in the Address and Remarks field of a control instruction.
SYMBOLIC ADDRESS	SYS.1SCAN
ABSOLUTE ADDRESS	Memory Location 15_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	None

ACTION

- The contents of the JA Register are saved.
- The <u>first</u> parameter in the Address and Remarks field of a control instruction is scanned and stored in 1WRD1 and 1WRD2 (locations 315_8 and 316_8). Spaces are always suppressed.
- Termination of the first parameter is determined by the appearance of a standard SYS break character (a comma, semicolon, slash, period, or dollar sign). This break character appears in the A Register, right-justified with leading zeroes.
- All characters of the first parameter (a maximum of 16) are stored upon exit from the subroutine into 1WRD1 and 1WRD2, acting as a double-length register. The characters are stored right-justified with leading zeroes.
- The number of characters recognized by the 1SCAN and 1SCANON subroutines is stored in the left address portion of 1CHARCT (location 26_8), just prior to the exit from the subroutine. (Refer to page 1-6.)
- Because the subroutine stores 16 or fewer characters, the results are not guaranteed if more than 16 are used.

EXIT FROM ROUTINE

Upon completion of the subroutine, control is returned to the next sequential instruction in the object program.

REMARKS

1. 1SCAN always suppresses spaces within the first parameter.
2. The 1SCAN subroutine uses Index Registers 1 and 2 and does not restore them.
3. If additional parameters of the same control line are to be scanned, 1SCANON (page 4-16) must be used.

EXAMPLE

(Refer to 1SCANON, page 4-16.1, for an example using the four scan subroutines.)

1SCANON — Scan Additional Parameters of a Control Instruction

FUNCTION	Continues the scanning process begun by 1SCAN (page 4-15) by storing the next parameter contained in the Address and Remarks field of a control instruction.
SYMBOLIC ADDRESS	SYS.1SCANON
ABSOLUTE ADDRESS	Memory Location 16_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	1. If spaces in the parameter are not to be suppressed, 1GETBL (page 4-13) must be executed prior to entering 1SCANON. (1SCAN always presets 1SCANON to suppress spaces.) 2. 1IGBL (page 4-14) must be executed to override a 1GETBL setting within the same scan routine.
ACTION	The same action listed for 1SCAN takes place for 1SCANON, except that the parameter beginning at the first column following the last break character encountered is scanned and stored. The use of 1SCAN <u>must</u> precede the use of 1SCANON; otherwise unpredictable results will occur.
REMARKS	1. Additional 1SCANON calls must be given for subsequent parameters to be scanned. 2. 1SCANON is always initially set by 1SCAN to suppress spaces. 3. The automatic settings of 1SCANON to suppress spaces continues through each subsequent use of 1SCANON until changed by a 1GETBL call. 1GETBL then remains in effect until a 1IGBL or a 1SCAN call is executed. 4. 1SCANON uses Index Registers 1 and 2 and does not restore them.

EXAMPLE

Assume that the Address and Remarks field of a control line stored in 1CONLIN contains the following characters:

5,0 ΔBCΔ/DΔEFΔΔΔΔΔG$

```
L   Location   Command    Address and Remarks

                  •
                  •
               JMP        SYS.1SCAN
                  •
                  •
```

Explanation: Execution of this instruction causes the first parameter (terminated by the comma) to be scanned and the single character (5) within it to be placed into 1WRD1 and 1WRD2 as W/00000000 and W/00000005 respectively. The break character (,) is placed into the A Register as W/0000000, and the number of characters (1) is placed as C/HLT,1 into 1CHARCT. A transfer of control to 1GETBL issued prior to the transfer of control to 1SCAN would have no effect on the above results.

```
L   Location   Command    Address and Remarks

                  •
                  •
               JMP        SYS.1SCANON
                  •
                  •
```

Explanation: Execution of this instruction causes the second parameter (terminated by a slash) to be scanned and the characters within it (0ΔBCΔ) to be placed into 1WRD1 and 1WRD2 as W/00000000 and W/000000BC respectively. The break character (/) is placed into the A Register as W/0000000/ and the number of characters (3) is placed as C/HLT,3 into 1CHARCT. Note that leading zeroes must be determined by the character count.

```
L   Location   Command    Address and Remarks

                  •
                  •
               JMP        SYS.1GETBL
               JMP        SYS.1SCANON
                  •
                  •
```

Explanation: Execution of these instructions causes the third parameter (terminated by a dollar sign) to be scanned and the characters within it (DΔEFΔΔΔΔΔG) to be placed into 1WRD1 and 1WRD2 as W/000000DΔ and W/EFΔΔΔΔΔG respectively. The break character ($) is placed into the A Register as W/0000000$ and the number of characters (10) is placed as C/HLT,10 into 1CHARCT.

1STRIPQ — Strip the Q Register of Spaces

FUNCTION	Suppress space characters in a word.
SYMBOLIC ADDRESS	SYS.1STRIPQ
ABSOLUTE ADDRESS	Memory Location 17_8
PREDOMINANT USER	SYS and/or the programmer
PRESET REQUIREMENTS	Before control is transferred to 1STRIPQ, the word to be stripped must be loaded into the Q Register.
ACTION	o The contents of the JA Register are saved. o All spaces are deleted from the word in the Q Register. o The stripped word is then transferred to the A Register, right-justified with leading zeroes.
EXIT FROM ROUTINE	Control is returned to the next sequential instruction in the object program.

EXAMPLE

```
L   Location   Command   Address and Remarks

                  o
                  o
                 TMQ      W/ΔSΔAΔΔMΔ$
                 JMP      SYS.1STRIPQ
                  o
                  o
```

Explanation: Execution of these instructions causes W/00000SAM to be placed in the A Register.

1XCONER — Control Line Error Type-Out

FUNCTION

Causes CONT. LINE ERR......... to be typed on the Console Typewriter, and SYS post-mortem dumping to be initiated.

SYMBOLIC ADDRESS

SYS.1XCONER

ABSOLUTE ADDRESS

Memory Location 20_8

PREDOMINANT USER

SYS

PRESET REQUIREMENTS

None

ACTION

Whenever control is transferred to 1XCONER, the following action takes place:

- The contents of the JA Register are stored in the left address portion of 1NTRYJA (location 60_8, refer to page 1-6).
- CONT. LINE ERR......... is typed on the Console Typewriter.

EXIT FROM ROUTINE

- Following the type-out, control is transferred to the SYS post-mortem dump routine (refer to 1ERRDMP, page 4-4).

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	CONT. LINE ERR.........	An illegal command or illegal parameter (of most control instructions) was detected in a control instruction being interpreted by SYS. Control is transferred to 1ERRDMP.

1NXTCRD — Obtain the Next Card

FUNCTION

Obtains from the System input tape (Tape 8) the next card in Code Mode (converting from Image if necessary) and stores it into the ten memory locations beginning at 1CONLIN (location 64_8).

SYMBOLIC ADDRESS

SYS.1NXTCRD

ABSOLUTE ADDRESS

Memory Location 21_8

PREDOMINANT USER

SYS and/or the programmer

PRESET REQUIREMENTS

If Bit 43 of 1CONBIT (location 63_8, page 1-6), is set to one, only the first six words of an Image Mode card will be converted to three words of Code Mode and transferred to 1CONLIN.

ACTION

- The location of the first word of the next card, which was read into an input buffer area in memory as part of a block from the System input tape (Tape 8), is stored in Index Register 1. This buffer area contains the image card if in Magtape, Image Mode (refer to CONIN, page 3-2.2.)
- If input is in Code Mode, the ten words in the buffer area which compose the next card will be transferred directly to 1CONLIN.
- If input is in Image Mode, it will automatically be converted to Code Mode during its transfer to 1CONLIN (subject to Bit 43 of 1CONBIT), and the location of the image card in the buffer area will be placed into Index Register 1. Illegal Image Mode characters appear as question marks (?) in Code Mode.

EXIT FROM ROUTINE

Control is returned to the next sequential instruction in the object program.

REMARKS

1. The features of 1NXTCRD are used most frequently by SYS to obtain control instructions for 1NXTCON (page 4-7), which executes them.
2. If a JOB card (page 3-2.1) is encountered, Error Type-Out 1 (page 4-19.1) will occur. The next job to be executed is the job just intercepted.

3. 1NXTCRD uses 1MAGRED (page 4-9) to load the input buffer from the input tape.

4. If input is via the Console Typewriter or paper tape, the next "card" will be obtained from that source, rather than from Tape 8.

CONSOLE TYPEWRITER ERROR TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	JOB CARD INTERCEPTED	A JOB card was intercepted by the 1NXTCRD subroutine. A post-mortem dump (1ERRDMP, page 4-4) will occur, following which the job specified by the intercepted JOB card will be processed.

EXAMPLE

Hollerith cards with pertinent data in the first word are to be read and processed individually from the SYS input tape.

```
L  Location  Command  Address and Remarks

   EXAMINE   JMP       SYS.1NXTCRD
             TMA       SYS.1CONLIN
             TMD       W/ENDINPUT
             JAED      FINISH
              •
              •        (Processing data)
              •
             JMP       EXAMINE
```

Explanation: Execution of these instructions causes the first word of each card to be placed into 1CONLIN. Data in 1CONLIN is then transferred to the A Register, where it is compared with W/ENDINPUT placed in the D Register. If the contents of the A and D Registers are not equal, the card is processed and the next card is compared. When the ENDINPUT card is encountered, equality between the two registers occurs and program control is transferred to FINISH.

1INTLD — Internal Loader

FUNCTION	Loads a program
SYMBOLIC ADDRESS	SYS.1INTLD
ABSOLUTE ADDRESS	Memory Location 22_8
PREDOMINANT USER	SYS and/or the programmer

PRESET REQUIREMENTS

Before control is transferred to 1INTLD, the A and Q Registers must be loaded as follows:

1. The identification *id* of the program to be loaded is to be placed in the A Register. The *id* must be eight characters or fewer, left-justified and spaced-filled to the right if necessary.

2. The following specified bit designations are to be placed in the Q Register:

Bits 20-23	The logical number of the tape on which the program is stored.
Bits 25-39	An offset value, which is to be added to the normal loading addresses. The combined effective address must not be less than 1000_8 or a Control Line Error will result.
Bit 41	If set to one, it is presumed that the program exists in restricted RPL format (page 3-5) and therefore may be loaded in a more efficient manner. (Effective only if Bit 47 is set to one.)
Bit 42	If set to one, subroutines will be loaded from Tape 7. (Effective only if Bit 45 is set to one.)
Bit 43	If set to one, the NAME/SYMBOL list will be printed. (Effective only if Bit 45 is set to one.)
Bit 44	If set to one, control is to be transferred to the starting address of a program after it is loaded.

Bit 45	If set to one, signifies that a REL program is to be loaded.
Bit 46	If set to one, signifies that an ABS program is to be loaded.
Bit 47	If set to one, signifies that an RPL program is to be loaded.

CAUTION: If Bits 45, 46, and 47 are all zero, Error Type-Out 1 (page 4-20.2) will occur and control will be transferred to 1ERRDMP (page 4-4). If more than one of these bits is given, the first listed (in the order 47, 46, and 45) will be assumed to be the correct bit.

ACTION

- The contents of the JA Register are saved.
- The tape specified by the Q Register will be searched forward to the sentinel block of Z's for the program designated by *id.* If the program is not found, the tape will be searched backward to the beginning-of-tape sensing strip. If the program still cannot be found, Error Type-Out 2 will be typed on the Console Typewriter and control will be transferred to 1ERRDMP (page 4-4).
- When the designated program is found, it is loaded into memory in the manner signified by the setting of Bits 41, 45, 46, and 47.
- Control will be transferred to the program just loaded if Bit 44 of the Q Register is set to one, or will be returned to the object program if Bit 44 is not set.

REMARKS

1. Internal loader calls may be inserted in a program at compilation time by use of the internal load generator, LOADGEN (refer to page 6-2).
2. The internal loader is especially useful during the running of segmented programs, for it enables one program to call in another without exiting to an external control instruction. (Refer to page 3-5.)

EXAMPLE

The following example illustrates the calling of an object program into memory by another object program during the latter's execution.

```
L   Location   Command   Address and Remarks

                  •

                  •

                  •

               TMA       W/SEG1ΔΔΔΔ$
               TMQ       N/6 T23;1/1 T47$
               JMP       SYS.1INTLD

                  •

                  •

                  •
```

Explanation: Execution of these instructions causes the ID of the segment to be loaded to be placed in the A Register, the RPL load order for Tape 6 to be placed in the Q Register and the eventual loading of the segment into memory. Control is returned to the object program because Bit 44 was zero.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	FORM OF LOAD IS NOT RPL, ABS, OR REL	Occurs if neither Bit 45, Bit 46, nor Bit 47 has been set to one when the internal loading feature of SYS is used. Control is transferred to 1ERRDMP (page 4-4).
2	NOT HERE	The program designated by *id* has not been found during a search forward and backward on Tape *t*. Control is transferred to 1ERRDMP. This type-out will not occur if Bits 41 and 47 were both set to one.
3	LOAD ERR	The internal load function did not receive proper indication of the type of program to be loaded (RPL, ABS, or REL.) Control is transferred to 1ERRDMP.

SECTION V

SYS SERVICE ROUTINES

SYS Service Routines are specialized programs designed specifically to aid the programmer in performing routine functions of tape maintenance, debugging and program monitoring. They are located on the System Program Tape (Tape 1) in RPL format, and are available any time SYS is being used.

Service routines are called via their particular program identity as described below. Any or all of their particular functions are then performed depending upon the programmer's use of optional control instructions recognized by the routine itself. These control instructions, including the command, parameters, and break characters, are written in the same manner as that described for SYS control lines (page 3-1).

CALLING SERVICE ROUTINES

Service routines may be called into memory from Tape 1 by two methods. The first, which is applicable to all service routines, uses the normal RPL control instruction (page 3-4.1). Tape 1, the routine's program identity, and the GO option should be written as its first three parameters.

Execution of the instruction in the following example causes the service routine BINDEL (page 5-4) to be called into operation:

L	Location	Command	Address and Remarks
		RPL	1,BINDEL,GO

The second method, applicable only to three service routines (SYSAIDE, TACSERVS and SYSRPLC), permits their loading via their corresponding SYS control lines (AIDE, TACSERV, and RPLC).

Execution of the instruction in the following example causes the service routine SYSAIDE (page 5-2) to be called into operation:

L	Location	Command	Address and Remarks
		AIDE	

Service routines at some installations may also be available on binary cards. These card decks may be loaded and run by an "ABS *" control instruction (page 3-4.2), thus eliminating the tape searching time required when calling the routines from tape.

Execution of the instruction in the following example causes the service routine TACSERVS (page 5-9) in ABS deck form to be called into operation:

```
L  Location  Command  Address and Remarks

                ABS    *,,GO
                 )
                 }     (Binary deck of TACSERVS)
                 )
```

TERMINATING SERVICE ROUTINES

Upon completion of the designated operations of a service routine, an ending instruction must be specified to terminate the routine.

Two such instructions, END and ENDALL, in general contain no parameters and serve only to halt action of the service routine and return control to SYS.

Where parameters or additional functions are included with a termination instruction, the instruction is described within the individual description of the service routine.

AIDE (SYSAIDE) SERVICE ROUTINE

FUNCTION

Copies and/or compares information contained on magnetic tapes.

SERVICE ROUTINE INITIATION

The SYSAIDE service routine may be called into operation by any of the following three instructions:

L	Location	Command	Address and Remarks
		AIDE	
		RPL	1,SYSAIDE,GO
		ABS }	*,,GO (Binary deck of SYSAIDE)

INPUT-OUTPUT SYSTEM

SYSAIDE uses PROC, a Philco input-output system, for all reads, writes, and tape movements. Errors are treated by the TAC subroutine, TAPER. (Refer to Philco Program Report 14, Tape Error Routine.)

CONTROL INSTRUCTIONS

SYSAIDE accepts seven control instructions, written in standard SYS control line format:

INSTRUCTION	FUNCTION
COPY	Transfers information without change from one tape to another.
COMPF	Compares information on two tapes in a forward direction.
COMPB	Compares information on two tapes in a backward direction.
COPYCOMP	Transfers information from one tape to another and then compares the information copied by reading both tapes in a backward direction.
REWIND	Rewinds one or more tapes.
REWINDLO	Rewinds one or more tapes with lockout.
ENDALL	Terminates the SYSAIDE service routine.

Illegal parameters generally cause SYSAIDE to return control immediately to SYS.

COPY

Transfers information without change from one magnetic tape to another.

Format

L	Location	Command	Address and Remarks
		COPY	*from,to,nbp*

Parameters

from A decimal number, 0 through 15, which indicates the tape from which information is to be transferred.

to A decimal number, 0 through 15, which indicates the tape to which information is to be transferred.

nbp A decimal number, 1 through 19,000, which indicates the number of blocks to be processed (copied). If zero is specified as *nbp,* the COPY command is ignored and the next SYSAIDE control instruction is requested. If a number greater than 19,000 is specified as *nbp,* the COPY command is ignored and control is returned to SYS.

Action

When the COPY control instruction is executed, *nbp* number of blocks are copied in the same mode from Tape *from* to Tape *to.* Following completion of the transfer, FINIS is typed on the Console Typewriter, and the tapes are positioned at the end of the last block copied. The SYSAIDE routine then requests its next control instruction.

Example

L	Location	Command	Address and Remarks
		COPY	9,14,512

Explanation: Execution of this instruction causes 512 blocks of information from Tape 9 to be copied onto Tape 14.

COMPF

Compares information on one magnetic tape with information on a second magnetic tape by reading both tapes in a forward direction.

Format

L	Location	Command	Address and Remarks
		COMPF	t_1,t_2,*nbp*,*m*

Parameters

t_1 A decimal number, 0 through 15, which indicates the first tape to be compared.

t_2 A decimal number, 0 through 15, which indicates the tape to be compared with the first.

nbp A decimal number, 1 through 19,000, which indicates the number of blocks to be processed (compared). (Refer to *nbp* of COPY for error indications.)

m (OPTIONAL). An alphabetic character which specifies the mode of output to be used for indicating inequalities. If the parameter is omitted, T is assumed.

F	Specifies that output is to be on the Console Typewriter
T	Specifies that output is to be on Tape 5, edited for the High-Speed Printer with Data Select 0

Action

When the COMPF command is executed, the following action takes place:

- *nbp* number of blocks on t_1 are compared with the same number of blocks on t_2.
- Any inequalities are printed on *m*. (Refer to page 5-2.5 for a description of the output.)
- If no differences are detected, NO DIFFERENCES will be typed on the Console Typewriter.
- Following the comparison of all information, both tapes are positioned at the end of the last block compared.
- SYSAIDE then requests its next control instruction.

Example

L	Location	Command	Address and Remarks
		COMPF	9,14,512,T

Explanation: Execution of this instruction causes 512 blocks of information on Tape 9 to be compared in a forward direction with 512 blocks of Tape 14. Any inequalities are to be recorded on Tape 5.

COMPB

Compares information on one magnetic tape with information on a second magnetic tape by reading both tapes in a backward direction.

Format

L	Location	Command	Address and Remarks
		COMPB	t_1,t_2,*nbp*,*m*

Remarks

The parameters, action, and example for COMPB are the same as those of the COMPF control instruction (page 5-2.1), except that the comparison is made in a backward direction, and FINIS will be typed on the Console Typewriter if no errors or inequalities are detected.

COPYCOMP

Transfers information from one magnetic tape to another, and then compares the information copied by reading both tapes in a backward direction.

Format

L	Location	Command	Address and Remarks
		COPYCOMP	*from,to,nbp,m*

Parameters

from A decimal number, 0 through 15, which indicates the tape from which information is to be transferred.

to A decimal number, 0 through 15, which indicates the tape to which information is to be transferred.

nbp A decimal number, 1 through 19,000, which indicates the number of blocks to be processed (copied and compared). (Refer to *nbp* on page 5-2.1 for error indications.)

m (OPTIONAL). An alphabetic character which specifies the mode of output to be used for indicating inequalities. If the parameter is omitted, T is assumed.

F	Specifies that output is to be on the Console Typewriter
T	Specifies that output is to be on Tape 5, edited for the High-Speed Printer with Data Select 0

Action

When the COPYCOMP command is executed, the following action takes place:

- *nbp* number of blocks are copied in the same mode from Tape *from* to Tape *to*.
- The *nbp* number of blocks copied from Tape *from* are then compared backward with the same number of blocks on Tape *to*.
- Any inequalities are printed on *m* (see above).
- If no differences are detected, FINIS will be typed on the Console Typewriter.
- When all the information is transferred and compared, the two tapes are positioned exactly as they were prior to the execution of the control instruction.
- SYSAIDE then requests its next control instruction.

Example

```
L  Location  Command    Address and Remarks

             COPYCOMP   9,14,512,F
```

Explanation: Execution of this instruction causes 512 blocks of information to be copied from Tape 9 onto Tape 14. The information recorded on Tape 14 is then compared in a backward direction with the information on the original tape. Any inequalities will be typed on the Console Typewriter.

REWIND

Rewinds one or more magnetic tapes without lockout.

Format

L	Location	Command	Address and Remarks
		REWIND	*t*, . . . ,*t*

Parameters

t One or more decimal numbers, 0 through 15, which indicate the tapes to be rewound.

Action

When the REWIND control instruction is executed, all tapes indicated by *t*,...,*t* are rewound without lockout. SYSAIDE then requests its next control instruction.

Example

```
L  Location  Command  Address and Remarks

             REWIND   3,9,11,14
```

Explanation: Execution of this instruction causes Tapes 3, 9, 11, and 14 to be rewound without lockout.

REWINDLO

Rewinds one or more magnetic tapes with lockout.

Format

L	Location	Command	Address and Remarks
		REWINDLO	*t*,...,*t*

Remarks

The parameters, action, and example for REWINDLO are the same as those of the REWIND control instruction, above, except that tapes are rewound with lockout.

HIGH-SPEED PRINTER FORMAT

Magnetic tape output is generated by SYSAIDE whenever an inequality is detected during a comparison, and the mode parameter contains the character T. The printed page output from the High-Speed Printer appears as follows:

```
      FROM t1          BLOCK  WORD            TO t2

AAAAAAAAAAAAAAAA  BBBBB   CCC   DDDDDDDDDDDDDDDD
        ●            ●     ●           ●
        ●            ●     ●           ●
        ●            ●     ●           ●
AAAAAAAAAAAAAAAA  BBBBB   CCC   DDDDDDDDDDDDDDDD

             END COMPARISONS SYSAIDE
```

The characters t_1 and t_2 represent the numbers of the magnetic tapes which were compared. The 16 A characters are the octal representation of the word contained on the first comparison tape. The five B characters are the decimal representation of the block in which the inequality was detected. The three C characters are the decimal representation of the word in which the inequality was detected. The 16 D characters are the octal representation of the same word as described by the A's contained on the second comparison tape. Each inequality is displayed on a separate line. All inequalities detected during any one control instruction (COMPF, COMPB, or COPYCOMP) are grouped as a section of output, which is preceded by the heading and followed by the message: END COMPARISONS SYSAIDE.

CONSOLE TYPEWRITER OUTPUT FORMAT

Inequalities are typed on the Console Typewriter if the character F is specified in the mode parameter. The typewriter page is the same as that described for the High-Speed Printer page except the heading and end messages do not appear.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	FINIS	The operation requested by a COPY, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine has been completed. No errors or inequalities were detected during the COMPB or COPYCOMP comparisons.
2	NO DIFFERENCES	The operation requested by a COMPF control instruction within the SYSAIDE service routine has been completed, and no errors or inequalities were detected.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	U.N.A. *t*	SYSAIDE attempted to issue an order to Tape *t* which was not available. If the Advance Bar is pressed, SYSAIDE will wait until operator intervention makes Tape *t* available and then will continue normal operation.
2	ILLEGAL BLK CNT	A decimal number larger than 19,000 was specified in an *nbp* parameter of a COPY, COMPF, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine. SYSAIDE will return control immediately to SYS.
3	ILLEGAL TAPE NUMBER	Either an alphabetic character or a decimal number larger than 15 was entered as a *from, to,* t_1 or t_2 parameter of a COPY, COMPF, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine. SYSAIDE will return control immediately to SYS.
4	NO ROCK	A possible error in the SYSAIDE program was detected. An error dump will be provided by SYS. (This type-out should not normally occur.)
5	CONTROL LINE ERROR	An illegal control instruction was encountered by SYSAIDE. Control is transferred to 1XCONER (page 4-18).

SPECIAL CONSOLE TYPEWRITER TYPE-OUT

Whenever Tape 4 (user's program tape) is to be written upon during operation of the SYSAIDE program, the following message is typed on the Console Typewriter:

```
PROGRAM ABOUT TO WRITE ON 4.  FLEXO WILL NOW
ACCEPT DESIRED TO TAPE NUMBER.
```

SYSAIDE does not execute the control instruction which requested the write order, but waits for an operator type-in from the Console Typewriter. The type-in may be <u>any</u> decimal number from zero through 15 followed by a period, and is interpreted as the number of the new tape onto which the writing is to take place. Tape 4 is now acceptable. The purpose of this check is to prevent inadvertent destruction of the user's program tape.

SYSAIDE ROUTINE EXAMPLE

```
L  Location  Command    Address and Remarks
             AIDE                      $    (1)
             REWIND     3,6,9          $    (2)
             COPY       3,9,150        $    (3)
             COPYCOMP   6,9,335,F      $    (4)
             COMPB      3,9,150,F      $    (5)
             REWIND     3,6            $    (6)
             REWINDLO   9              $    (7)
             ENDALL                    $    (8)
```

Explanation — Execution of these instructions causes the following action to take place:

1. SYSAIDE is called into operation.
2. Tapes 3, 6, and 9 are rewound.
3. 150 blocks of information are copied from Tape 3 to Tape 9.
4. 335 blocks of information are copied from Tape 6 onto Tape 9 and are then compared in a backward direction. Any differences are typed on the Console Typewriter.
5. 150 blocks of information (referred to in Step 3) are compared in a backward direction. Any differences are typed on the Console Typewriter.
6. Tapes 3 and 6 are rewound without lockout.
7. Tape 9 is rewound with lockout.
8. Control is returned to SYS.

ANALYZER SERVICE ROUTINE

FUNCTION

Prepares a listing of all references made by a program or a portion of a program to a specified area in memory. Any portion or all of memory may be specified.

SERVICE ROUTINE INITIATION

The ANALYZER service routine may be called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,ANALYZER,GO
		ABS }	*,,GO (Binary deck of ANALYZER)

PROGRAM TO BE ANALYZED

The program to be analyzed may be in RPL or ABS format and may appear on any magnetic tape except 3, 5, or 6. It is not loaded into memory. An ABS program on tape should be preceded by a dummy PMAX card. If this card is not present, the ABS program may still be run by positioning the tape at the start of the program.

INPUT-OUTPUT SYSTEM

ANALYZER contains its own input-output system for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

The ANALYZER service routine accepts two control instructions: ANALYZE and ENDALL. The instructions are written in standard SYS control line format. Illegal parameters (except where otherwise noted) cause a Control Line Error to occur (refer to 1XCONER, page 4-18).

ANALYZE

Searches for a specific program on a designated tape, examines all instructions of that program, and prepares a listing of those instructions within a designated portion of the program which reference a specified area of memory. Any number of ANALYZE instructions may appear prior to an ENDALL instruction, and each ANALYZE instruction may reference the same or different programs.

Format

L	Location	Command	Address and Remarks
	format	ANALYZE	*t,id,beg-c,end-c,beg-m,end-m*

Parameters

format Format of the program to be analyzed. One of the following must be selected:

RPL	Indicates that the program is in RPL format
ABS	Indicates that the program is in ABS format

t A decimal number, 0 through 15 (excluding 3, 5, 6, and 8) which indicates the tape that contains the input program. The parameter may also be an asterisk (*), in which case an ABS deck must follow the control instruction on Tape 8.

id 16 or fewer alphanumeric characters which indicate the identity of the input program. Spaces are significant. If the *id* is eight characters or fewer, only the first eight characters of each program on tape will be compared with the *id* during the search. (Refer to ONE-WORD SEARCH, page 3-4.) If this parameter is omitted, Tape *t* will be assumed to be positioned at the start of the ABS program.

beg-c An octal number which indicates the beginning location of the portion of coding to be analyzed.

end-c An octal number which indicates the ending location of the portion of coding to be analyzed.

beg-m An octal number which indicates the beginning location of the memory area to be analyzed for references by the portion of coding designated by *beg-c* to *end-c*.

end-m An octal number which indicates the ending location of the memory area to be analyzed for references by the portion of coding designated by *beg-c* to *end-c*.

Action

When the ANALYZE control instruction is executed, the following action takes place:

- Tape *t* is searched for program *id*.
- If a sentinel block of Z's is encountered prior to the designated *id*, the tape will be searched backward to the beginning of the tape. If the program still is not encountered, *id* MISSING will be typed on the Console Typewriter and the current ANALYZE control instruction will be ignored. ANALYZER then requests its next control instruction.
- After the designated *id* is located, *format* ANALYZE *id* is typed on the Console Typewriter.
- Each instruction of the program to be analyzed is then examined individually from tape to determine:

1. If its location falls within the portion of coding from *beg-c* to *end-c*.

2. If the memory address which it references falls within the memory area from *beg-m* to *end-m*.

3. If the instruction contains a reference to an index register or is an index register-modified instruction.

- All instructions meeting conditions (1) and (2) or (1) and (3) will be sequenced by referenced location and will be stored in memory.
- If memory is filled before input is exhausted, the sequenced instructions will be written onto Tape 3. If memory is again filled before input is exhausted, the second portion of sequenced instructions will be written on Tape 6.
- If memory is filled a third time, and input is still not exhausted, the statement: PROGRAM EXCEEDED ANALYZER MEMORY CAPACITY is placed on the last page of output. Any remaining portion of the input program will not be processed by the current ANALYZER call.
- All processed output is then written onto Tape 5, for printing on the High-Speed Printer in Data Select 0.
- Input tape *t* is normally positioned following the final block of the analyzed program.
- Tapes 3 and 6 are set to their positions prior to use by the ANALYZE control instruction.
- The next ANALYZER control instruction is requested.

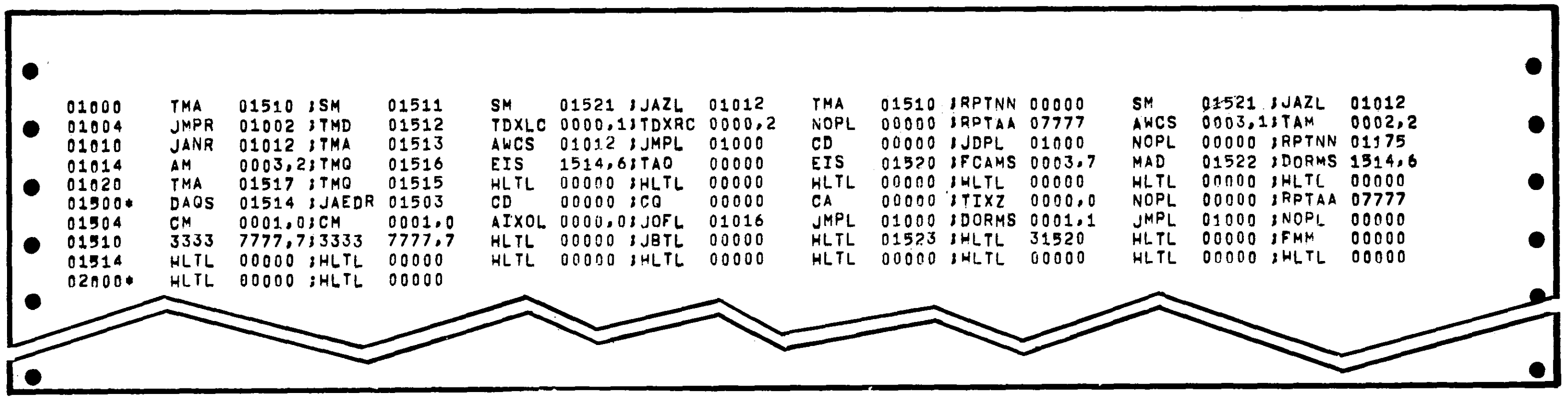
Example

L	Location	Command	Address and Remarks
	RPL	ANALYZE	4,ALTAC3,1000,12000,0,77777$

Explanation: Execution of this instruction causes a listing of RPL program ALTAC3, located on Tape 4, to be prepared. The listing will include those references to memory locations 0 through 77777_8 made by program locations 1000_8 through 12000_8.

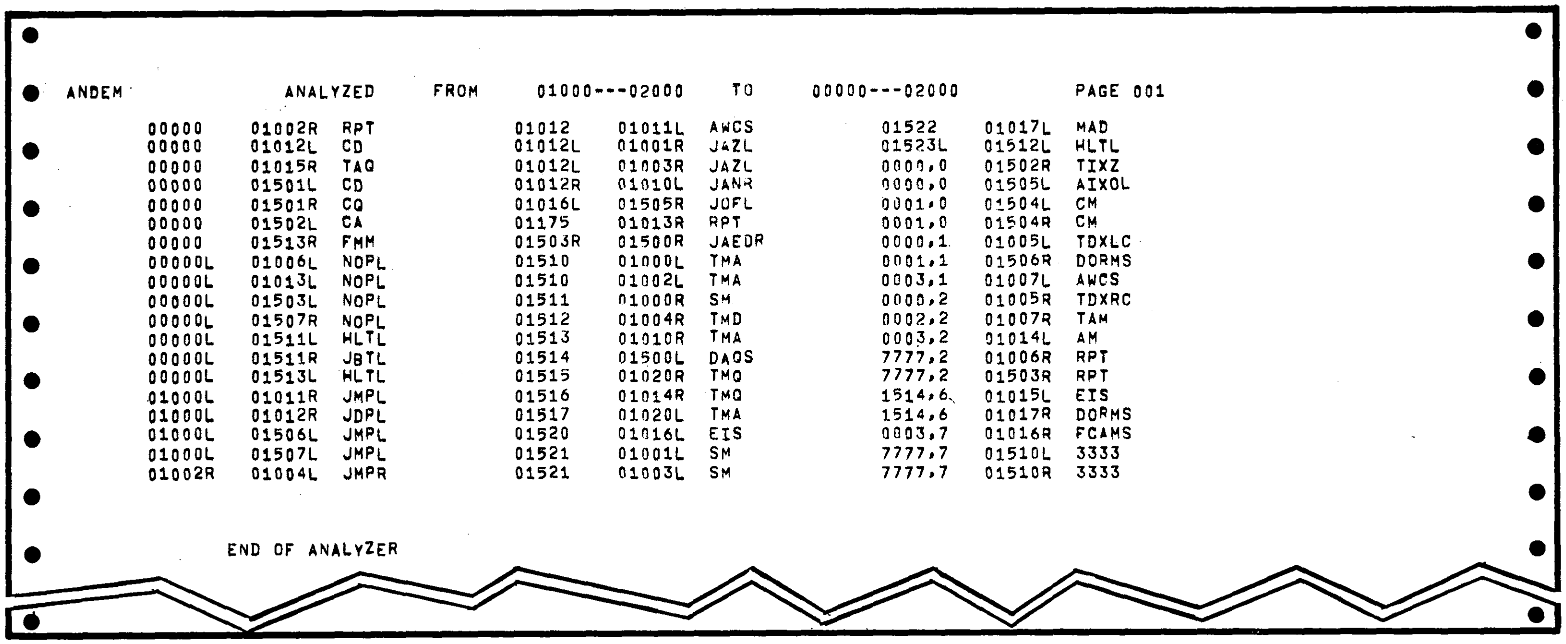
OUTPUT FORMAT

The output of the ANALYZER service routine is written onto Tape 5 for off-line printing on the High-Speed Printer with Data Select 0. Each page of output consists of a one-line heading statement, a listing of the sorted analyzed output, and an ending statement. (Refer to page 5-3.3 for an illustration of ANALYZER output.)



```
01000    TMA   01510 ;SM     01511    SM     01521 ;JAZL   01012    TMA   01510 ;RPTNN 00000    SM    01521 ;JAZL  01012
01004    JMPR  01002 ;TMD    01512    TDXLC  0000,1;TDXRC  0000,2   NOPL  00000 ;RPTAA 07777    AWCS  0003,1;TAM   0002,2
01010    JANR  01012 ;TMA    01513    AWCS   01012 ;JMPL   01000    CD    00000 ;JDPL  01000    NOPL  00000 ;RPTNN 01175
01014    AM    0003,2;TMQ    01516    EIS    1514,6;TAQ    00000    EIS   01520 ;FCAMS 0003,7   MAD   01522 ;DORMS 1514,6
01020    TMA   01517 ;TMQ    01515    HLTL   00000 ;HLTL   00000    HLTL  00000 ;HLTL  00000    HLTL  00000 ;HLTL  00000
01500*   DAQS  01514 ;JAEDR  01503    CD     00000 ;CQ     00000    CA    00000 ;TIXZ  0000,0   NOPL  00000 ;RPTAA 07777
01504    CM    0001,0;CM     0001,0   AIXOL  0000,0;JOFL   01016    JMPL  01000 ;DORMS 0001,1   JMPL  01000 ;NOPL  00000
01510    3333  7777,7;3333   7777,7   HLTL   00000 ;JBTL   00000    HLTL  01523 ;HLTL  31520    HLTL  00000 ;FMM   00000
01514    HLTL  00000 ;HLTL   00000    HLTL   00000 ;HLTL   00000    HLTL  00000 ;HLTL  00000    HLTL  00000 ;HLTL  00000
02000*   HLTL  00000 ;HLTL   00000
```

Program ANDEM As It Would Appear If Loaded Into Memory †



```
ANDEM            ANALYZED     FROM      01000---02000     TO      00000---02000           PAGE 001

         00000      01002R   RPT               01012      01011L   AWCS            01522      01017L   MAD
         00000      01012L   CD                01012L     01001R   JAZL            01523L     01512L   HLTL
         00000      01015R   TAQ               01012L     01003R   JAZL            0000,0     01502R   TIXZ
         00000      01501L   CD                01012R     01010L   JANR            0000,0     01505L   AIXOL
         00000      01501R   CQ                01016L     01505R   JOFL            0001,0     01504L   CM
         00000      01502L   CA                01175      01013R   RPT             0001,0     01504R   CM
         00000      01513R   FMM               01503R     01500R   JAEDR           0000,1     01005L   TDXLC
         00000L     01006L   NOPL              01510      01000L   TMA             0001,1     01506R   DORMS
         00000L     01013L   NOPL              01510      01002L   TMA             0003,1     01007L   AWCS
         00000L     01503L   NOPL              01511      01000R   SM              0000,2     01005R   TDXRC
         00000L     01507R   NOPL              01512      01004R   TMD             0002,2     01007R   TAM
         00000L     01511L   HLTL              01513      01010R   TMA             0003,2     01014L   AM
         00000L     01511R   JBTL              01514      01500L   DAQS            7777,2     01006R   RPT
         00000L     01513L   HLTL              01515      01020R   TMQ             7777,2     01503R   RPT
         01000L     01011R   JMPL              01516      01014R   TMQ             1514,6     01015L   EIS
         01000L     01012R   JDPL              01517      01020L   TMA             1514,6     01017R   DORMS
         01000L     01506L   JMPL              01520      01016L   EIS             0003,7     01016R   FCAMS
         01000L     01507L   JMPL              01521      01001L   SM              7777,7     01510L   3333
         01002R     01004L   JMPR              01521      01003L   SM              7777,7     01510R   3333


               END OF ANALYZER
```

ANALYZER Output of Program ANDEM ††

† Obtained by a post-mortem dump in Command format.

†† The control instruction used to obtain the output was: ABS ANALYZE 4,ANDEM,1000,2000,0,2000$

Heading Statement

The heading statement appears in the following format:

id ANALYZED FROM *beg-c---end-c* TO *beg-m---end-m* PAGE*n*

where *id* is the identity of the analyzed program, and *beg-c, end-c, beg-m,* and *end-m* are parameters of the ANALYZE control instruction. Page *n* represents the page number of the printed output.

Output Listing

The analyzed output listing is arranged in three vertical columns of 54 lines each. Each item is sorted and listed by referenced memory location, the location of the instruction, and the instruction itself. If a word is referenced as a full word and as a half word, the corresponding output is printed in the order: full-word reference, left-half reference, and right-half reference. All index register references are listed at the end of the output. Data on the final page of the listing is evenly distributed among the three columns.

Ending Statement

The ending statement is printed as: END OF ANALYZER.

If there was too much input data for the ANALYZER routine to process, the following ending statement will appear:

PROGRAM EXCEEDED ANALYZER MEMORY CAPACITY

REMARKS

1. Processing of each ANALYZE instruction is completed before the next instruction is executed.
2. Separate outputs are written for each ANALYZE control instruction executed.

CONSOLE TYPEWRITER NORMAL TYPE-OUT

NO.	TYPE-OUT	EXPLANATION
1	*format* ANALYZE *id*	A program with a format of *format* and an ID of *id* is about to be analyzed.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	*id* MISSING	An RPL or ABS program specified by *id* was not encountered during a forward and backward search of Tape *t*. The current ANALYZE control instruction is ignored.

CONTINUED

CONSOLE TYPEWRITER ERROR TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
2	CONTROL LINE ERROR	One of the following errors was detected: • *format* is not RPL or ABS. • The command is not ANALYZE or ENDALL. • *t* is greater than 15. • *t* is 3, 5, 6, or 8. • *end-c* is less than *beg-c*. • *end-m* is less than *beg-m*. • Six parameters are not present in the Address and Remarks field. ANALYZER is immediately terminated and control is transferred to 1XCONER (page 4-18).
3	BAD SECTION WORD	A format error was detected in the RPL input, and control is transferred to 1NXTCRD (page 4-19).
4	CHKSUM	An ABS input card does not have a proper checksum, but it is processed and operation is resumed.
5	ILLEGAL CARD	A card in an ABS input deck is not in ABS format, and control is transferred to 1XCONER if input is from Tape 8; otherwise, control is transferred to 1NXTCRD.
6	TAPE *t* WR RING	No Write Ring is on Tape *t*. To continue, the operator should insert a Write Ring and press the Advance Bar.
7	TAPE *t* IN LOCAL	Tape *t* is in local operation. To continue, the operator should place Tape *t* in automatic mode and press the Advance Bar.
8	TAPE *t* ROCKED 5	A parity or sprocket error persisted on Tape *t* during five successive retries by ANALYZER. The operator may press the Advance Bar to continue retrying.
9	TAPE *t* FAULTY	A non-recoverable tape error occurred on Tape *t*. The IOP Fault Register will indicate the type of fault. The job must be terminated at this point.

BINDEL SERVICE ROUTINE

FUNCTION

Deletes RPL, ABS, and REL programs from the User's Program Tape (Tape 4).

SERVICE ROUTINE INITIATION

The BINDEL service routine may be called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,BINDEL,GO
		ABS } }	*, , GO (Binary deck of BINDEL)

INPUT-OUTPUT SYSTEM

BINDEL uses IOPS and PROC, two Philco input-output systems, for all reads, writes, and tape movements. PROC uses the TAC subroutine TAPER for error processing. (Refer to Program Report 14, Tape Error Routine.)

SERVICE ROUTINE CONTROLS

Three controls are used during the operation of BINDEL: (1) identification of programs to be deleted, (2) a toggle preset option to indicate where the output listing is to be placed, and (3) a standard control instruction, END, which indicates termination of service routine action.

Identification of Programs to be Deleted

Programs to be deleted from Tape 4 are designated by their ID's. These are written in the Address and Remarks field of succeeding instruction lines immediately following the RPL control instruction which initiates BINDEL. One program ID is written per line. Nothing may be written in the Command field of the line, or the entire line will be ignored. As many as 500 ID's, arranged in any sequence, may be listed.

Pre-Set Toggle Option

An output listing of the ID's of programs remaining on the updated tape, as well as a listing of ID's of deleted programs, is always placed on Tape 5 for off-line printing on the High-Speed Printer. This listing may also be placed on the Console Typewriter if Toggle Switch Zero is set to the ON position prior to execution of the BINDEL routine. When Normal Type-Out 1 (page 5-4.2) occurs, the computer halts and Toggle Zero may be set if desired.

Control Instruction

The BINDEL service routine accepts only one control instruction, END, which terminates action of the routine and returns control to SYS. END is written in standard SYS control line format and is comparable to the ENDALL control instruction associated with most other service routines.

ACTION

When the BINDEL service routine is initiated, the following action takes place:

- DELETION ROUTINE FLEXO? is typed on the Console Typewriter (refer to Console Typewriter Normal Type-Out 1, page 5-4.2). and the computer halts. The operator may then set Toggle Switch Zero to the ON position if an output listing is desired on the Console Typewriter.
- After the operator presses the Advance Bar, the contents of Tape 4 are copied onto Tape 6 and both tapes are rewound.
- The contents of Tape 6, except for those programs to be deleted, are copied onto Tape 3.
- The updated version of the User's Program Tape is then copied from Tape 3 to Tape 4, and both tapes are rewound.
- A listing of the ID's of all programs on the updated tape and of the deleted programs is then placed on Tape 5 for off-line printing on the High-Speed Printer. If Toggle Switch Zero is set, the listing will also appear on the Console Typewriter. (Refer to Output Format, page 5-4.2.)
- After the listings are completed, DELETION FINISHED, 6 IS BACKUP will be typed on the Console Typewriter.

HIGH-SPEED PRINTER FORMAT

The program output is edited and transferred to Tape 5 for printing on the High-Speed Printer in Data Select 0. The printed page output from the High-Speed Printer appears in the following formats:

	FORMAT	EXPLANATION
1	Heading Print-Out	REMAINING PROGRAMS *date & time* *
	Body Contents	Program identities, preceded by their respective program block counts, are listed vertically in three columns of 53 rows each per page. The listing is in the sequence of programs contained on the updated user's program tape (Tape 4).
	Ending Print-Out	END REMAINING TOTAL BLOCKS *n* were *n* is the decimal number of blocks on the user's program tape.
2	Heading Print-Out	DELETED PROGRAMS
	Body Contents	Program identities are listed vertically in four columns of 53 rows each per page. Identities are listed in order of deletion.
	Ending Print-Out	END DELETED

* The date and time will appear only if an Accounting Clock is present.

HIGH-SPEED PRINTER FORMAT (Continued)

	FORMAT	EXPLANATION
3	Heading Print-Out	PROGRAMS NOT HERE
	Body Contents	Program identities of those programs which were to be deleted but were not found on the user's program tape are listed vertically in a single column.
	Ending Print-Out	END DELETION REPORT

CONSOLE TYPEWRITER OUTPUT FORMAT

If Toggle Switch Zero is in the ON position, the following type-outs may occur:

1	REMAINING	(tab) (tab)	DELETED
	(List of ID's of programs remaining, without the block count.)		(List of ID's of programs deleted.)
2	PROGRAMS NOT HERE		
	(List of ID's of programs to be deleted which were not found on the User's Program Tape.)		

REMARKS

An asterisk before an ID in the output of either the High-Speed Printer and/or the Console Typewriter denotes the program is in ABS or REL format.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	DELETION ROUTINE FLEXO?	Enables the operator to set Toggle Switch Zero to ON, if output via the Console Typewriter is desired. The Advance Bar should be pressed to continue computer operation.
2	DELETION FINISHED, 6 IS BACKUP	The action of service routine BINDEL is completed, and a copy of the original programs on Tape 4 has been placed on Tape 6. The Advance Bar should be pressed to obtain the next job.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	CONTROL LINE ERROR	More than 500 identity cards were detected. BINDEL operation is terminated and control is transferred to 1XCONER (page 4-18).
2	JOB CARD INTERCEPTED	A JOB control line was detected prior to the appearance of an END control instruction. BINDEL operation is terminated and a post-mortem dump is executed (refer to 1ERRDMP, page 4-4). The intercepted JOB card will be executed.

EXAMPLE

```
L  Location  Command  Address and Remarks
             RPL      1,BINDEL,GO
                      ALPHA
                      BETA
                      GAMMA
             END
```

Explanation: Execution of these instructions causes programs ALPHA, BETA, and GAMMA to be deleted from Tape 4.

DATA SERVICE ROUTINE

FUNCTION

Transfers data required by a user's program from the SYS input tape (Tape 8) onto a specified tape. At the option of the user, the routine may also convert Image Mode input to Code Mode output and/or alter card format.

SERVICE ROUTINE INITIATION

The DATA service routine may be called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,DATA,GO
		ABS }	*,,GO (Binary deck of DATA)

INPUT-OUTPUT SYSTEM

DATA uses XORD, a Philco input-output subroutine, for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

DATA accepts two control instructions: TAPE and ENDDATA. The instructions are written in standard SYS control line format. Illegal parameters cause the routine to be terminated and control to be transferred to 1XCONER (page 4-18).

TAPE

Specifies the data transfer to be performed by the DATA service routine, and must be immediately followed by the data to be transferred.

Format

L	Location	Command	Address and Remarks
		TAPE	*t,mode,words,cards*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — which indicates the tape onto which the data is to be transferred. Any other number will cause Error Type-Out 2 (page 5-5.3) to occur.

mode (OPTIONAL). The mode in which the input data is to be transferred onto the output tape. One of the following four options may be selected:

CODE or HOLL	Indicates that data is to be transferred to Tape *t* in Code Mode. Input from Tape 8 may be in either Code or Image Mode.
IMAGE or BINARY	Indicates that data is to be transferred to Tape *t* in Image Mode. Input from Tape 8 must be in Image Mode. If Tape 8 is in Code Mode and IMAGE or BINARY is specified, Error Type-Out 3 (page 5-5.3) will occur.

If *mode* is omitted, input data will be transferred to Tape *t* in Code Mode, 10 words per card 12 cards per block. If *mode* is omitted, *words* and *cards* must also be omitted and *t* must be followed by a dollar sign ($). Blank columns on cards are transcribed onto Tape *t* as zeroes if the transfer is in Image Mode, or as spaces (60_8) if the transfer is in Code Mode.

words (OPTIONAL). A decimal number which indicates the number of words per card to be transferred to the output tape. If *mode* is CODE or HOLL, words may be any number from one through 10; if *mode* is IMAGE or BINARY, *words* may be any number from one through 20. If *words* is specified as a parameter, the *cards* parameter is required.

cards (OPTIONAL). A decimal number, one through 128, which indicates the number of cards per block to be transferred to the output tape.

The product of *words* times *cards* must be equal to or less than 128_{10}, or a Control Line Error will occur.

Notes:

1. This parameter is required if the parameter *words* is specified.
2. Blank columns on a card are transcribed onto Tape *t* as zeroes if the transfer is in Image Mode, or as spaces (60_8) if the transfer is in Code Mode.
3. If the number of cards supplied as input data is not sufficient to meet the cards-per-block requirement for the final block, cards of all zeroes or spaces will be provided for transfer to the output tape until the final block has been filled with the specified number of cards per block.

4. The remaining words in each block, determined by 128 minus *words* times *cards* will contain filler characters (32_8).

5. If both *words* and *cards* are omitted, input data will be transferred to Tape *t* as: 10 words per card, 12 cards per block if *mode* is CODE or HOLL; 20 words per card, six cards per block if *mode* is IMAGE or BINARY.

Remarks

1. The TAPE control instruction must be the first card following the call for DATA; otherwise, Error Type-Out 1 (page 5-5.3) will occur.

2. Tape *t*, to which data is to be transferred, is not rewound by the DATA routine either before or after it is used.

3. Input data is transcribed onto Tape *t* starting at the beginning of the next block.

Example

L	Location	Command	Address and Remarks
		TAPE	9,HOLL,5,25

Explanation: Execution of this instruction causes the DATA service routine to transfer data from Tape 8 onto Tape 9 in Code Mode, five words per card, 25 cards per block. The data on Tape 8 may be in either Code or Image Mode.

ENDDATA

Terminates the DATA service routine by writing the final output buffer block of data onto Tape *t*, and causes control to be returned to SYS.

Format

L	Location	Command	Address and Remarks
		ENDDATA	

Parameters

None

Remarks

Operation of the DATA routine may be resumed at this point, if additional input data is to be transferred. Re-entry is accomplished by a "JMP *" instruction following ENDDATA. The "JMP *" must be immediately followed by a new TAPE control instruction.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	TAPE NOT RECOG-NIZED	The first card supplied to the DATA routine does not contain the word TAPE in the Command field. Control is transferred to 1XCONER (page 4-18).
2	TAPE SELECTED FOR DATA ILLEGAL	A decimal number other than 0, 3, 6, 7, or 9 through 15 was specified as Tape *t*. Control is transferred to 1XCONER.
3	INPUT TAPE IS CODE, CANNOT TRANSFER IN IMAGE	An illegal conversion request was made by the TAPE instruction. The *mode* parameter is BINARY or IMAGE where input data is in Code Mode. Control is transferred to 1XCONER.
4	JOB CARD INTER-CEPTED	A JOB control instruction was detected following a TAPE instruction and prior to detecting an ENDDATA instruction. Control is transferred to 1ERRDMP and the intercepted JOB card is executed.
5	TAPE *t* WR RING	There is no Write Ring on Tape *t*. To continue, the operator may insert a Write Ring and press the Advance Bar.
6	TAPE *t* IN LOCAL	Tape *t* is in local operation. To continue, the operator should place the tape in automatic mode and press the Advance Bar.
7	TAPE *t* ROCKED 5	A parity, sprocket, or space error occurred on Tape *t* during five successive retries. The operator should press the Advance Bar to continue trying to correct the error.
8	TAPE *t* FAULTY	A non-recoverable error was encountered on Tape *t*. Further action depends upon the operator.

EXAMPLE

```
L  Location  Command   Address and Remarks

             REWIND    0,9              $ (1)
             RPL       1,DATA,GO        $ (2)
             TAPE      0,IMAGE,16,8     $ (3)
               •                          (4)
               •       (Data to be transcribed)
               •
             ENDDATA                    $ (5)
             JMP       *                $ (6)
             TAPE      9,CODE,4,30      $ (7)
               •                          (8)
               •       (Data to be transcribed)
               •
             ENDDATA                    $ (9)
```

Explanation — Execution of these instructions causes the following action to take place:

1. Tapes 0 and 9 are rewound to receive data from the input tape.
2. Service routine DATA is called into operation.
3. Tape 0 is specified as the tape onto which the Image Mode data is to be transferred in Image Mode, 16 words per card, eight cards per block.
4. The data to be transferred is processed as designated in step (3) of the explanation.
5. ENDDATA terminates the service routine action as determined by the first TAPE instruction. Control is returned to SYS.
6. The "JMP *" instruction causes return of control to the DATA service routine.
7. Tape 9 is the specified tape onto which the following data is to be transferred. This transfer is to be in Code Mode, four words per card, 30 cards per block.
8. The data to be transferred is processed as designated in step (7) of the explanation.
9. ENDDATA terminates the action of the second TAPE instruction. Control is returned to SYS.

RPLC (SYSRPLC) SERVICE ROUTINE

FUNCTION

Revises a specified tape by adding, deleting, or correcting designated RPL programs contained on that tape.

SERVICE ROUTINE INITIATION

The SYSRPLC service routine may be called into operation by any of the following three instructions:

L	Location	Command	Address and Remarks
		RPLC	
		RPL	1,SYSRPLC,GO
		ABS	*,,GO
		}	(Binary deck of SYSRPLC)

INPUT FORMAT

Control instructions within SYSRPLC cause all information contained on designated tapes to be copied and/or skipped. The *id* parameter within any of these instructions, however, must designate an RPL program.

INPUT-OUTPUT SYSTEM

SYSRPLC uses its own input-output system, similar to the MAGTAPE subroutines (pages 4-9 to 4-11), for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

SYSRPLC accepts the following ten control instructions, which are written in standard SYS control line format:

INSTRUCTION	FUNCTION
NEWTAPE	Specifies the tape onto which the updated information is to be written.
REWIND	Rewinds magnetic tapes.
COPY	Transfers information to the output tape specified by NEWTAPE.
COPYTIL	Transfers information to the output tape until a designated RPL program is encountered.

CONTINUED

CONTROL INSTRUCTIONS (Continued)

INSTRUCTION	FUNCTION
SKIPTIL	Bypasses all information on an input tape until a designated RPL program is encountered.
DELETE	Copies all information until a designated RPL program is encountered, and then bypasses the program.
ADD	Bypasses all information until a designated RPL program is encountered, and then copies that program onto the output tape.
CORRECT	Copies all information on an input tape until a designated RPL program is encountered, makes corrections to that program, and copies it onto the output tape.
IDCHANGE	Performs functions identical to CORRECT, and also permits the changing of a program ID.
ENDALL	Terminates the SYSRPLC routine and returns control to SYS.

Illegal parameters generally cause SYSRPLC to be terminated and control to be transferred to 1XCONER (page 4-18).

If eight or fewer characters are specified as the program identity parameter for any of the control instructions within SYSRPLC, only the first eight characters of the ID of the program on tape will be compared during a search. (Refer to ONE-WORD SEARCH, page 3-4).

Where designated programs cannot be located, NOT HERE is typed on the Console Typewriter, and control is transferred to 1ERRDMP (page 4-4).

Any reference to an output tape is assumed to be the tape designated by the NEWTAPE control instruction.

Immediately following initiation of the SYSRPLC service routine, RPLC is typed on the Console Typewriter.

NEWTAPE

Specifies the magnetic tape onto which the updated information is to be placed.

Format

L	Location	Command	Address and Remarks
		NEWTAPE	*t*

Parameter

t A decimal number — 0, 3, 6, 7 or 9 through 15 — which indicates the output tape.

Action

When the NEWTAPE instruction is executed, all control instructions within the SYSRPLC service routine are preset to have the output of each instruction placed on the same final output tape. SYSRPLC then requests its next control instruction.

Remarks

1. NEWTAPE must precede all other control instructions within SYSRPLC, with the exception of the REWIND instruction. If neither is encountered as the first instruction, Error Type-Out 1 (page 5-6.10) will occur.

2. Only one NEWTAPE instruction may be issued for each SYSRPLC call.

REWIND

Rewinds one or more magnetic tapes without lockout.

Format

L	Location	Command	Address and Remarks
		REWIND	*t,...,t*

Parameters

t One or more decimal numbers, 0 through 15, which indicate the magnetic tapes to be rewound.

Action

When the REWIND control instruction is executed, all tapes indicated by *t,...,t* are rewound without lockout. The next RPLC control instruction is then requested.

COPY

Transfers without change all information from a designated tape onto the output tape.

Format

L	Location	Command	Address and Remarks
		COPY	*t,sent*

Parameters

t A decimal number, 0 through 15, which indicates the tape from which the contents are to be copied.

sent (OPTIONAL). Eight or fewer alphanumeric characters which compose the first 120 words of a sentinel block. The parameter is used to indicate to SYSRPLC that all programs from Tape *t* are to be copied until the sentinel block designated by *sent* is encountered. All characters except break characters are acceptable. Spaces are ignored. If fewer than eight characters are used, the sentinel will be right-justified and left-filled with zeroes. If more than eight characters are used, the last eight characters (excluding spaces) are used as the sentinel. If both the comma and sentinel parameter are omitted, the normal sentinel block of Z's is assumed. If a comma is written and the sentinel is omitted, a sentinel of all zeroes is assumed.

Action

When the COPY control instruction is executed, the following action takes place:

- All information on Tape *t* is copied onto the output tape until a sentinel block specified by *sent* is encountered.
- If *sent* is omitted, a sentinel block of Z's will terminate the copy action.
- When the sentinel block is encountered, it is bypassed and Tape *t* is positioned immediately following the sentinel block.
- SYSRPLC then requests its next control instruction.

Example

```
L   Location   Command   Address and Remarks

               COPY      9,AAAAAAAA
```

Explanation: Execution of this instruction causes all information on Tape 9 to be copied onto the output tape until a sentinel block of A's is encountered.

COPYTIL

Transfers without change all information from a designated tape onto the output tape until a specified RPL program is encountered.

Format

L	Location	Command	Address and Remarks
		COPYTIL	*t,id*

Parameter

t A decimal number, 0 through 15, which indicates the tape whose contents are to be copied.

id 16 or fewer alphanumeric characters which provide the complete identity of an RPL program to be located. Spaces are significant.

Action

When the COPYTIL instruction is executed, the following action takes place:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is copied onto the output tape until an RPL program designated by *id* is encountered.
- Upon completion of the copy operation, Tape *t* is positioned at the start of the RPL program designated by *id*, and the next SYSRPLC control instruction is requested.

Remarks

The RPL Program *id* is not copied.

SKIPTIL

Causes all information on a designated tape to be bypassed until a specified RPL program is encountered.

Format

L	Location	Command	Address and Remarks
		SKIPTIL	*t,id*

Parameters

The parameters are the same as those of the COPYTIL control instruction (page 5-6.3).

Action

When the SKIPTIL control instruction is executed:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is passed over until the RPL program designated by *id* is encountered.
- Upon location of the *id*, Tape *t* is positioned at the beginning of that program and the next SYSRPLC control instruction is requested.

DELETE

Copies all information from a designated tape onto the output tape until a specified RPL program is encountered. That program is then passed over.

Format

L	Location	Command	Address and Remarks
		DELETE	*t,id*

Parameters

The parameters are the same as those of the COPYTIL control instruction (page 5-6.3).

Action

When the DELETE control instruction is executed:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is copied onto the output tape until the RPL program designated by *id* is encountered.
- Program *id* is bypassed and Tape *t* is positioned at the end of the program.
- The next SYSRPLC control instruction is then requested.

ADD

Causes all information on a designated tape to be bypassed until a specified RPL program is located. That program is then copied onto the output tape.

Format

L	Location	Command	Address and Remarks
		ADD	*t,id*
		ADD	*,*id*

Parameters

The *t* and *id* parameters are the same as those of the COPYTIL control instruction (page 5-6.3).

The parameter * may be used in place of *t* if an ABS program deck immediately follows the instruction, and the deck is to be converted into RPL format and written onto NEWTAPE. If a dummy PMAX card heads the ABS deck, *id* will be ignored, and the new program ID will be obtained from the dummy PMAX card. If no dummy PMAX card is present, the program ID will be taken from the *id* parameter.

If both the dummy PMAX card and the *id* parameter are not present, ADD * WITH NO ID will be typed on the Console Typewriter and control will be transferred to 1XCONER. If the input is not in image mode, control will be transferred to 1XCONER.

Action

When the ADD control instruction is executed:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is bypassed until the RPL program designated by *id* is encountered.
- Program *id* is copied onto the output Tape and Tape *t* is positioned at the end of the program.
- RPLC then requests its next control instruction.

CORRECT

Copies all information from a designated tape onto the output tape until a specified RPL program is located. Corrections are made to that program and it is also copied onto the output tape.

Format

L	Location	Command	Address and Remarks
		CORRECT CORRECT	*t,id* *, id

Parameters

The parameters are the same as those of the COPYTIL control instruction (page 5-6.3).

Action

function of the * parameter
s same as that for the ADD*
ction, above, and the identical
typeout will occur if both
ummy PMAX card and the id
ameter are not present.

When the CORRECT instruction is executed:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is copied onto the output tape until the RPL program designated by *id* is encountered.
- This RPL program is copied onto the output tape, with the corrections (which follow the CORRECT instruction) inserted as separate RPL sections at the end of the RPL. (These sections will overlay the proper locations at load time to provide the necessary corrections.)

- Tape *t* remains positioned at the end of the program.

- SYSRPLC then requests its next control instruction.

Correction Format

Corrections to the RPL program designated by *id* appear immediately after the CORRECT control instruction. These corrections specify the location of the half-word to be corrected, and the new value to be inserted into the half-word. One correction per line is permitted. An END instruction, written in the Command field, must be used to terminate the correction procedure.

Corrections are written as follows:

L	Location	Command	Address and Remarks
p	$addr_1$	*instr*	$addr_2$

The parameters are written as:

p An alphabetic character, L or R, which indicates whether the change is to start in the left or right half of the word specified by $addr_1$.

L (or blank)	Specifies the left instruction
R	Specifies the right instruction

$addr_1$ An octal address, written in the Location field, which indicates the location of the first word to be changed. If consecutive instructions or locations are to be corrected, the location of only the first need be specified.

instr The new word constant (W/), octal constant (O/), or command part of a new instruction is written in the Command field. The command is written as a standard TAC mnemonic command. Word constants may be eight or fewer alphanumeric characters. If fewer than eight characters are used, the constant is assumed to be left-justified and right-filled with spaces. Octal constants must contain exactly 16 digits. If more or fewer are specified, a Control Line Error (page 4-18) will result.

addr$_2$ The address portion of a new instruction written in octal in the Address and Remarks field. This parameter must be written in octal for <u>all</u> instructions, including shifts and repeats. It may be index register-modified in the usual manner — using a comma followed by the decimal designation of the register.

Remarks:

1. Constants as corrections do not require the parameter *p*, because it is assumed the entire word is changed. If an L is specified as *p* with a constant, it will be ignored. If an R is specified, Error Type-Out 6 (page 5-6.10) will occur.

2. Corrections may be in any sequence, if the *addr*$_1$ parameter is specified for each correction. If consecutive instructions are to be corrected, however, only the first requires the parameter *addr*$_1$.

3. A double repeat mnemonic (Philco 212 instruction) must have at least three modifiers associated with it. If only 3 modifiers are given, N is assumed to complete the command and is regarded as the first modifier, eg., DRASA is assumed to be the mnemonic DRNASA.

Example

```
L  Location  Command   Address and Remarks

             CORRECT   9,JOE
   7472      TMD       13065
R  10344     JAED      7546,5
             END
```

Explanation: Execution of these instructions causes all information on Tape 9 to be copied onto the output tape until RPL program JOE is encountered. When located, JOE is copied onto the output tape, and the two instructions, TMD and JAED, are inserted as separate sections at the end of JOE. These sections will overlay locations 7472_8 and $10344R_8$, respectively, at load time. SYSRPLC then requests its next control instruction.

IDCHANGE

Provides a new ID for a designated RPL program and performs the identical functions of CORRECT.

Format

L	Location	Command	Address and Remarks
		IDCHANGE	*t,id*

Parameters

The parameters are the same as those of the COPYTIL control instruction (page 5-6.3).

MARCH 1964

Action

When IDCHANGE is executed:

- All information on Tape *t*, beginning at the point where *t* is currently positioned, is copied onto the output tape until the RPL program designated by *id* is encountered.

- The first correction following the IDCHANGE instruction must be the new ID. An I is written in the Label column and the new ID is written in the Command and Address and Remarks fields as follows:

L	Location	Command	Address and Remarks
I		*new id*	

- The action for any subsequent corrections is the same as that for the CORRECT instruction.

- SYSRPLC then requests its next control instruction.

Remarks

The new ID designated by *new id* may be 16 or fewer alphanumeric characters. Spaces are significant and no break characters are permitted.

Example

```
L  Location  Command   Address and Remarks

             IDCHANGE 12,PROGRAM1
I            PROGRAM2
             END
```

Explanation: Execution of these instructions causes all information on Tape 12 to be copied onto the output tape until RPL program PROGRAM1 is encountered. When located, the old ID is replaced by *new id*, and PROGRAM1 is copied onto the output tape. If there had been any other corrections, they would have been treated in the same manner as effected by the CORRECT control instruction. SYSRPLC then requests its next control instruction.

ENDALL

Terminates the action of the SYSRPLC service routine and returns control to SYS.

L	Location	Command	Address and Remarks
		ENDALL	*finis*

Parameters

finis (OPTIONAL). An ending parameter, specified either as LIST or *sent.*

LIST	Indicates that a list of the ID's of all RPL programs transferred to the output tape is to be typed on the Console Typewriter.
sent	Eight or fewer alphanumeric characters (except LIST) which compose the 128 words of a sentinel block. The block is written on the output tape.

Action

When ENDALL is executed, the following action takes place:

- A sentinel block of Z's, or the configuration specified by the *sent* parameter, if it is used, is written on the output tape.
- All tapes used by the SYSRPLC routine are rewound.
- If LIST is specified as *finis*, a listing of the ID's of all RPL programs transferred to the output tape is typed on the Console Typewriter.
- TOTAL BLKS COPIED *n* is then typed on the Console Typewriter, where *n* indicates the total number of blocks copied, excluding the final sentinel block.
- Control is returned to SYS.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	RPLC	The SYSRPLC routine has been initiated. No operator action is necessary.
2	TOTAL BLKS COPIED *n*	The SYSRPLC routine has been terminated and the total number of blocks transferred to the output tape is indicated by *n*.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	NEWTAPE CARD NOT SOON ENOUGH	A control instruction other than a REWIND was encountered prior to a NEWTAPE instruction. Control is transferred to 1XCONER (page 4-18).
2	NOT HERE	A program designated by *id* cannot be found. SYSRPLC operation is terminated and control is transferred to 1ERRDMP (page 4-4).
3	UNIT NOT AVAILABLE	A designated magnetic tape is not available, or there is no Write Ring on the output tape. The operator may correct the condition and press the Advance Bar to continue.
4	BAD TAPE	A non-recoverable error was detected on a designated tape. All tapes used by SYSRPLC are rewound and the program must be manually terminated. This type-out may occur if a sentinel block is not encountered.
5	BAD SECTION WORD	The input tape is not in acceptable RPL format. The action is the same as that for Error Type-Out 4.
6	BAD CARD # *n*	A correction instruction is illegal, where *n* is the sequence number of the instruction, as counted by the SYSRPLC routine, starting from one with each correct instruction. This correction instruction is ignored. Each time the Advance Bar is pressed, the remaining correction instructions are typed out until a new octal location is encountered in the Location field of an instruction, or a new SYSRPLC instruction is encountered. Normal action will be continued if the Advance Bar is pressed again.
7	ADD * WITH NO ID	Both an *id* parameter of an ADD * control instruction *and* a dummy PMAX card heading the ABS card deck following the ADD * instruction are missing. Control is transferred to 1XCONER.
8	BAD CHECKSUM - - - - - sequence	The checksum on one of the cards in the ABS program deck following an ADD * ~~control instruction~~ [illegible] does not agree with the computed sum. (The sequence parameter indicates the sequence number of the card with the incorrect checksum. Control is transferred to 1XCONER.

SYSRPLC SERVICE ROUTINE EXAMPLE

```
L   Location   Command    Address and Remarks
               RPLC                           $ (1)
               REWIND     9,10,11,12          $ (2)
               NEWTAPE    9                   $ (3)
               COPYTIL    10,HOMER            $ (4)
               ADD        11,ALPHA            $ (5)
               CORRECT    12,GAMMA            $ (6)
L   4063       TMA        1075                $ (7)
R   5321       TDXL       4,5                 $ (8)
               END                            $ (9)
               COPY       10,AAAAAAAA         $ (10)
               ENDALL     LIST                $ (11)
```

Explanation — Execution of these instructions causes the following action to take place:

1. SYSRPLC is called into operation.
2. Tapes 9, 10, 11, and 12 are rewound.
3. Tape 9 is specified as the final output tape.
4. All information on Tape 10 is copied onto Tape 9 until RPL program HOMER is encountered. Tape 10 remains positioned at the beginning of program HOMER.
5. All information on Tape 11 is bypassed until RPL program ALPHA is encountered. ALPHA is copied onto Tape 9. Tape 11 remains positioned at the beginning of the next block past ALPHA.
6. All information on Tape 12 is copied onto Tape 9 until RPL program GAMMA is encountered.
7. GAMMA is copied onto Tape 9, and the TMA instruction is inserted as a separate RPL section at the end of GAMMA. At load time it will overlay location $4063L_8$ of program GAMMA.
8. The TDXL instruction is inserted as a separate RPL section following the RPL section referred to in step 7. At load time, it will overlay location $5321R_8$ of program GAMMA.
9. The END instruction signifies the end of corrections.
10. All information on Tape 10, beginning with RPL program HOMER, is copied onto Tape 9, until a sentinel block of A's is encountered.
11. A sentinel block of Z's is written on Tape 9 and Tapes 9, 10, 11, and 12 are rewound. The SYSRPLC routine is then terminated with the typing on the Console Typewriter of a list of the ID's of all programs copied onto Tape 9.

SYSTRACE SERVICE ROUTINE

FUNCTION

Monitors all or a portion of a running program and provides edited output of selected program instructions which are positioned within, or whose addresses access, a designated area of memory. A printed output for these instructions is produced, which consists of each instruction's location, its command and address, and the settings of all addressable registers and console switches taken just after each instruction is executed.

Any number of traces may be requested per SYSTRACE call. The program or designated portion is then run and traced separately for each trace requested.

SERVICE ROUTINE INITIATION

The SYSTRACE service routine may be called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,SYSTRACE,GO,*addr*
		ABS }	*,, GO, *addr* (Binary deck of SYSTRACE)

where *addr* is an octal number which defines the first of the 2074_8 consecutive memory locations allocated to the SYSTRACE routine. These memory locations must be outside the area used by the program being traced, as well as outside the area used by SYS (locations 0 through 1000_8).

INPUT PROGRAM REQUIREMENTS

The program to be traced must be loaded prior to the SYSTRACE service routine.

INPUT-OUTPUT SYSTEM

SYSTRACE contains its own input-output system for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

The SYSTRACE service routine accepts two control instructions: *id,* which specifies the parameters for each trace, and ENDALL. ENDALL terminates the SYSTRACE routine and returns control to SYS. The instructions are written in standard SYS control line format. Illegal parameters cause SYSTRACE to return control immediately to SYS.

id

Provides a parameter statement for each trace to be performed, consisting of an ID to be printed identifying each trace, and information specifying the type of trace to be executed. Any number of *id* control instructions can be accepted.

Format

L	Location	Command	Address and Remarks
		id	*start,mth,end,nth,low,high,entrance*

Parameters

id Eight or fewer alphanumeric characters which indicate the ID of a particular trace. Spaces are significant. Neither break characters nor the configuration ENDALL is acceptable. The ID is used as part of the title for each page of output.

start Five or fewer octal digits which indicate the location of the trace trap. The trap is set in the program to be traced by moving the left instruction of *start* into the SYSTRACE routine. In its place is substituted a transfer of control to provide linkage between the two programs. If *start* is not encountered, tracing will not commence and the user's program will run until its normal termination.

mth Eight or fewer octal digits which specify that the Mth time the left instruction in location *start* is to be executed, tracing is to commence. If zero is specified as *mth,* or if this instruction is not performed *mth* times, tracing will not commence and the user's program will run until its normal termination.

end Five or fewer octal digits which indicate the exit location for the trace routine of a particular *id* instruction. The left instruction is used as the exit. If the tracing routine has begun but *end* is not encountered, the trace will loop continuously until the operator manually terminates the routine.

nth Eight or fewer octal digits which specify that the Nth time the left instruction in location *end* is to be executed after the trace has started, tracing is to cease. The next SYSTRACE control instruction is then immediately requested. If the tracing routine has begun and zero is specified as *nth,* the trace will loop continuously until the operator manually terminates the routine.

low Five or fewer octal digits which specify the lower boundary of a designated memory area, and which are used to determine those instructions of the user's program that are to be edited for SYSTRACE output. These instructions will be edited either if they are positioned within this memory area designated by *low* and *high*, or if they access a memory location within this same designated area. If *low* is greater than *high*, a Control Line Error will occur.

high Five or fewer octal digits which specify the upper boundary of the designated memory area as explained within *low*.

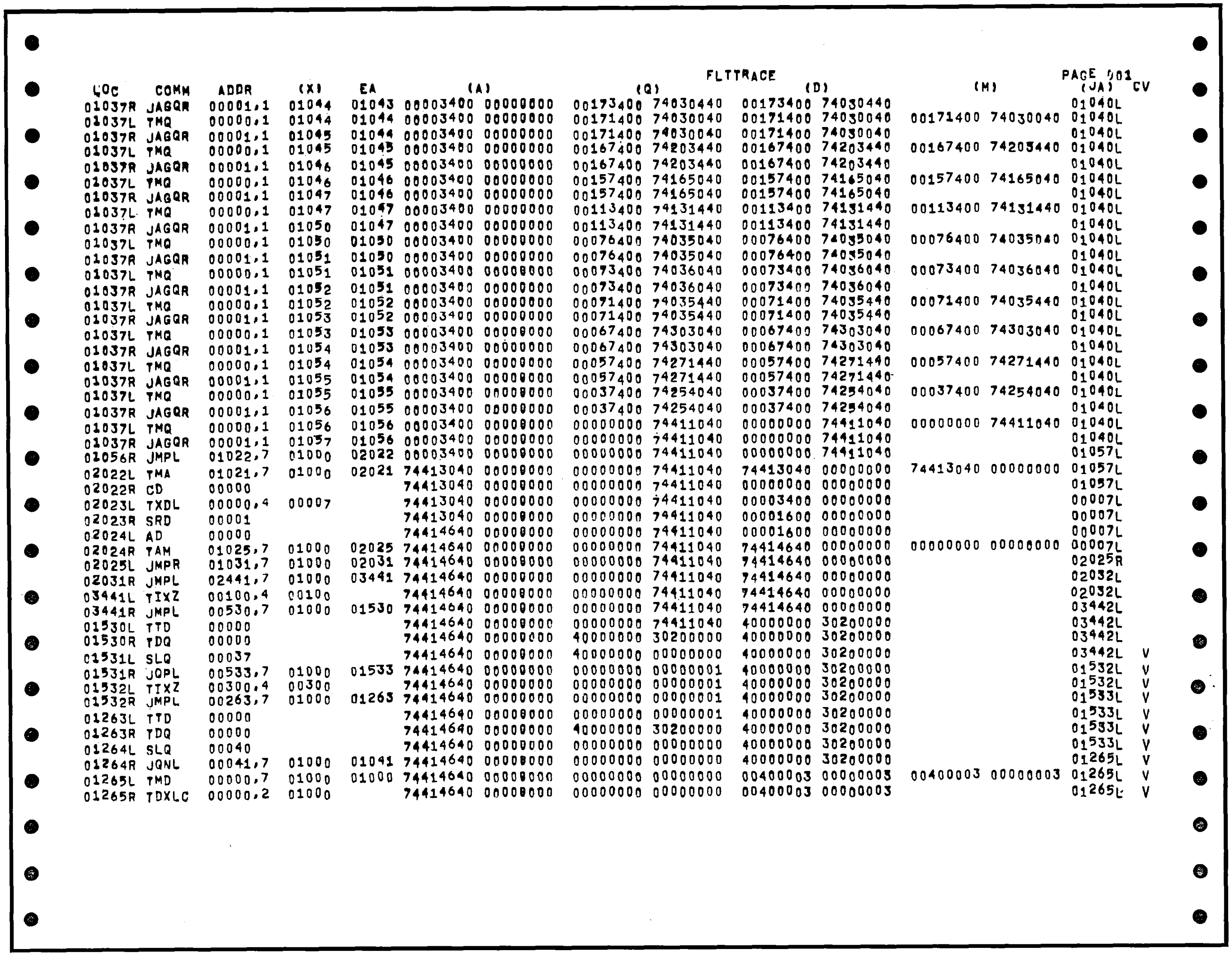
entrance Five or fewer octal digits which indicate the starting (entrance) address of the program to be traced for this particular trace. After SYSTRACE is loaded and the *id* parameters are interpreted to specify the required trace, control is transferred to location *entrance* of the first *id* control word, and the user's program runs under its own control.

ERROR FORMAT

Whenever a command fault or a HLT is executed (except as noted in the following paragraph), that instruction is automatically written on the output tape in quaternary format and the current trace is terminated. A new SYSTRACE control instruction is then requested. If a HLT had been executed, a press of the Advance Bar will request the next instruction.

If a command fault occurs because execution of a floating-point command was attempted in a Philco 210 or 211 without floating-point circuitry, the following action takes place:

- The instruction at fault will be located in the left instruction portion of the Program Register.
- The instruction in the right portion of the PR will be a NOP with the address of the floating-point instruction in its address field.
- To have the faulty line written as part of the output, the operator must manually transfer control the the left instruction in the memory location 1124 octal locations beyond the first memory location occupied by SYSTRACE.
- The current trace will then be terminated and a new SYSTRACE control instruction is requested.



```
                                                         FLTTRACE                                        PAGE 001
LOC    COMM   ADDR     (X)   EA          (A)                (Q)                (D)                (M)              (JA)  CV
01037R JAGQR  00001,1  01044 01043 00003400 00000000  00173400 74030440  00173400 74030440                    01040L
01037L TMQ    00000,1  01044 01044 00003400 00000000  00171400 74030040  00171400 74030040  00171400 74030040 01040L
01037R JAGQR  00001,1  01045 01044 00003400 00000000  00171400 74030040  00171400 74030040                    01040L
01037L TMQ    00000,1  01045 01045 00003400 00000000  00167400 74203440  00167400 74203440  00167400 74203440 01040L
01037R JAGQR  00001,1  01046 01045 00003400 00000000  00167400 74203440  00167400 74203440                    01040L
01037L TMQ    00000,1  01046 01046 00003400 00000000  00157400 74165040  00157400 74165040  00157400 74165040 01040L
01037R JAGQR  00001,1  01047 01046 00003400 00000000  00157400 74165040  00157400 74165040                    01040L
01037L TMQ    00000,1  01047 01047 00003400 00000000  00113400 74131440  00113400 74131440  00113400 74131440 01040L
01037R JAGQR  00001,1  01050 01047 00003400 00000000  00113400 74131440  00113400 74131440                    01040L
01037L TMQ    00000,1  01050 01050 00003400 00000000  00076400 74035040  00076400 74035040  00076400 74035040 01040L
01037R JAGQR  00001,1  01051 01050 00003400 00000000  00076400 74035040  00076400 74035040                    01040L
01037L TMQ    00000,1  01051 01051 00003400 00000000  00073400 74036040  00073400 74036040  00073400 74036040 01040L
01037R JAGQR  00001,1  01052 01051 00003400 00000000  00073400 74036040  00073400 74036040                    01040L
01037L TMQ    00000,1  01052 01052 00003400 00000000  00071400 74035440  00071400 74035440  00071400 74035440 01040L
01037R JAGQR  00001,1  01053 01052 00003400 00000000  00071400 74035440  00071400 74035440                    01040L
01037L TMQ    00000,1  01053 01053 00003400 00000000  00067400 74303040  00067400 74303040  00067400 74303040 01040L
01037R JAGQR  00001,1  01054 01053 00003400 00000000  00067400 74303040  00067400 74303040                    01040L
01037L TMQ    00000,1  01054 01054 00003400 00000000  00057400 74271440  00057400 74271440  00057400 74271440 01040L
01037R JAGQR  00001,1  01055 01054 00003400 00000000  00057400 74271440  00057400 74271440                    01040L
01037L TMQ    00000,1  01055 01055 00003400 00000000  00037400 74254040  00037400 74254040  00037400 74254040 01040L
01037R JAGQR  00001,1  01056 01055 00003400 00000000  00037400 74254040  00037400 74254040                    01040L
01037L TMQ    00000,1  01056 01056 00003400 00000000  00000000 74411040  00000000 74411040  00000000 74411040 01040L
01037R JAGQR  00001,1  01057 01056 00003400 00000000  00000000 74411040  00000000 74411040                    01040L
01056R JMPL   01022,7  01000 02022 00003400 00000000  00000000 74411040  00000000 74411040                    01057L
02022L TMA    01021,7  01000 02021 74413040 00000000  00000000 74411040  74413040 00000000  74413040 00000000 01057L
02022R CD     00000                74413040 00000000  00000000 74411040  00000000 00000000                    01057L
02023L TXDL   00000,4  00007       74413040 00000000  00000000 74411040  00003400 00000000                    00007L
02023R SRD    00001                74413040 00000000  00000000 74411040  00001600 00000000                    00007L
02024L AD     00000                74414640 00000000  00000000 74411040  00001600 00000000                    00007L
02024R TAM    01025,7  01000 02025 74414640 00000000  00000000 74411040  74414640 00000000  00000000 00000000 00007L
02025L JMPR   01031,7  01000 02031 74414640 00000000  00000000 74411040  74414640 00000000                    02025R
02031R JMPL   02441,7  01000 03441 74414640 00000000  00000000 74411040  74414640 00000000                    02032L
03441L TIXZ   00100,4  00100       74414640 00000000  00000000 74411040  74414640 00000000                    02032L
03441R JMPL   00530,7  01000 01530 74414640 00000000  00000000 74411040  74414640 00000000                    03442L
01530L TTD    00000                74414640 00000000  00000000 74411040  40000000 30200000                    03442L
01530R TDQ    00000                74414640 00000000  40000000 30200000  40000000 30200000                    03442L
01531L SLQ    00037                74414640 00000000  40000000 00000000  40000000 30200000                    03442L  V
01531R JQPL   00533,7  01000 01533 74414640 00000000  00000000 00000001  40000000 30200000                    01532L  V
01532L TIXZ   00300,4  00300       74414640 00000000  00000000 00000001  40000000 30200000                    01532L  V
01532R JMPL   00263,7  01000 01263 74414640 00000000  00000000 00000001  40000000 30200000                    01533L  V
01263L TTD    00000                74414640 00000000  00000000 00000001  40000000 30200000                    01533L  V
01263R TDQ    00000                74414640 00000000  40000000 30200000  40000000 30200000                    01533L  V
01264L SLQ    00040                74414640 00000000  00000000 00000000  40000000 30200000                    01533L  V
01264R JQNL   00041,7  01000 01041 74414640 00000000  00000000 00000000  40000000 30200000                    01265L  V
01265L TMD    00000,7  01000 01000 74414640 00000000  00000000 00000000  00400003 00000003  00400003 00000003 01265L  V
01265R TDXLC  00000,2  01000       74414640 00000000  00000000 00000000  00400003 00000003                    01265L  V
```

First Page of SYSTRACE Output from Program FLTTRACE

HIGH-SPEED PRINTER FORMAT

The output of the SYSTRACE service routine is written onto Tape 5 for off-line printing on the High-Speed Printer with Data Select 0. Each page of output is headed by a two-line title consisting of the particular trace ID and page number on the first line, and the following eleven columnar headings on the second line (in left-to-right order):

HEADING	MEANING
LOC	instruction location
COMM	instruction command
ADDR	instruction address
(X)	the contents of the index register after execution, if applicable
EA	the effective address when different from the instruction address
(A)	the contents of the A Register
(Q)	the contents of the Q Register
(D)	the contents of the D Register
(M)	the contents of the effective address before execution
(JA)	the contents of the JA Register after execution
CV	the status of the ICO and Overflow Indicators, designated by C if the ICO is set to ON, or by V if the Overflow is set to ON. A space (Δ) designates that an indicator was reset.

Following the headings are 60 lines of single-spaced output, each line representing a single, edited instruction. All addresses are in octal, index register references are in decimal, and the A, Q and D Registers as well as memory locations are in octal, unless a floating-point command is involved. In such a case, A, Q, D, and memory are in floating-point decimal form as indicated:

$$\pm n.nnnnnnnnn\pm eee$$

where the *n*'s represent a decimal integer and fraction, and the *e*'s represent a power of 10.

Refer to the figure on page 5-7.3 for an example of SYSTRACE output.

EXAMPLE

```
L  Location  Command   Address and Remarks

             JOB       TRACE TRACK                          $ (1)
             RPL       4,TRACK                              $ (2)
             RPL       1,SYSTRACE,GO,10000                  $ (3)
             TRACK1    1050,1,1060,1,1050,1055,1005         $ (4)
             TRACK2    2070,1,3005,4,2075,3000,1005         $ (5)
             ENDALL                                         $ (6)
```

Explanation — Execution of these instructions causes the following action to take place:

1. SYS is prepared for the next job, and JOB TRACE TRACK is typed on the Console Typewriter.

2. The RPL program TRACK is loaded into memory from Tape 4.

3. The RPL program SYSTRACE is loaded from Tape 1 into memory at location 10000_8, and control is transferred to the starting address of SYSTRACE.

4. The parameters of TRACK1 set SYSTRACE to trace, from the first time through location 1050_8 until the first time through location 1060_8. Only those instructions which either are positioned within or access memory within locations 1050_8 and 1055_8 are edited for output. Control is then transferred to program TRACK at location 1005_8. TRACK then runs under its own control until location 1050_8 is encountered, initiating the trace routine where the tracing specified by the parameters of TRACK1 is performed.

5. The parameters of TRACK2 set SYSTRACE to trace from the first time through location 2070_8 until the fourth time through location 3005_8. Only those instructions which either are positioned within or access memory within locations 2075_8 and 3000_8 are edited for output. Control is then transferred to program TRACK at location 1005_8. TRACK then runs under its own control until location 2070_8 is encountered, initiating the trace routine where the tracing specified by the parameters of TRACK2 is performed.

6. After the final SYSTRACE control instruction is encountered, control is returned to SYS.

TACLTC SERVICE ROUTINE

FUNCTION

Compares two TAC Language format tapes card by card.

SERVICE ROUTINE INITIATION

The TACLTC service routine is called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,TACLTC,GO
		ABS } }	*,,GO (Binary deck of TACLTC)

INPUT-OUTPUT SYSTEM

The TACLTC routine uses PROC, a Philco input-output system, for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

TACLTC accepts two control instructions: COM and ENDALL. The instructions are written in standard SYS control line format. Illegal commands or parameters cause the routine to be terminated and control to be transferred to 1XCONER (page 4-18).

COM

Specifies the magnetic tapes to be compared, the number of blocks to be processed, and the mode of output, i.e., whether discrepancies of information are to be typed on the Console Typewriter or printed by the High-Speed Printer (via Tape 5). Any number of COM instructions may precede the ENDALL instruction.

Format

L	Location	Command	Address and Remarks
		COM	*t_1,t_2,nbp,mode*

Parameters

t_1
t_2 Both are decimal numbers, 0 through 15 (excluding 1 and 5), which indicate the magnetic tapes to be compared. If either is specified as 1, 5, or a number greater than 15, Error Type-Out 2 or 3 (page 5-8.4) will occur. If control instructions are entered via Tape 8 (the normal SYS input tape), this tape also will be excluded as a parameter. If used, Error Type-Out 3 (page 5-8.4) will occur. In addition, t_2 may not be the same tape specified by t_1, or Error Type-Out 7 (page 5-8.4) will occur.

nbp A decimal number, 1 through 19,000, which indicates the number of blocks to be processed (compared). If it is 0 or a number greater than 19,000, Error Type-Out 5 or 6 will occur (page 5-8.4).

mode An alphabetic character — T, F, or B — which indicates whether the output is to be printed on the Console Typewriter, in High-Speed Printer format via Tape 5, or both. If any character other than the above three is specified, or if the parameter is omitted, Error Type-Out 8 (page 5-8.4) will occur. One of the following must be selected:

T Indicates that the output, consisting of the contents of both cards in TAC Language format, is to be placed on Tape 5 for off-line printing on the High-Speed Printer.

F Indicates that the output, consisting of the Identity and Sequence field of the card from Tape t_1, is to be typed on the Console Typewriter. A plus sign (+) will appear following the Identity and Sequence field, if information in this particular field differs between tapes t_1 and t_2.

B Indicates that the output is to appear both on Tape 5 for off-line printing and on the Console Typewriter.

ACTION

When the TACLTC service routine is initiated, the following action takes place:

- TACLTC — TAC LANGUAGE TAPE COMPARISON is typed on the Console Typewriter.
- If any instruction other than COM or ENDALL is encountered in the first card following the TACLTC call, Error Type-Out 1 (page 5-8.3) will occur.
- If Tape 4 (user's program tape) is specified either as t_1 or t_2, UNIT 4 SELECTED FOR COMPARISON - DEPRESS ADVANCE IF OK will be typed on the Console Typewriter. If Tape 4 is to be compared, computer action will be continued by pressing the Advance Bar.
- If *mode* is T or B, the output heading (page 5-8.2) is produced and written on Tape 5. If *mode* is F, no heading is produced.
- The two tapes for comparison are assumed to be positioned such that the first block to be read from each tape is the first block to be compared.
- If none of the comparisons of the first block agree, ALL CARDS OF FIRST BLK DISAGREED - DEPRESS ADVANCE IF OK will be typed on the Console Typewriter. To continue, the Advance Bar may be pressed.
- As the tapes are compared, any discrepancies are immediately transferred to Tape 5 or the Console Typewriter as specified by the *mode* parameter.

- Following the comparison, COMPARISON COMPLETE is typed on the Console Typewriter, and both tapes are left positioned at the end of the designated number of blocks to be processed *nbp*.

- After ENDALL is encountered, END OF TAC LANGUAGE TAPE COMPARISON is typed on the Console Typewriter, and control is returned to SYS.

HIGH-SPEED PRINTER FORMAT

Output is written on Tape 5 only when *mode* is specified as either T or B. This output consists of a heading, a listing in TAC language format of any cards which differed, and an ending statement. If no discrepancies occurred, only the heading and ending will be present.

Heading

The heading contains three statements, written in the following format:

TACLTC – TAC LANGUAGE TAPE COMPARISON

LOGICAL TAPES t_1 AND t_2

NUMBER OF BLOCKS COMPARED *nbp*

Card Listing

If any cards differed, they will be listed in TAC language format beneath the body heading:

```
LOGICAL TAPE UNIT NUMBER          CARD CONTENTS
```

and each difference detected causes two lines of printing to be generated. A vertical space is inserted between each pair of lines.

The first line consists of the number of Tape t_1 and the contents of the card from that tape. The second line consists of corresponding information from Tape t_2.

Ending Statement

The magnetic tape output is concluded with the ending:

COMPARISON COMPLETE

indicating that the comparison process for a particular COM instruction has been terminated.

Remarks

A new heading is generated on Tape 5 for each COM control instruction with a *mode* parameter containing a T or B.

CONSOLE TYPEWRITER OUTPUT FORMAT

Output is typed on the Console Typewriter only when *mode* is specified as either F or B. This output consists of the heading CARD ID followed by a listing of the Identity and Sequence fields of cards from Tape t_1, if any differences in the cards between t_1 and t_2 are detected. A plus sign (+) will appear following the Identity and Sequence field if information differed between t_1 and t_2.

COMPARISON COMPLETE is typed after completion of each COM control instruction, and END OF TAC LANGUAGE TAPE COMPARISON is typed after an ENDALL control instruction is encountered.

CONSOLE TYPEWRITER NORMAL TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	TACLTC-TAC LANGUAGE TAPE COMPARISON	The TACLTC service routine has been initiated.
2	COMPARISON COMPLETE	The comparisons specified by the parameters of a COM control instruction have been completed.
3	END OF TAC LANGUAGE TAPE COMPARISON	An ENDALL control instruction was encountered by TACLTC; control is transferred to 1NXTCON (page 4-7).
4	DIFFERENCES	The *mode* parameter of a COM control instruction was a T (tape output only), and differences were detected during the card comparisons. The differences are written onto Tape 5.
5	NO DIFFERENCES	The *mode* parameter of a COM control instruction was a T and no differences were detected during the card comparisons.
6	CARD ID	The *mode* parameter of a COM control instruction was either an F (Console Typewriter) or a B (both Typewriter and Tape 5) and differences were detected during the card comparisons. The Identity and Sequence field of the cards from Tape t_1 that differed from Tape t_2 are listed beneath CARD ID.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	CONTROL LINE COMMAND NOT COM	A control instruction which was neither COM nor ENDALL was encountered by the TACLTC routine. Control is transferred to 1XCONER (page 4-18).
2	DESIGNATED TAPE UNIT IS GREATER THAN 15	The t_1 or t_2 parameter of a COM control instruction specified a number greater than 15. Control is transferred to 1XCONER.

CONSOLE TYPEWRITER ERROR TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
3	UNIT (1, 5, or 8) SELECTED FOR COMPARISON	The t_1 or t_2 parameter of a COM control instruction specified Tape 1, 5 or 8 (illegal tapes). (Tape 8 is illegal if control input is via Tape 8.) Control is transferred to 1XCONER.
4	UNIT 4 SELECTED FOR COMPARISON DEPRESS ADVANCE IF OK	The t_1 or t_2 parameter of a COM control instruction specified Tape 4. The program executed a halt. Because Tape 4 is the user's program tape, it is highly probable that its comparison may not be desired. If Tape 4 is to be compared, the Advance Bar should be pressed.
5	NUMBER OF BLOCKS TO BE COMPARED IS EQUAL TO 0	The *nbp* parameter of a COM control instruction specified 0. Control is transferred to 1XCONER.
6	NUMBER OF BLOCKS TO BE COMPARED IS GREATER THAN 19000	The *nbp* parameter of a COM control instruction specified a number greater than 19,000. Control is transferred to 1XCONER.
7	TAPES FOR COMPARISON HAVE SAME TAPE NO.	The t_1 and t_2 parameters of a COM control instruction have specified the same tape. Control is transferred to 1XCONER.
8	ILLEGAL OUTPUT PARAMETER	The *mode* parameter of a COM control instruction is not T, F, or B. Control is transferred to 1XCONER.
9	ALL CARDS OF FIRST BLK DISAGREED—DEPRESS ADVANCE IF OK	The program executed a halt because none of the twelve cards of the first block of Tape t_1 agreed with their counterparts on Tape t_2. The program will resume its comparison if the Advance Bar is pressed.
10	NO WRITE RING ON LOGICAL TAPE 5. CORRECT AND DEPRESS ADVANCE	Tape 5 does not have a Write Ring. The program will resume its comparison, if a Write Ring is placed on Tape 5 and the Advance Bar is pressed.

CONSOLE TYPEWRITER ERROR TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
11	LOGICAL TAPE UNIT *t* IN LOCAL. CORRECT AND DEPRESS ADVANCE.	Tape *t* is in local operation. The program will resume its comparison, if the tape is placed in automatic mode and the Advance Bar is pressed.
12	SPROCKET OR PARITY ERROR, DEPRESS ADVANCE TO ROCK TAPE 5 MORE TIMES	A parity or sprocket error was detected and five error correction attempts failed. Each time the Advance Bar is pressed, five additional correction attempts will be made. If the error remains uncorrected, control can be transferred manually to SYS.
13	REWIND NOT ACCEPTED AFTER TIO	A Proc-detected error has occurred. Control is transferred to 1ERRDMP (page 4-4).
14	NONRECOVERABLE TAPE ERROR	A PROC-detected non-recoverable tape error has occurred. Control is transferred to 1ERRDMP (page 4-4).

EXAMPLE

L	Location	Command	Address and Remarks	
		RPL	1,TACLTC,GO	$ (1)
		COM	3,6,20,B	$ (2)
		COM	9,10,15,F	$ (3)
		COM	11,14,23,T	$ (4)
		ENDALL		$ (5)

Explanation — Execution of these instructions causes the following action to take place:

1. The service routine TACLTC is called into operation.
2. Twenty blocks on Tapes 3 and 6 are compared. If any differences occur, output is produced both on Tape 5 for off-line printing and on the Console Typewriter.
3. Fifteen blocks on Tapes 9 and 10 are compared. If any differences occur, output is produced on the Console Typewriter.
4. Twenty-three blocks on Tapes 11 and 14 are compared. If any differences occur, output is produced on Tape 5 for off-line printing.
5. ENDALL causes END OF TAC LANGUAGE TAPE COMPARISON to be typed on the Console Typewriter and control to be returned to SYS.

TACSERV SERVICE ROUTINE

FUNCTION

Copies TAC language information from one magnetic tape to another, adding, deleting, or replacing cards or programs during the process.

SERVICE ROUTINE INITIATION

The TACSERV service routine may be called by any of the three following instructions:

L	Location	Command	Address and Remarks
		TACSERV	
		RPL	1,TACSERV2,GO
		ABS }	*,,GO (Binary deck of TACSERV)

INPUT REQUIREMENTS

Information on the input tape must be in TAC language format in Code Mode, 10 words per card, 12 cards per block. Each program on the tape must contain an I card as the first card of the first block, and an END card as the final card of the program. An I card contains an I in Column 9, blanks in Columns 10 through 16, and the program identity in Columns 17 through 32.

In this version of TACSERV, Tape Units 1, 2, 4, 5, and 8 are defined as illegal. All other tapes may be used without restriction.

INPUT-OUTPUT SYSTEM

TACSERV uses XORD, a Philco input-output subroutine, for all reads, writes, and tape movements.

CONTROL INSTRUCTIONS

TACSERV accepts 21 control instructions, written in standard SYS control line format. The instructions are grouped into two classes — Class I and Class II — preceded by a mandatory initilization instruction, NEWTAPE. Class I instructions provide program-by-program and tape movement operations, while Class II instructions provide card-by-card operations for revision of TAC language programs. Class II is initiated by the CORRECT instruction and is usually terminated by the ENDPROG instruction.

The 21 TACSERV control instructions are:

	Instruction	Function	Page
Initialization	NEWTAPE	Assigns the output tape.	5-9.3
Class I	ADDPROG	Transfers a specified program without change from the input to the output tape.	5-9.4
	COPY	Transfers information without change to the output tape until a specified sentinel block is encountered.	5-9.5
	COPYTIL	Transfers information without change to the output tape until a specified program is encountered.	5-9.5
	CORRECT	Transfers information without change to the output tape until a specified program is encountered, and sets TACSERV to the Class II mode (individual program corrections). Program correction instructions (Class II) must follow.	5-9.6
	DELPROG	Deletes a specified program from an input tape.	5-9.7
	ENDALL	Terminates the TACSERV routine and returns control to SYS.	5-9.8
	LOCATE	Searches for a specified program on an input tape.	5-9.9
	LOCSENT	Searches for a specified sentinel block on an input tape.	5-9.9
	REWIND	Rewinds one or more magnetic tapes.	5-9.10
	REWINDLO	Rewinds one or more magnetic tapes with lockout.	5-9.10
	SENTINEL	Designates an ending sentinel block.	5-9.11
	WRTSENT	Writes a sentinel block on a specified tape.	5-9.11
Class II	ADD	Adds one or more cards to a program.	5-9.12
	DELETE	Deletes one or more cards from program.	5-9.16
	ENDPROG	Terminates individual program correction, and resets TACSERV to Class I mode.	5-9.18
	LOCFLAD	Sets TACSERV to search for cards by symbolic address.	5-9.19
	LOCSEQ	Sets TACSERV to search for cards by sequence number.	5-9.19
	NOSEQ	Halts assignment of new sequence numbers within a program.	5-9.20
	REPLACE	Replaces one or more cards in a program.	5-9.20
	SEQ	Assigns new sequence numbers to cards within a program.	5-9.22

ERROR DETECTION

If any TACSERV errors are detected, the following action will be initiated:

- TACS E*n* will be typed on the Console Typewriter, where *n* denotes a particular error in the listing (page 5-9.26).
- The card in error and a description of the error will be written on the SYS output tape edited for printing on the High-Speed Printer.
- Control will be transferred to location 1ERRDMP (SYS error entry).

Tape errors will be processed by the XORD subroutine.

ONE-WORD ID SEARCH

A one-word identity search may be used to locate a program with an ID of more than eight characters. If an ID parameter of eight characters or less is terminated by a dollar sign ($), the second ID word of the program will be ignored; otherwise, a 16-character comparison will be made.

SENTINEL WORDS

All Philco 2000 characters except break characters (. , ; /) are acceptable within a sentinel word of a sentinel block. Spaces are significant. If a dollar sign appears prior to the eighth character of a sentinel (*sent*) parameter of a TACSERV control instruction, the sentinel word will be right-justified with leading zeroes. If more than eight characters are specified, Error E11 (page 5-9.26) will result.

SENTINEL BLOCKS

Whenever reference is made to "sentinel" or "sentinel block" within an input tape sentinel search description, each of the first 120 words of the sentinel block are assumed to contain:

- The *sent* configuration of a particular TACSERV instruction,
- The SENTINEL control instruction configuration if the above is not specified,
- A word of Z's if neither of the above is specified.

INITIALIZATION INSTRUCTION

NEWTAPE Assigns the output tape.

Format

L	Location	Command	Address and Remarks
		NEWTAPE	t$ subhead

Parameter

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the output tape.

subhead (OPTIONAL) a subtitle, composed of 40 or less characters beginning at col 41, to be printed immediately preceding the TACSERV listing as indicated by the first parameter of the ENDALL instruction (page 5-9.8). If a[illegible] issued, it must be preceded by $.

Action

No immediate action. The NEWTAPE instruction sets TACSERV to write all output from subsequent TACSERV control instructions on the designated output tape.

The next TACSERV control instruction is then requested.

Remarks

1. NEWTAPE must be specified as the first TACSERV instruction, and only one such instruction may be issued for each TACSERV run; otherwise, Error E4 (page 5-9.26) will result.

2. The tape designated by NEWTAPE may be rewound, but is illegal if specified within a tape unit parameter for any other TACSERV instruction.

Example

L	Location	Command	Address and Remarks
		NEWTAPE	6

Explanation: Execution of this instruction causes all output within the current TACSERV run to be written onto Tape 6.

CLASS I INSTRUCTIONS

Class I instructions perform all functions of the service routine except individual program corrections. TACSERV is preset to the Class I mode by the execution of the NEWTAPE control instruction (page 5-9.3). Only Class I instructions may be issued in this mode; otherwise, Error E4 (page 5-9.26) will result.

The CORRECT instruction (page 5-9.6) initiates the Class II mode (individual program corrections).

ADDPROG

Transfers a program from an input tape to the output tape.

Format

L	Location	Command	Address and Remarks
		ADDPROG	*t,id*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the input tape.

id 16 or fewer alphanumeric characters that provide the identity of the program to be added. Spaces are significant.

<u>NOTE</u>: An asterisk (*) specified in place of *t,id* indicates that a TACL card deck follows the ADDPROG control card and is to be written onto the output tape.

Action

All information on Tape *t*, beginning at the point at which the tape is currently positioned, is bypassed until the TACL program designated by *id* is encountered.

The program identified by *id* is copied onto the output tape and Tape *t* is positioned at the beginning of the next block following that program.

If ADDPROG * is specified, all the cards following that control card, up to and including the first END card encountered, are transferred to the output tape.

The next TACSERV control instruction is then requested.

If the sentinel block is encountered prior to the program *id*, Error E1 (page 5-9.26) will result.

Example

L	Location	Command	Address and Remarks
		ADDPROG	0,NAME1

Explanation: Execution of this instruction causes all information on Tape 0, beginning at the point at which the tape is currently positioned, to be bypassed until TACL program NAME1 is encountered. Program NAME1 is then copied onto the output tape and Tape 0 is positioned at the start of the next block following program NAME1.

COPY

Transfers without change all information from an input tape onto the output tape until a specified sentinel is located.

Format

L	Location	Command	Address and Remarks
		COPY	*t,sent*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape to be copied.

sent (OPTIONAL) Eight or fewer alphanumeric characters that comprise the first 120 words of a sentinel block. The sentinel specified by a COPY instruction overrides any previously-specified sentinel for the duration of that COPY instruction. If *sent* is omitted, the sentinel specified by the SENTINEL control instruction (page 5-9.11) or a sentinel block of Z's will terminate the copying action.

Action

All information on Tape *t*, beginning at the point at which *t* is currently positioned, is copied onto the output tape until the sentinel block is encountered.

The sentinel block is not copied onto the output tape, and Tape *t* is positioned at the start of the next block following the sentinel block.

The next TACSERV control instruction is then requested.

Example

```
L  Location  Command  Address and Remarks

             COPY     9,AAAAAAAA
```

Explanation: Execution of this instruction causes all information on Tape 9, beginning at the point at which the tape is currently positioned, to be copied onto the output tape until a sentinel block of A's is encountered.

COPYTIL

Transfers without change all information from a designated input tape to the output tape until a specified program is encountered.

Format

L	Location	Command	Address and Remarks
		COPYTIL	*t,id*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape to be copied.

id 16 or fewer alphanumeric characters that provide the identity of the program to which the tape is to be copied. Spaces are significant.

Action

All information on Tape *t*, beginning at the point at which *t* is currently positioned, is copied onto the output tape until the program designated by *id* is encountered.

Tape *t* is positioned at the beginning of the designated program.

The next TACSERV control instruction is then requested.

If the sentinel block is encountered prior to the program designated by *id*, Error E1 (page 5-9.26) will result.

Example

L	Location	Command	Address and Remarks
		COPYTIL	9,GAMMA

Explanation: Execution of this instruction causes Tape 9, beginning at the point at which the tape is currently positioned, to be copied onto the output tape until program GAMMA is encountered. The tape is positioned at the start of GAMMA.

CORRECT

Sets TACSERV to the Class II mode (individual program corrections).

Format

L	Location	Command	Address and Remarks
		CORRECT	*t,id*

Parameters

t A decimal number – 0, 3, 6, 7, or 9 through 15 – that indicates the tape to be copied.

id (OPTIONAL) 16 or fewer alphanumeric characters that provide the identity of the program to be corrected. Spaces are significant. The *id* parameter may be omitted if the tape is pre-positioned.

Action

All information on Tape *t* , beginning at the point at which *t* is currently positioned, is copied onto the output tape until the program designated by *id* is encountered.

Tape *t* is positioned at the beginning of the designated program.

If *id* is omitted, no search will be made for the program ID and no copying will be performed. Instead, corrections will be made from the point at which the input tape is currently positioned.

TACSERV is set to accept Class II instructions only (refer to page 5-9.12).

TACSERV is set to search for cards in the LOCFLAD mode (by symbolic location). (Refer to LOCFLAD, page 5-9.19.)

The next TACSERV control instruction is then requested.

If the specified sentinel block is encountered prior to the program designated by *id*, Error E1 (page 5-9.26) will result.

If a Class I instruction is encountered during the Class II mode, Error E4 (page 5-9.26) will result.

Remarks

The Class II mode is terminated by the execution of an ENDPROG control instruction (page 5-9.18).

Example

```
L  Location  Command   Address and Remarks

             CORRECT   10,ZEBRA
```

Explanation: Execution of this instruction causes all information on Tape 10, beginning at the point at which the tape is currently positioned, to be copied onto the output tape until program ZEBRA is encountered. The tape is positioned at the start of program ZEBRA. Class II instructions must follow.

DELPROG

Deletes a specified program from an input tape.

Format

L	Location	Command	Address and Remarks
		DELPROG	*t,id*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape from which the program is to be deleted.

id 16 or fewer alphanumeric characters that provide the identity of the program to be deleted. Spaces are significant.

Action

Tape *t*, beginning at the point at which *t* is currently positioned, is copied onto the output tape until the program designated by *id* is encountered.

The designated program is not copied onto the output tape, but is bypassed and the tape is positioned at the beginning of the block following the program.

The next TACSERV control instruction is then requested.

If the sentinel block is encountered prior to the program designated by *id*, Error E1 (page 5-9.26) will result.

Example

```
L  Location  Command   Address and Remarks

             DELPROG   11,YOKE
```

Explanation: Execution of this instruction causes all information on Tape 11, beginning at the point at which the tape is currently positioned, to be copied onto the output tape until program YOKE is encountered. Program YOKE is not copied onto the output tape, but is bypassed and the input tape is positioned at the start of the next block following the program.

ENDALL

Terminates the TACSERV service routine.

Format

L	Location	Command	Address and Remarks
		ENDALL	*list,sent*

Parameters

list (OPTIONAL) This parameter, written as LIST, indicates that an edited listing of the output tape is to be written on the system output tape for printing on the High-Speed Printer.

sent (OPTIONAL) Eight or fewer alphanumeric characters that are to comprise the 128 words of a sentinel block to be written on the output tape. If *sent* is omitted, a sentinel of Z's is assumed.

NOTE: Either or both parameters may be omitted. If only *sent* is used, it must be preceded by a break character; otherwise, Error E4 (page 5-9.26) will result.

Action

Two sentinel blocks of Z's or the configuration specified by *sent* are written onto the output tape designated by NEWTAPE (page 5-9.3).

If LIST is specified, a listing of the ID and block count of each program on the output tape, plus the total block count including sentinels, will be produced for printing on the High-Speed Printer. If LIST is omitted, no listing will be processed. If LIST is specified, the date and time are also produced for printing and will also appear two lines below the TACSERV 2 heading. Action of the TACSERV routine is terminated and control is returned to SYS via symbolic location 1NXTCON.

Remarks

Tapes are not rewound by TACSERV after execution of the ENDALL instruction.

Example

```
L  Location  Command  Address and Remarks

                      ENDALL   LIST,BBBBBBBB
```

Explanation: Execution of this instruction causes: (1) two sentinel blocks of B's to be written on the output tape, (2) a listing of the ID and block count of each program written on the output tape plus total block count to be written on the system output tape for printing on the High-Speed Printer, and (3) control to be returned to SYS.

LOCATE

Searches for a designated program on an input tape.

Format

L	Location	Command	Address and Remarks
		LOCATE	*t,id*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape on which the designated program is located.

id 16 or fewer alphanumeric characters that provide the identity of the program to be located. Spaces are significant.

Action

Tape *t* is searched for the program designated by *id.*

When the specified program is encountered, the tape is positioned at the start of the program.

The next TACSERV control instruction is then requested.

If the sentinel block is encountered prior to the program designated by *id,* Error E1 (page 5-9.26) will result.

Example

L	Location	Command	Address and Remarks
		LOCATE	9,XRAY

Explanation: Execution of this instruction causes Tape 9 to be searched for program XRAY and the tape to positioned at the start of that program.

LOCSENT

Searches for a designated sentinel block on the input tape.

Format

L	Location	Command	Address and Remarks
		LOCSENT	*t,sent*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape on which the designated sentinel block is located.

sent Eight or fewer alphanumeric characters that comprise the first 120 words of a sentinel block. If *sent* is omitted, Error E3 (page 5-9.26) will result.

Action

Tape *t* is searched forward until the sentinel specified by *sent* is encountered as the first 120 words of a block.

Tape *t* is positioned at the start of the next block following the sentinel block.

The next TACSERV control instruction is then requested.

Remarks

Any sentinel other than the one specified by LOCSENT will be disregarded.

Example

L	Location	Command	Address and Remarks
		LOCSENT	9,AAAAAAAA

Explanation: Execution of this instruction causes Tape 9 to be searched for a sentinel block of A's. When the sentinel is located, the tape will be positioned at the beginning of the block following the sentinel block.

REWIND

Rewinds one or more magnetic tapes.

L	Location	Command	Address and Remarks
		REWIND	*t*,...,*t*

Parameters

t One or more decimal numbers — 0, 3, 6, 7, or 9 through 15 — that indicate the tapes to be rewound.

Action

The tapes indicated by *t*,...,*t* are rewound.
The next TACSERV control instruction is then requested.

Example

L	Location	Command	Address and Remarks
		REWIND	3,6,9,10

Explanation: Execution of this instruction causes Tapes 3, 6, 9, and 10 to be rewound.

REWINDLO

Rewinds one or more magnetic tapes with lockout.

Format

L	Location	Command	Address and Remarks
		REWINDLO	*t*,...,*t*

Parameter

t One or more decimal numbers — 0, 3, 6, 7, or 9 through 15 — that indicate the tapes to be rewound.

Action

The tapes indicated by *t*,...,*t* are rewound with lockout.
The next TACSERV control instruction is then requested.

Remarks

The tape specified by NEWTAPE cannot be rewound with lockout. If such an attempt is made, Error E2 (page 5-9.26) will result.

Example

L	Location	Command	Address and Remarks
		REWINDLO	10,11

Explanation: Execution of this instruction causes Tapes 10 and 11 to be rewound with lockout.

SENTINEL

Designates an ending sentinel block.

Format

L	Location	Command	Address and Remarks
		SENTINEL	*sent*

Parameter

sent Eight alphanumeric characters (except break characters) that comprise the first 120 words of the sentinel block.

Action

There is no immediate action. TACSERV is set so that the operation of all subsequent instructions — prior to another SENTINEL instruction — will be terminated whenever the sentinel block designated by *sent* is encountered.

The next TACSERV control instruction is then requested.

Remarks

If the SENTINEL control instruction is omitted, a sentinel block of 120 words of Z's will be assumed by TACSERV for termination of subsequent instruction action.

Example

L	Location	Command	Address and Remarks
		SENTINEL	AAAAAAAA

Explanation: Execution of this instruction causes the action of all subsequent instructions to be terminated whenever a sentinel block of A's is encountered.

WRTSENT

Writes a specified sentinel block onto a designated tape.

Format

L	Location	Command	Address and Remarks
		WRTSENT	*t,sent*

Parameters

t A decimal number — 0, 3, 6, 7, or 9 through 15 — that indicates the tape on which the sentinel block is to be written.

sent Eight alphanumeric characters that comprise the 128 words of the sentinel block to be written.

Action

A sentinel block comprised of 128 words of the configuration specified by *sent* is written onto Tape *t*.

The next TACSERV control instruction is then requested.

Example

L	Location	Command	Address and Remarks
		WRTSENT	13,MMMMMMMM

Explanation: Execution of this instruction causes 128 words of M's to be written onto Tape 13.

CLASS II INSTRUCTIONS

Class II instructions perform individual program corrections such as the addition, deletion, or replacement of cards. TACSERV is set to accept Class II instructions by the execution of the CORRECT instruction (page 5-9.6). Only Class II instructions may be issued in this mode; otherwise, Error E4 (page 5-9.26) will result.

Only one program may be processed for each CORRECT instruction. Corrections to a program must be made in the sequential order of the program.

This mode is terminated by the processing of an END card within the program being corrected, either by the execution of an ENDPROG instruction (page 5-9.18) or by the replacement of the END card itself.

TRANSFER OF DATA

Whenever an ADD, REPLACE, or DELETE instruction is executed, information is transferred either from the beginning of the program, or from the point of last change (via a previous ADD, REPLACE, or DELETE instruction) to the point at which cards are to be currently added, replaced or deleted. Whenever the ENDPROG instruction is executed, the balance of the program is transferred from the input tape to the output tape, and the Class II phase is terminated.

ADD

Adds one or more cards to a program. ADD operates either in LOCFLAD or LOCSEQ mode (page 5-9.19).

Format and Parameters

LOCFLAD Mode

LOCFLAD Mode (specified number of cards to be added)

L	Location	Command	Address and Remarks
		ADD	*symb,n,total*

LOCFLAD Mode (unspecified number of cards)

L	Location	Command	Address and Remarks
		ADD	*symb,n*

symb The symbolic address used with *n* that designates the card in front of which the new card(s) are to be added.

n A decimal number that indicates the number of cards beyond the *symb* card, ahead of which the cards are to be added. If *n* is zero, the card identified by *symb* is the one ahead of which the new cards will be added.

total (OPTIONAL) A decimal number that indicates the number of consecutive cards to be added. When *total* is specified, any card but an END card may be added to the program being revised. If an END card should be detected in such a case, Error E10 (page 5-9.26) will result. If *total* is omitted, cards will be added until a TACSERV Class II instruction is encountered.

Format and Parameters

LOCSEQ Mode

LOCSEQ Mode (specified number of cards to be added)

L	Location	Command	Address and Remarks
		ADD	*xxxxxxxx, total*

LOCSEQ Mode (unspecified number of cards)

L	Location	Command	Address and Remarks
		ADD	*xxxxxxxx*

xxxxxxxx The eight-character Identity and Sequence field of the card in front of which new cards are to be added.

total (OPTIONAL) A decimal number that indicates the number of consecutive cards to be added. When *total* is specified, any card but an END card may be added to the program being revised. If an END card should be detected in such a case, Error E10 (page 5-9.26) will result. If *total* is omitted, cards will be added until a TACSERV Class II instruction is encountered.

Action

The TACL program specified by the most recent CORRECT control instruction (page 5-9.6) is copied card-by-card onto the output tape until the specified card is encountered. The specified card is not transferred to the output tape and remains available for further processing.

The cards to be added are written onto the output tape. The next TACSERV control instruction is then requested.

If the specified card is not encountered prior to the END card of the program being revised, Error E5 or E6 (page 5-9.26) will result, depending upon the mode of search (LOCFLAD or LOCSEQ, respectively).

Remarks

1. Cards are always added preceding the card specified by *symb* +*n* in LOCFLAD mode or *xxxxxxxx* in LOCSEQ mode.
2. If *symb* is blank or zero, cards will be added immediately preceding the *n*th card beyond the one last affected by an ADD, DELETE or REPLACE instruction for this program. If there were no such previous instructions, the cards will be added prior to the *n*th card from the beginning of the program.
3. If *symb* and *n* are both zero, the current location on the input tape is assumed.
4. The cards to be added must immediately follow the ADD control instruction.
5. An END card may not be added to a program being revised. If such an attempt is made, Error E10 (page 5-9.26) will result.

Example One

Assume that two instructions are to be added in LOCFLAD mode, between the third and fourth instructions of the following coding:

```
L  Location  Command  Address and Remarks

   TEST      CA
             SLAQ     6
             TMD      W/00000001
             JMP      START
```

The following TACSERV instructions are used to add the new cards:

```
L  Location  Command  Address and Remarks

             ADD      TEST,3,2
             JAED     (P)+2H
             AM       ZEBRA
```

Execution of the TACSERV instruction ADD causes the preceding portion of the program, from either its beginning or point of last change to the JMP START card (the card indicated by TEST+3) to be written onto the output tape. At this point, the two cards following ADD are written following the TMD W/00000001 card on the output tape. The input tape remains positioned at the JMP START card.

The corrected portion of the program on the output tape now appears as:

```
L  Location  Command  Address and Remarks

   TEST      CA
             SLAQ     6
             TMD      W/00000001
             JAED     (P)+2H
             AM       ZEBRA
```

Example Two

Assume that two instructions are to be added in LOCSEQ mode preceding the fourth instruction of the coding below.

Id & Seq	L	Location	Command	Address and Remarks
XRAY0012			CA	
XRAY0013			SLAQ	6
XRAY0014			TMD	W/00000001
XRAY0015			JMP	START

The following TACSERV instructions are used to enter the new cards:

Id & Seq	L	Location	Command	Address and Remarks
			ADD	XRAY0015
			JAED	(P)+2H
			AM	ZEBRA
			ENDPROG	

Execution of the TACSERV instruction ADD causes the preceding portion of the program, from either its beginning or point of last change up to but not including card XRAY0015 (JMP START), to be written onto the output tape. At this point, the two cards following ADD are written following the TMD W/00000001 card on the output tape. ENDPROG indicates to TACSERV that there are no further corrections to be made to this program, and the remainder of the program is transferred to the output tape.

The corrected portion of the program on the output tape now appears as:

Id & Seq	L	Location	Command	Address and Remarks
XRAY0012			CA	
XRAY0013			SLAQ	6
XRAY0014			TMD	W/00000001
			JAED	(P)+2H
			AM	ZEBRA
XRAY0015			JMP	START

DELETE

Deletes one or more cards from a TACL program. DELETE operates either in LOCFLAD or LOCSEQ mode (page 5-9.19).

Format and Parameters
LOCFLAD Mode

LOCFLAD Mode

L	Location	Command	Address and Remarks
		DELETE	*symb,n,total*

symb The symbolic address used in conjunction with *n* that designates the first card to be deleted.

n A decimal number that indicates the position (number of cards past the card designated by *symb*) of the first card to be deleted. If *n* is zero, the card identified by *symb* will be the first to be deleted.

total (OPTIONAL) This parameter, which indicates the number of cards to be deleted, may be written as one of the following:

- A decimal number that specifies the exact number of cards to be deleted.
- TO END indicates that all cards from the point at which the program is currently positioned, up to but not including the END card of the program being revised, are to be deleted.

If *total* is omitted, one card will be deleted.

Format and Parameters
LOCSEQ Mode

LOCSEQ Mode

L	Location	Command	Address and Remarks
		DELETE	*xxxxxxxx,total*

xxxxxxxx The eight-character Identity and Sequence field of the first card to be deleted.

total (OPTIONAL) This parameter, which indicates the number of cards to be deleted, may be written as one of the following:

- A decimal number that specifies the exact number of cards to be deleted.
- TO END indicates that all cards from the point at which the program is currently positioned, up to but not including the END card of the program being revised, are to be deleted.

If *total* is omitted, one card will be deleted.

Action

The TACL program designated by the most recent CORRECT instruction (page 5-9.6) is copied card-by-card onto the output tape until the specified c ırd is encountered.

The specified card (first card to be deleted) plus all subsequent consecutive cards as determined by *total* are bypassed on the input tape. The input tape is positioned at the card following the last one bypassed.

If TO END is specified as *total*, all cards of the program being revised, from the specified card up to but not including the END card, will be bypassed (deleted). The input tape will then be positioned at the END card.

The next TACSERV control instruction is then requested.

If the specified card is not encountered prior to processing the END card of the program being revised, Error E5 or E6 (page 5-9.26) will result, depending upon the mode of search.

Remarks

1. If *symb* is blank or zero, cards will be deleted starting with the *n*th card beyond the last one referenced in a previous ADD, DELETE or REPLACE instruction for this program. If there were no such previous instructions, cards will be deleted beginning with the *n*th card of the program.
2. If *symb* and *n* are both zero, the current location on the input tape is assumed.
3. An END card of a program being revised cannot be deleted. If such an attempt is made, Error E9 (page 5-9.26) will result.

Example

Assume that the third and fourth instructions are to be deleted in LOCFLAD mode from the collowing coding.

```
L  Location  Command  Address and Remarks

   TEST      CA
             SLAQ     6
             TQD
             JMP      OUT
             TMD      W/00000001
             JAED     (P)+2H
             AM       ZEBRA
             JMP      START
```

The following TACSERV instruction is used to delete the two instructions.

```
L  Location  Command  Address and Remarks

             DELETE   TEST,2,2
```

Execution of the TACSERV instruction DELETE causes the preceding portion of the program, from either its beginning or point of last change, up to but not including the TQD instruction, to be written on the output tape. At this point, the TQD instruction and the one immediately following it are bypassed on the input tape and the tape is positioned at the TMD W/00000001 card.

The corrected portion of the program on the output tape now appears as:

L	Location	Command	Address and Remarks
	TEST	CA	
		SLAQ	6

The card TMD W/00000001 and those following remain on the input tape and are available for further processing.

ENDPROG

Terminates individual program correction.

Format

L	Location	Command	Address and Remarks
		ENDPROG	

Parameters

None

Action

The balance of the program being corrected, from the point at which the tape is currently positioned, is copied onto the output tape.

The input tape is positioned at the beginning of the block following the last block of the program just processed.

TACSERV is set to NOSEQ mode, and reset to accept Class I instructions.

The next TACSERV instruction is then requested. If any instruction other than Class I is issued following ENDPROG, Error E4 (page 5-9.26) will result.

Remarks

If an END card was replaced during correction of a program, an ENDPROG control instruction must not appear for that program. All action described above for ENDPROG will automatically be performed following replacement of an END card. If an attempt is made to execute an ENDPROG instruction, Error E4 (page 5-9.26) will result.

LOCFLAD

Sets the TACSERV routine to find cards by symbolic address for subsequent ADD, DELETE, and REPLACE control instructions (pages 5-9.12, 5-9.16, and 5-9.20).

Format

L	Location	Command	Address and Remarks
		LOCFLAD	

Parameters

None

Action

There is no immediate action. All subsequent card searching within the current CORRECT instruction phase is set to use symbolic addresses in the Location field.

The next TACSERV control instruction is then requested.

Remarks

1. TACSERV is automatically set to the LOCFLAD mode of operation whenever a CORRECT instruction (page 5-9.6) is executed.
2. The LOCFLAD mode is terminated by the execution of a LOCSEQ instruction.

LOCSEQ

Sets the TACSERV routine to find cards by Identity and Sequence number for subsequent ADD, DELETE and REPLACE control instructions (pages 5-9.12, 5-9.16, and 5-9.20).

Format

L	Location	Command	Address and Remarks
		LOCSEQ	

Parameters

None

Action

There is no immediate action. All subsequent card searching within the current CORRECT instruction phase is set to use the sequence number in the Identity and Sequence field.

The next TACSERV control instruction is then requested.

Remarks

The LOCSEQ mode of operation is terminated by the execution of a LOCFLAD instruction.

NOSEQ

Terminates assignment of new sequence numbers to cards of a TACL program being corrected. (Assignment of sequence numbers is initiated by the SEQ control instruction, page 5-9.22.)

Format

L	Location	Command	Address and Remarks
		NOSEQ	

Parameters

None

Action

There is no immediate action. When NOSEQ is executed, all cards written on the output tape from this point forward will be transferred with their Identity and Sequence field unaltered.

The next TACSERV control instruction is then requested.

REPLACE

Replaces one or more cards in a TACL program. REPLACE operates either in LOCFLAD or LOCSEQ mode (page 5-9.19).

Format and Parameters
LOCFLAD Mode

LOCFLAD Mode

L	Location	Command	Address and Remarks
		REPLACE	*symb,n,total*

symb The symbolic address used in conjunction with *n* that designates the first card to be replaced.

n A decimal number that indicates the position (number of cards past the card designated by *symb*) of the first card to be replaced. If *n* is zero, the card identified by *symb* will be the first to be replaced.

total (OPTIONAL) A decimal number that indicates the number of consecutive cards to be replaced. If *total* is omitted, one card will be replaced.

Format and Parameters
LOCSEQ Mode

LOCSEQ Mode

L	Location	Command	Address and Remarks
		REPLACE	*xxxxxxxx,total*

xxxxxxxx The eight-character Identity and Sequence field of the first card to be replaced.

total (OPTIONAL) A decimal number that indicates the number of consecutive cards to be replaced. If *total* is omitted, one card will be replaced.

Action

The TACL program specified by the most recent CORRECT instruction (page 5-9.6) is copied card-by-card onto the output tape until the specified card is encountered.

The specified card plus subsequent cards as determined by *total* are replaced. Their corresponding replacement cards (following the REPLACE instruction) are written onto the output tape. The input tape is positioned at the card following the last one replaced.

The next TACSERV control instruction is then requested.

If the specified card is not encountered prior to processing the END card of the program being revised, Error E5 or E6 (page 5-9.26) will result, depending upon the mode of search.

Remarks

1. If *symb* is blank or zero, cards will be replaced starting with the *n*th card beyond the last one referenced in a previous ADD, DELETE or REPLACE instruction for this program. If there were no previous such instructions, cards will be replaced beginning with the *n*th card of the program.

2. If *symb* and *n* are both zero, the current location on the input tape is assumed.

3. An END card of a program being revised may be replaced only by another END card. If an END card is replaced by any card other than an END card, or if an END card is used to replace any card other than an END card, Error E8 (page 5-9.26) will result.

Example

Assume that the third and fourth instructions are to be replaced in LOCFLAD mode in the following coding.

```
L  Location  Command  Address and Remarks

   TEST      CA
             SLAQ     6
             TMQ      W/77777777
             JAEQ     (P)+3H
             AM       ZEBRA
             JMP      START
```

The following TACSERV instructions are used to replace the two cards.

```
L  Location  Command   Address and Remarks

             REPLACE   TEST,2,2
             TMD       W/00000001
             JAED      (P)+2H
```

Execution of the TACSERV instruction REPLACE causes the preceding portion of the program, from either its beginning or point of last change up to but not including the TMQ W/77777777 card (indicated by TEST+2), to be written on the output tape. At this point, the TMQ W/77777777 card and the one following on the input tape are bypassed. The two cards following REPLACE are written on the output tape following SLAQ 6.

The input tape remains positioned at the AM ZEBRA card, which is available for further processing.

The revised portion of the program on the output tape now appears as:

```
L  Location  Command  Address and Remarks

   TEST      CA
             SLAQ     6
             TMD      W/00000001
             JAED     (P)+2H
```

SEQ

Sets TACSERV to assign new sequence numbers, beginning with a designated configuration, to each subsequent card of a TAC language program being transferred to the output tape.

Format

	Location	Command	Address and Remarks
		SEQ	*seqno,inc,cols*

Parameters

seqno (OPTIONAL) Eight alphanumeric characters that indicate the configuration to be placed in the Identity and Sequence field of the next output card of the program being revised. The *seqno* parameter may consist of one to seven alphabetic characters or spaces followed by one to seven decimal characters, or may consist of eight decimal characters. The total must be eight, representing the eight columns of the Identity and Sequence field. If *seqno* is omitted, the Identity and Sequence field of the first card transferred to the output tape will be transferred as the first sequence number.

inc (OPTIONAL) A decimal number that indicates the amount by which *seqno* is to be incremented from card to card. If *inc* is omitted, an increment of one will be assumed.

cols (OPTIONAL) A decimal number, 1 through 8, that indicates the maximum number of columns of *seqno* that may be affected by the incrementing process. If *cols* is omitted, four columns will be assumed. The number specified as *cols* must be equal to or less than the number of numeric digits within the *seqno* configuration; otherwise, Error E7 (page 5-9.26) will result.

Action

There is no immediate action. All cards transferred to the output tape following execution of the SEQ instruction will be sequenced according to the above parameter specifications.

The next TACSERV control instruction is then requested.

Remarks

1. Incrementation numbering is modulo 10^c, where c is the number of columns to be affected.

2. Sequencing is terminated by the NOSEQ control instruction (page 5-9.20) or the ENDPROG control instruction (page 5-9.18).

3. Each of the SEQ parameters may be used independently of the others, or may be omitted.

4. If all parameters are omitted, the sequence number of the first card transferred to the output tape following execution of the SEQ instruction will be assumed as the initial sequence number. Subsequent cards will be sequenced in Columns 5, 6, 7, and 8 and incremented by one.

5. If only the first parameter (*seqno*) is specified, *seqno* will be the sequence number of the first card. Subsequent cards will be sequenced on the last four columns and incremented by one.

6. If only the first two parameters (*seqno* and *inc*) are specified, *cols* will be four, with an increment designated by *inc*. No break character is required for omission of the final parameter.

7. If the first parameter is omitted, but the remaining two are specified, the Identity and Sequence field of the first card will be assumed. In this case, a break character must be written preceding the *inc* parameter.

8. If the first two parameters are omitted and *cols* is specified, the Identity and Sequence field of the first card will be assumed, and subsequent card sequence numbers will be incremented by one. Columns will be affected as indicated by *cols*. Two break characters must preceed the *cols* parameter.

9. If the first and third parameters are omitted and only *inc* is specified, the Identify and Sequence field of the first card will be assumed. Incrementation will be indicated by *inc*, and four columns are to be affected by the sequencing process.

10. If *inc* is omitted, an increment of one will be assumed. Two break characters must be present between the first and third parameters to indicate the missing parameter.

Example

L	Location	Command	Address and Remarks
		SEQ	XRAY0001,2,3

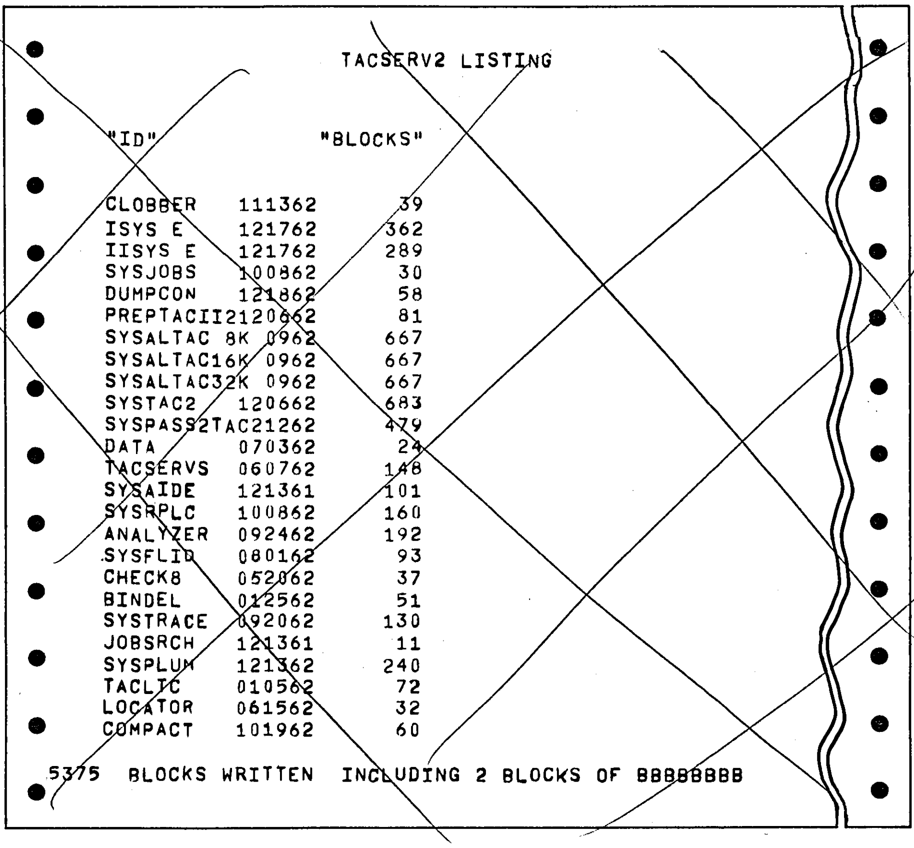
Explanation: Execution of this instruction presets TACSERV to begin sequencing with the configuration XRAY0001 with an increment of two, affecting only the last three columns. The first card is to be given the sequence number XRAY0001, the second card to be given the number XRAY0003, and so on. If the number XRAY0999 is reached, the following number will be XRAY0001.

TACSERV OUTPUT

Output from TACSERV may consist of an updated TAC language tape and a printed listing of programs written onto the output tape. If a TACSERV control instruction error is detected, the error will be indicated on the Console Typewriter and High-Speed Printer.

Program Listing

If LIST is written as a parameter of the ENDALL control instruction (page 5-9.8), a listing of the ID and block count of each program written on the output tape, ~~plus~~ the date and time the total block count, including sentinels, will be edited and placed on the system output tape for printing on the High-Speed Printer. ~~A typical listing is illustrated below:~~



```
                    TACSERV2 LISTING

   "ID"                 "BLOCKS"

   CLOBBER     111362        39
   ISYS E      121762       362
   IISYS E     121762       289
   SYSJOBS     100862        30
   DUMPCON     121862        58
   PREPTACII2120662          81
   SYSALTAC 8K 0962         667
   SYSALTAC16K 0962         667
   SYSALTAC32K 0962         667
   SYSTAC2     120662       683
   SYSPASS2TAC21262         479
   DATA        070362        24
   TACSERVS    060762       148
   SYSAIDE     121361       101
   SYSRPLC     100862       160
   ANALYZER    092462       192
   SYSFLID     080162        93
   CHECK8      052062        37
   BINDEL      012562        51
   SYSTRACE    092062       130
   JOBSRCH     121361        11
   SYSPLUM     121362       240
   TACLTC      010562        72
   LOCATOR     061562        32
   COMPACT     101962        60

 5375  BLOCKS WRITTEN  INCLUDING 2 BLOCKS OF BBBBBBBB
```

Console Typewriter Error Indications

Whenever an error is detected during operation of the TACSERV routine, the following indication is typed on the Console Typewriter:

TACS E*n*

where *n* indicates the type of error as described on page 5-9.26.

High-Speed Printer Error Indications

Detection of an error also causes the following indication to be written on the system output tape, edited for printing on the High-Speed Printer:

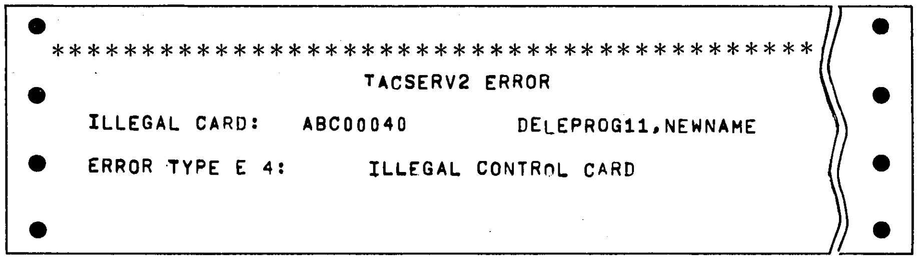
TACSERV2 ERROR

ILLEGAL CARD *card contents*

ERROR TYPE E*n* *error statement*

where *card contents* indicates the contents of the entire illegal card, and *n* and *error statement* indicate the appropriate error indication.

A typical print-out of a TACSERV error is illustrated below:



```
****************************************************
                         TACSERV2 ERROR
   ILLEGAL CARD:    ABC00040        DELEPROG11,NEWNAME
   ERROR TYPE E 4:      ILLEGAL CONTROL CARD
```

Error Exit

Following detection and indication of a TACSERV error, control is transferred to 1ERRDMP. A listing of all TACSERV errors follows below:

TACSERV ERRORS

Error Type	Error Statement on Printer Listing	Explanation
E1	ID NOT FOUND	The program ID requested was not found prior to the sentinel block.
E2	ILLEGAL TAPE NUMBER *n*	The tape number is: • Reserved for the operating system • Greater than 15 • The output tape

Continued

Error Type	Error Statement on Printer Listing	Explanation
E3	MISSING PARAMETER	The card being processed does not have a required parameter.
E4	ILLEGAL CONTROL CARD	A control card: • Is in improper format • Has been issued at the wrong time.
E5	SYMBOL *symb* MISSING	The requested symbol was not found prior to an END card.
E6	SEQ NUMBER *xxxxxxxx* MISSING	The requested sequence number was not found prior to an END card.
E7	ILLEGAL PARAMETER	Too many parameters, or an illegal character.
E8	ILLEGAL END CARD REPLACEMENT	An attempt was made to replace an END card with a card other than an END card.
E9	ATTEMPT TO DELETE END CARD	An attempt was made to delete an END card.
E10	ATTEMPT TO ADD END CARD	An attempt was made to add an END card.
E11	IMPROPER SENTINEL PARAMETER	More than eight characters are in a sentinel parameter.

XORD Errors

Errors detected by the XORD subroutine during reading, writing and tape movement cause the computer to halt with M/11111 displayed in the Program Register. Refer to the XORD Program Description for a listing of the errors typed on the Console Typewriter at the time of the XORD error halt.

TACSERV EXAMPLE

Assume that all programs on Tape 9, up to but not including program ZEBRA, and all programs on Tape 11, up to <u>and</u> including program ABLE, are to be transferred to Tape 10. Program ABLE is to be revised as it is transferred.

```
L  Location  Command   Address and Remarks
             NEWTAPE   10                $(1)
             REWIND    9,10,11           $(2)
             COPYTIL   9,ZEBRA           $(3)
             CORRECT   11,ABLE           $(4)
             LOCSEQ                      $(5)
             SEQ       ABLE0001,,3       $(6)
             ADD       ABLE0032,1        $(7)
             TMA       CARD              $(8)
             ENDPROG                     $(9)
             ENDALL    LIST,BBBBBBBB     $(10)
```

Explanation: Execution of these instructions causes the following action to take place:

1. Tape 10 is assigned as the output tape.
2. Tapes 9, 10, and 11 are rewound.
3. All information on Tape 9 is copied onto the output tape from the beginning of the tape until program ZEBRA is encountered. Tape 9 remains positioned at the beginning of that program.
4. All information on Tape 11 is copied onto the output tape until program ABLE is encountered. The tape is positioned at the start of program ABLE, and TACSERV is set to the Class II phase for card-by-card revision of the program.
5. TACSERV is set to search ABLE by card sequence number.
6. Cards transferred to the output tape are to be given new sequence numbers, beginning with ABLE0001, starting with the program I card. Only the last three columns of the Identity and Sequence number are to be affected by sequencing. Omission of the second parameter indicates that card sequence numbers will be incremented by one.
7. All cards from the beginning of the program, up to but not including card ABLE0032, are sequenced and transferred to the output tape. The input tape is positioned at card ABLE0032.
8. The card containing the instruction TMA CARD is also sequenced and written onto the output tape.
9. ENDPROG causes TACSERV to transfer the remainder of the program, beginning with card ABLE0032, to the output tape, reassigning sequence numbers in consecutive order. TACSERV is then reset to the Class I phase of instructions.
10. The ENDALL instruction causes TACSERV to produce a listing of the ID and block count of each program on the output tape, plus the total block count, and date and ti for printing on the High-Speed Printer. Two sentinel blocks of 128 words of B's are written onto Tape 10. Action of the TACSERV routine is terminated, and control is transferred to 1NXTCON.

SYSFLID SERVICE ROUTINE

FUNCTION

Finds or Lists the IDentities of RPL, ABS, REL, and TAC language programs on magnetic tape. FLID may also be used to position any tape following a specified sentinel block.

SERVICE ROUTINE INITIATION

The SYSFLID service routine may be called into operation by either of the following instructions:

L	Location	Command	Address and Remarks
		RPL	1,SYSFLID,GO
		ABS }	*,,GO (Binary deck of SYSFLID)

INPUT REQUIREMENTS

In order for SYSFLID to recognize a program, the program must start at the beginning of a block. If a program starts elsewhere in the block, it will not be detected by SYSFLID, and will be identified on the listing as OTHER. OTHER indicates that a block contains information not identifiable as RPL, ABS, REL, or TACL.

ABS programs must be preceded by a dummy PMAX card to be recognized by SYSFLID.

INPUT-OUTPUT SYSTEM

The SYSFLID service routine uses XORD for all reading, writing, and tape movements.

CONTROL INSTRUCTIONS

SYSFLID accepts six control instructions, written in standard control line format:

INSTRUCTION	FUNCTION
REWIND	Rewinds one or more tapes.
LOCSENT	Locates a specified sentinel block.
SENTINEL	Designates an ending sentinel block for subsequent FIND or LIST control instructions.
FIND	Searches for a specified program.
LIST	Provides a listing of the type, ID, and block count of programs on a tape.
ENDALL	Terminates action of the SYSFLID service routine and returns control to SYS.

Illegal control instructions and/or parameters cause CONTROL ERROR to be typed on the Console Typewriter and control to be transferred to 1ERRDMP (page 4-4), which provides a post-mortem dump.

REWIND

Rewinds one or more magnetic tapes.

Format

L	Location	Command	Address and Remarks
		REWIND	*t,...,t*

Parameters

t One or more decimal numbers, 0 through 15, which indicate the tapes to be rewound.

Action

When the REWIND control instruction is executed, all tapes indicated by *t,...,t* are rewound. SYSFLID then requests its next control instruction.

Example

L	Location	Command	Address and Remarks
		REWIND	3,9,11,14

Explanation: Execution of this instruction causes tapes 3, 9, 11, and 14 to be rewound.

LOCSENT

Searches a designated tape forward until a specified sentinel block is located.

Format

L	Location	Command	Address and Remarks
		LOCSENT	*t,sent*

Parameters

t A decimal number, 0 through 15, which indicates the tape to be searched.

sent Eight alphanumeric characters which comprise the first 20 words of the sentinel block. Spaces are significant. If more than eight characters are specified, only the first eight will be recognized.

Action

When the LOCSENT control instruction is initiated, the following action takes place:

- Tape *t* is searched in a forward direction until SYSFLID locates a block whose first 120 words contain the configuration designated by *sent*.
- If the specified sentinel block is not found, the sentinel search will continue until a non-recoverable tape error is encountered, or the program is halted by an operator.
- When the specified sentinel block is located, Tape *t* is positioned immediately following the sentinel block.
- The next SYSFLID control instruction is then requested.

Example

```
L   Location   Command    Address and Remarks

               LOCSENT    9,AAAAAAAA
```

Explanation: Execution of this instruction causes Tape 9 to be searched for sentinel block of A's. After the specified sentinel block is located, Tape 9 is positioned immediately following the sentinel block.

SENTINEL

Presets SYSFLID so that subsequent FIND or LIST control instructions will be terminated whenever the sentinel block indicated by this instruction is encountered. The SENTINEL instruction, if used, must precede the FIND or LIST instructions to which it relates.

Format

L	Location	Command	Address and Remarks
		SENTINEL	*sent*

Parameter

sent Eight alphanumeric characters which comprise the first 120 words of the sentinel block. Spaces are significant. If more than eight characters are specified, only the first eight will be recognized.

Action

When the SENTINEL control instruction is initiated, the following action takes place:

- SYSFLID is preset so that the operation of all subsequent FIND or LIST instructions – prior to another SENTINEL instruction – will be terminated whenever the sentinel block designated by *sent* is encountered.
- The next SYSFLID control instruction is then requested.

Remarks

If the SENTINEL control instruction is omitted, a sentinel block of Z's is assumed by SYSFLID for termination of FIND and LIST operations.

Example

L	Location	Command	Address and Remarks
		SENTINEL	LLLLLLLL

Explanation: Execution of this instruction causes the action of all subsequent FIND or LIST instructions to be terminated whenever a sentinel block of L's is encountered.

FIND

Searches a designated tape for a specific program.

Format

L	Location	Command	Address and Remarks
		FIND	*t,format,id*

Parameters

t A decimal number, 0 through 15, which indicates the tape to be searched.

format The type of program to be located. One of the following must be selected:

RPL	Indicates that an RPL program is to be located
ABS	Indicates that an ABS program is to be located
REL	Indicates that an REL program is to be located
TACL	Indicates that a TAC language program in Code or Image Mode is to be located

id Sixteen or fewer alphanumeric characters which indicate the ID of the program to be located. Spaces are significant. If eight or fewer characters are specified as *id,* only the first eight characters of each program on tape will be compared with the *id* (refer to ONE-WORD SEARCH, page 3-4).

Action

When the FIND control instruction is initiated, the following action takes place:

- Tape *t* is searched forward for the designated program, specified by the *id* and *format* parameters.
- When the designated program is encountered, Tape *t* is positioned at the beginning of the program. The next SYSFLID control instruction is then requested.
- If a sentinel block designated by the SENTINEL control instruction (page 5-10.2), or a sentinel block of Z's (whichever is in effect), is encountered prior to the designated program, *id* MISSING will be typed on the Console Typewriter. Control will then be transferred to 1ERRDMP (page 4-4).

Example

L	Location	Command	Address and Remarks
		FIND	4,RPL,PROGRAMA

Explanation: Execution of this instruction causes Tape 4 to be searched for the RPL program PROGRAMA. After the program is located, Tape 4 is positioned at the beginning of PROGRAMA. If a designated sentinel block is encountered prior to PROGRAMA, the Console Typewriter will type PROGRAMA MISSING. Control will then be transferred to 1ERRDMP.

LIST

Searches a designated tape and provides a listing of the type, ID, and block count of programs on the tape. The listing may be typed on the Console Typewriter and/or printed off-line on the High-Speed Printer.

Format

L	Location	Command	Address and Remarks
		LIST	*t,format,mode* $ subhead

Parameters

t A decimal number, 0 through 15, which indicates the tape to be searched.

format The type of program to be listed. One of the following must be selected:

RPL	Indicates that RPL programs are to be listed
ABS	Indicates that ABS programs are to be listed
REL	Indicates that REL programs are to be listed
TACL	Indicates that TAC language programs (in Code or Image Mode) are to be listed
ALL	Indicates that all of the above types of programs are to be listed

mode The type of output for the listing. (Refer to Output Format, page 5-10.6.) Three options are available:

T	Indicates that the listing is to be edited and written on Tape 5 for off-line printing on the High-Speed Printer.
B	Indicates that the listing is to be placed on Tape 5 for printing on the High-Speed Printer and typed on the Console Typewriter.
(Blank) or Δ	The omission of the *mode* parameter indicates that the listing is to be typed on the Console Typewriter.

subhead (optional) A subtitle composed of 40 or less characters beginning in col. 41, to be printed or typed immediately preceding the listing. If subhead is specified, it must be preceded by a $

FEB 1964
~~November 1962~~

Action

When the LIST control instruction is initiated, the following action takes place:

- Tape *t* is searched for the type of program(s) specified by the *format* parameter.
- As each of the programs designated by *format* is encountered, its program type, ID and block count are typed on the Console Typewriter and/or are written on Tape 5 for off-line printing, depending upon the *mode* parameter.
- The listing action is terminated whenever the designated sentinel is encountered.
- The next SYSFLID control instruction is then requested.

Remarks

1. In addition to the type, ID and block count of each of the programs designated by *format* is encountered, SYSFLID lists all sentinels and a count of all blocks from the initial position on the tape, up to and including the final sentinel block.

2. If ALL is specified as the *format* parameter, blocks of information not recognized as RPL, ABS, REL, or TACL programs will be listed as ++++OTHER.

3. Data which appears in the same format as a sentinel block, i.e., contains 120 similar words, is listed as a sentinel but will not stop the listing function.

4. An RPL program which contains any unrecognizable RPL control word will be listed by SYSFLID, but will be identified as a BAD RPL. The block containing the unrecognizable control word and any blocks remaining in the program will be identified as ++++OTHER.

5. If any sentinel block is encountered prior to an END card in a TACL program, or prior to a binary END or JMP card in an ABS or REL program, the listing will include:

The date and time (described with CLOCK instruction pg 3-8.4) will be printed on each listing immediately below its title, FLID STING.

   - The program type as TACL or BINARY (the latter denoting an ABS or REL program),
   - The first eight characters of the program ID,
   - The words NO END (to denote omission of the TAC END card or the binary jump card), and
   - The block count of the program.

Example

L	Location	Command	Address and Remarks
		LIST	3,TACL,B

Explanation: Execution of this instruction causes TACL programs and sentinel blocks on Tape 3 to be listed until the designated sentinel is encountered. The listing plus a total block count of the tape is typed on the Console Typewriter and written on Tape 5 for off-line printing on the High-Speed Printer.

ENDALL

Terminates action of the SYSFLID service routine and returns control to SYS.

Format

L	Location	Command	Address and Remarks
		ENDALL	

Parameters

None

Action

When the ENDALL control instruction is initiated, action of the SYSFLID routine is terminated and control is returned to SYS.

OUTPUT FORMAT

A printed listing of programs will be produced via the Console Typewriter and/or Tape 5 edited for off-line printing on the High-Speed Printer whenever the LIST control instruction is specified. ~~The figure on page 5-10.7 indicates the format of such a listing as it would appear on the High-Speed Printer~~. The format of both the Console Typewriter and High-Speed Printer listings is similar.

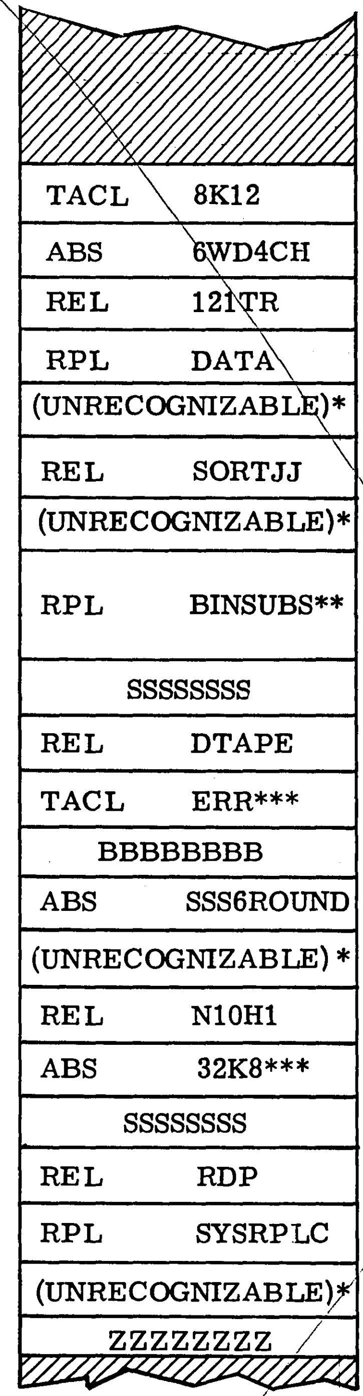
The printed listing is initiated by date and time (as specified by the CLOCK uction, pg 3-8.4). If the subhead parameter e LIST instruction is specified, the subtitle will be printed between the date and time and the body of the ing. The body of the listing is ~~[illegible]~~ single spaced, and contains the type of program, the program's ID & block count, in left-to-right order. At the end of the listing, the total block count is given.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	*id* MISSING	A specified sentinel block was encountered prior to program *id* during execution of a FIND control instruction. Control is transferred to 1ERRDMP (page 4-4) which provides a post-mortem dump.
2	CONTROL ERROR	An illegal control instruction has been submitted to SYSFLID. Control is transferred to 1ERRDMP as above.
3	TAPE *t* WR RING	There is no Write Ring on Tape *t*. The operator may insert a Write Ring and press the Advance Bar to continue.

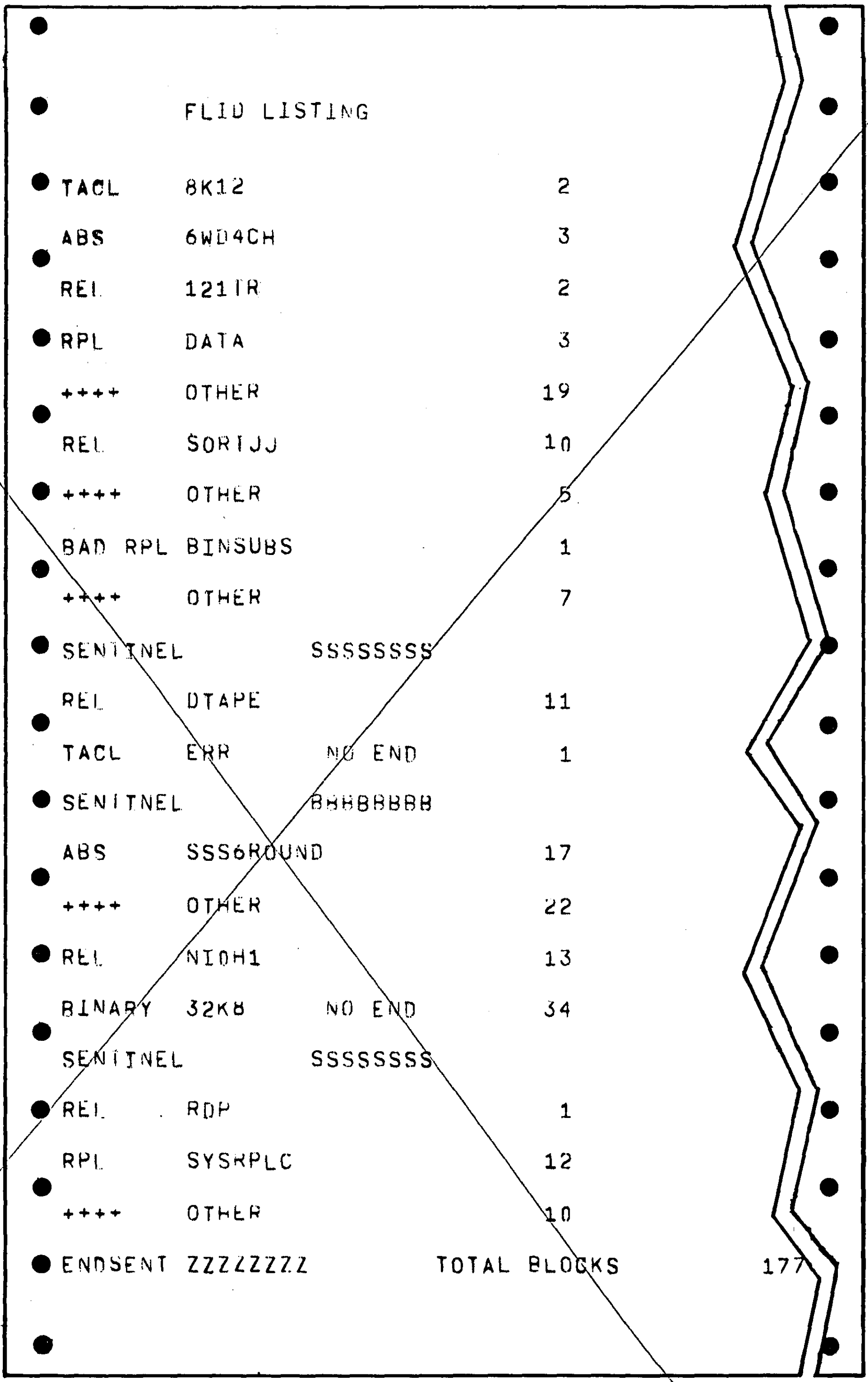
CONTINUED

TAPE 4



* Refer to Remark 2
** Refer to Remark 4
*** Refer to Remark 5
(Page 5-10.5)

HIGH SPEED PRINTER LISTING



```
          FLID LISTING

TACL      8K12                          2
ABS       6WD4CH                        3
REL       121TR                         2
RPL       DATA                          3
++++      OTHER                        19
REL       SORTJJ                       10
++++      OTHER                         5
BAD RPL BINSUBS                         1
++++      OTHER                         7
SENITNEL           SSSSSSSS
REL       DTAPE                        11
TACL      ERR        NO END             1
SENITNEL           BBBBBBBB
ABS       SSS6ROUND                    17
++++      OTHER                        22
REL       N10H1                        13
BINARY    32K8       NO END            34
SENITNEL           SSSSSSSS
REL       RDP                           1
RPL       SYSRPLC                      12
++++      OTHER                        10
ENDSENT ZZZZZZZZ         TOTAL BLOCKS           177
```

OUTPUT FROM A SYSFLID PROGRAM

The tape illustrated on the left contains a selection of programs from which the listing on the right was produced. The control instruction for the listing is: LIST 4,ALL,T. The end sentinel is a block of Z's.

CONSOLE TYPEWRITER ERROR TYPE-OUTS (Continued)

NO.	TYPE-OUT	EXPLANATION
4	TAPE *t* IN LOCAL	Tape *t* is in local operation. The operator should place Tape *t* in automatic mode and press the Advance Bar to continue.
5	TAPE *t* ROCKED 5	A parity or sprocket error persisted on Tape *t* during five successive retries. The operator may press the Advance Bar to continue retrying.
6	TAPE *t* FAULTY	A non-recoverable error was encountered on Tape *t*. Further action is dependent upon the operator.

SERVICE ROUTINE EXAMPLE

Assume that a listing is to be prepared of all TACL programs following a sentinel block of A's on Tape 9, after which Tape 10 is to be positioned at the RPL program SIMPLEX.

```
L   Location   Command    Address and Remarks

               REWIND     9,10               $   (1)
               LOCSENT    9,AAAAAAAA         $   (2)
               LIST       9,TACL,T           $   (3)
               FIND       10,RPL,SIMPLEX     $   (4)
               ENDALL                        $   (5)
```

Explanation — Execution of these instructions causes the following action to take place:

1. Tapes 9 and 10 are rewound.

2. The designated sentinel block, whose first 120 words contain the configuration AAAAAAAA, is located, and Tape 9 is positioned immediately past the sentinel block.

3. A listing of all TACL programs on Tape 9 between the sentinel block of A's and the ending sentinel block of Z's is placed on Tape 5 for off-line printing on the High-Speed Printer.

4. Tape 10 is searched forward until RPL program SIMPLEX or the ending sentinel block of Z's is located. If SIMPLEX is found, the tape will be positioned at the beginning of the program. If the sentinel block of Z's is encountered prior to SIMPLEX, the Console Typewriter will type SIMPLEX MISSING, and control will be transferred to 1ERRDMP (page 4-4).

5. The SYSFLID service routine is terminated and control is returned to SYS.

SECTION VI
TAC LIBRARY ROUTINES

Associated with the Philco Operating System are several subroutines and generators available for placement on the TAC library tape (Tape 7). These subroutines and generators have application only within programs run under control of SYS.

Subroutine descriptions, published separately from this manual, may be inserted in this section for the convenience of the user. For a list of these subroutines, refer to the Philco Catalog of Printed Material.

SYS generators are included with the manual in this section. These generators, such as SNAPGEN and LOADGEN, provide at compilation time, rather than at run time, machine language for specific SYS functions.

CALLING TAC AND ALTAC SUBROUTINES

TAC subroutines are called either by an S in the Label field and the call word placed in the Command field, or by the SUBR pseudo-command in the Command field and the call word in the Address and Remarks field.

ALTAC subroutines are called by the standard CALL statement.

CALLING SYS GENERATORS

SYS generators are called into operation using the call word in the Command field and the parameters in the Address and Remarks field.

LOADGEN — Loading Generator

FUNCTION

Generates machine coding at compilation time, which when executed, causes a program to be loaded via the internal loader. LOADGEN enables a program to be loaded during the running of another program.

FORMAT

For RPL and ABS programs

L	Location	Command	Address and Remarks
		call	*t,id,go,loc* $

For REL programs

L	Location	Command	Address and Remarks
		LOADREL	*t,id,go,list,subs* $

PARAMETERS

call The generator call word which specifies the format of the program to be loaded. One of the following must be selected:

LOADRPL	Indicates that a normal RPL program is to be loaded.
SEGRPL	Indicates that a program in restricted RPL format (page 3-5) is to be loaded.
LOADABS	Indicates that an ABS program is to be loaded.

t A decimal number, 0 through 15, which indicates the tape containing the program to be loaded. It may also be a symbolic address of a location, which at run time, contains the specified tape number in bit positions 20 through 23.

id Eight or fewer alphanumeric characters which indicate the ID of the program to be loaded. It may not be the symbolic location of a word containing the ID.

go (OPTIONAL). Specifies whether or not control is to be transferred to the program just loaded. If GO is written, control will be transferred to the program just loaded. If this parameter is omitted, control will be returned to the calling program after the program specified by the ID has been loaded.

If any other characters are present in the go parameter, the following print-out will appear on the Code-Edit following compilation:

THE REMAINING CHARACTERS IN THIS CALL ARE *xx...x*

where the *x*'s indicate the remaining characters in the Address and Remarks field.

loc (OPTIONAL). A value which may be added to the loading address of the RPL or ABS program to be loaded to indicate the new loading location for this program. The value is assumed to be octal. If any characters other than octal are used, the parameter is assumed to be the symbolic or absolute location of a word containing, in bit positions 1 through 15 at run time, the desired starting address of the program.

list (OPTIONAL). This parameter, written as LIST, indicates that the NAME/SYMBOL list is to be edited for printing and placed on Tape 5 immediately following the successful completion of the loading process.

subs (OPTIONAL). Written as SUBS, this parameter specifies that subroutines for a REL program are to be loaded from Tape 7.

ACTION

During compilation of the object program, TAC replaces the programmer's LOADGEN call with a transfer of control to the associated coding to be generated by LOADGEN.

REMARKS

1. After a program is loaded via LOADGEN, control may be transferred to it, or be returned to the original program.
2. LOADGEN is especially useful for loading segmented programs. (Refer to page 3-5.)
3. The SYS internal loader may also be used at run time via the SYS entry 1INTLD (page 4-20).

EXAMPLE

L	Location	Command	Address and Remarks
		SEGRPL	6,SEG1,GO$

Explanation: Execution of this instruction causes coding to be generated at compilation time, that will load program SEG1, located on Tape 6, into memory at run time in the RPL format produced by the SEGMENT control instruction. Control is transferred to the program immediately after it is loaded. Tape 6 remains positioned at the end of SEG1.

SNAPGEN — Snapshot Generator

FUNCTION

Generates machine coding at compilation time, which when executed, produces selective snapshot dumps during the running of an object program whenever designated conditions are met. SNAPGEN produces the same snapshot dumps as those produced by the SNAP control instruction (page 3-7.2).

CALL WORDS

The SNAPGEN generator may be called either by a SNAP or a SNAPTOG call word. SNAP requests a dump whenever specified conditions are met; SNAPTOG indicates that the snapshot dump is to take place (depending upon conditions) only if a designated toggle switch is in the ON position during the running of a program.

FORMAT

Without the use of toggles

L	Location	Command	Address and Remarks
		SNAP	*format;start;end;cond* $

With the use of toggles

L	Location	Command	Address and Remarks
		SNAPTOG	*tog;format;start;end;cond* $

PARAMETERS

All parameters must be separated by semicolons. The *format* and *cond* parameters are exactly the same as the *format* and *cond* parameters of the SNAP control instruction (page 3-7.2). The remaining parameters, however, are not similar and may not be interchanged.

tog (for SNAPTOG only). A decimal number, 0 through 47, which indicates the toggle switch that must be in the ON position before this particular snapshot dump can occur. The parameter may also be a symbol which must be defined by an ASGN or SAME card at compilation time as a decimal number, 0 through 47.

format An alphabetic character (excluding E and N) which indicates the format in which information within the specified memory locations *start* and *end* are to be dumped. E and N specify a return to SYS. One of the following must be selected:

A	Alphanumeric conversion
C	Command conversion
F	Floating-point conversion
H	Hexadecimal conversion
O	Octal conversion
S	Fixed-point conversion
E	Specifies transfer of control to SYS to perform a post-mortem dump (via 1ERRDMP, page 4-4).
N	Specifies transfer of control to SYS to execute the next SYS control instruction.

start An absolute or symbolic address which indicates the first location to be dumped. Any numeric designation is assumed to to be decimal. Any configuration preceded by an M/ is assumed to be octal. Index registers may be designated by the use of X, such as 1X = Index Register 1.

end An absolute or symbolic address which indicates the last location to be dumped. The same rules for *start* pertain to *end.*

cond (OPTIONAL). Specifies a condition which must be satisfied before a snapshot dump is executed. If the parameter is omitted, the dump is assumed to be unconditional, and a dollar sign must be written following the *end* parameter. If *cond* is specified, it must be followed by a dollar sign. Refer to the *cond* parameter of the SNAP control instruction, page 3-7.2.1, for a listing and explanation of the four types of conditions.

Restrictions. All numeric addresses specified within the *cond* parameter are assumed to be octal. M/ designations may not be used. The letter X may not be used to designate index registers within index register-modified addresses.

ACTION

During compilation of the object program, TAC replaces the programmer's SNAPGEN call with coding which will transfer control to the associated coding to be generated by SNAPGEN. Included in this coding is a snap table (page 3-7.2.1) which contains the information required to perform the snapshot dump, the call upon the snapshot routine, and a return to the object program.

REMARKS

1. SNAP generator calls may not be placed under the influence of an RPT, TIO, or SKC instruction.
2. Upon execution of the generated coding, the contents of the JA Register are not saved.
3. There is no limit to the number of calls within a particular program.

CONSOLE TYPEWRITER ERROR TYPE-OUTS

NO.	TYPE-OUT	EXPLANATION
1	E-SNAP	An error was detected by SNAPGEN during compilation. If errors were found in subsequent SNAPGEN calls, the type-out is not repeated. Coding will be generated, which will enable each call containing an error to be bypassed at run time. For each snap error detected, the error condition will be printed on the Code-Edit, along with the statement, THE REMAINING CHARACTERS IN THIS CALL ARE *xx. . .x,* where the *x*'s indicate the remaining characters, if any, in the Address and Remarks field.

EXAMPLE

```
L  Location  Command   Address and Remarks

                •
                •
                •
             TMA       2052$
             TMD       2047$
             JAED      FINIS$
             SNAP      A;1094;JOE;5/9$
             TMA       XRAY$
             AMS       GEORGE$
             SNAPTOG   1;C;START, 1X;START, 3X;A <3210$
                •
                •
                •
```

Explanation: Execution of the SNAP and SNAPTOG call words inserted into the program to be compiled, sets up the snapshot parameters, so that the designated snap dumps may occur during the running of the program. The SNAP call causes an alphanumeric snapshot dump of decimal location 1094 to memory location JOE the fifth through the ninth time the SNAP call is encountered. The SNAPTOG call causes a command dump of memory location START plus the contents of Index Register 1 through memory location START plus the contents of Index Register 3, if Toggle Switch 1 is set to the ON position and if the contents of the A Register are less than 3210_8.

APPENDIX A
CROSS-INDEX OF CONTROL INSTRUCTIONS, ENTRIES AND SERVICE ROUTINES

Control Instruction	Classification	Page
ABS	Loading	3-4.2
ALPHA	Debugging	3-7.5
ALTAC	Compilation	3-3.2
CLEAR	Special	3-8.3
CLOCK	Special	3-8.4
CMD (COMMAND)	Debugging	3-7.4
COBOL	Compilation	3-3.3
COMMON	Loading	3-4.5
CONIN	Initialization	3-2.2
DUMP	Debugging	3-7.1
HLT	Special	3-8.5
IBIT	Special	3-8.1
JMP (JMPL, JMPR)	Special	3-8.7
JOB	Initialization	3-2.1
JOBSRCH	Special	3-8.2
LOCRPL	Tape Handling	3-6.10
LOCSENT	Tape Handling	3-6.7
LOCTACL	Tape Handling	3-6.9
MASTER	Segmentation	3-5.2
OCT	Debugging	3-7.3
ORIGIN	Loading	3-4.4
PROGTAPE	Special	3-8.8
READB	Tape Handling	3-6.4
READF	Tape Handling	3-6.3
REL	Loading	3-4.3
REM	Special	3-8.6
REPORT	Compilation	3-3.4
REWIND	Tape Handling	3-6.1
REWINDLO	Tape Handling	3-6.2
RPL	Loading	3-4.1
SEGMENT	Segmentation	3-5.1
SNAP	Debugging	3-7.2
TAC	Compilation	3-3.1
WRITE	Tape Handling	3-6.5
WRTABS	Tape Handling	3-6.11
WRTRPL	Tape Handling	3-6.8
WRTSENT	Tape Handling	3-6.6

Classification	Control Instruction	Page
Compilation	ALTAC	3-3.2
	COBOL	3-3.3
	REPORT	3-3.4
	TAC	3-3.1
Debugging	ALPHA	3-7.5
	CMD (COMMAND)	3-7.4
	DUMP	3-7.1
	OCT	3-7.3
	SNAP	3-7.2
Initialization	CONIN	3-2.2
	JOB	3-2.1
Loading	ABS	3-4.2
	COMMON	3-4.5
	ORIGIN	3-4.4
	REL	3-4.3
	RPL	3-4.1
Segmentation	MASTER	3-5.2
	SEGMENT	3-5.1
Special	CLEAR	3-8.3
	CLOCK	3-8.4
	HLT	3-8.5
	IBIT	3-8.1
	JMP (JMPL, JMPR)	3-8.7
	JOBSRCH	3-8.2
	PROGTAPE	3-8.8
	REM	3-8.6
Tape Handling	LOCRPL	3-6.10
	LOCSENT	3-6.7
	LOCTACL	3-6.9
	READB	3-6.4
	READF	3-6.3
	REWIND	3-6.1
	REWINDLO	3-6.2
	WRITE	3-6.5
	WRTABS	3-6.11
	WRTRPL	3-6.8
	WRTSENT	3-6.6

SYS Entry	Page	SYS Entry	Page
Location Zero	4-2	1IGBL	4-14
1BLKIN	4-12	1INTLD	4-20
1ENDJOB	4-6	1MAGRED	4-9
1ERRDMP	4-4	1MAGREW	4-11
1GETBL	4-13	1MAGWRT	4-10

SYS Entry	Page	SYS Entry	Page
1NXTCON	4-7	1SUBERR	4-5
1NXTCRD	4-19	1SYSIN	4-3
1SCAN	4-15	1TYPOUT	4-8
1SCANON	4-16	1XCONER	4-18
1STRIPQ	4-17		

SERVICE ROUTINE CONTROL INSTRUCTIONS

CONTROL INSTRUCTION	SERVICE ROUTINE	PAGE
ADD	RPLC TACSERV	5-6.5 5-9.12
ADDPROG	TACSERV	5-9.4
ANALYZE	ANALYZER	5-3.0
COM	TACLTC	5-8.0
COMPB	AIDE	5-2.2
COMPF	AIDE	5-2.1
COPY	AIDE RPLC TACSERV	5-2.1 5-6.2 5-9.5
COPYCOMP	AIDE	5-2.3
COPYTIL	RPLC TACSERV	5-6.3 5-9.5
CORRECT	RPLC TACSERV	5-6.5 5-9.6
DELETE	RPLC TACSERV	5-6.4 5-9.16
DELPROG	TACSERV	5-9.7
END	BINDEL	5-4.0
ENDALL	AIDE ANALYZER RPLC SYSFLID SYSTRACE TACLTC TACSERV	5-1.1 5-1.1 5-6.9 5-10.6 5-1.1 5-1.1 5-9.8
ENDDATA	DATA	5-5.2
ENDPROG	TACSERV	5-9.18
FIND	SYSFLID	5-10.3
id	SYSTRACE	5-7.1
IDCHANGE	RPLC	5-6.7
LIST	SYSFLID	5-10.4
LOCATE	TACSERV	5-9.9
LOCFLAD	TACSERV	5-9.19
LOCSENT	SYSFLID TACSERV	5-10.1 5-9.9
LOCSEQ	TACSERV	5-9.19
NEWTAPE	RPLC TACSERV	5-6.1 5-9.3
NOSEQ	TACSERV	5-9.20
REPLACE	TACSERV	5-9.20
REWIND	AIDE RPLC SYSFLID TACSERV	5-2.4 5-6.2 5-10.1 5-9.10
REWINDLO	AIDE TACSERV	5-2.4 5-9.10
SENTINEL	SYSFLID TACSERV	5-10.2 5-9.11
SEQ	TACSERV	5-9.22
SKIPTIL	RPLC	5-6.4
TAPE	DATA	5-5.0
WRTSENT	TACSERV	5-9.11

SERVICE ROUTINE	CONTROL INSTRUCTION	PAGE
AIDE (SYSAIDE)	 COMPB COMPF COPY COPYCOMP ENDALL REWIND REWINDLO	5-2 5-2.2 5-2.1 5-2.1 5-2.3 5-1.1 5-2.4 5-2.4
ANALYZER	 ANALYZE ENDALL	5-3 5-3.0 5-1.1
BINDEL	 END	5-4 5-4.0
DATA	 ENDDATA TAPE	5-5 5-5.2 5-5.0
RPLC (SYSRPLC)	 ADD COPY COPYTIL CORRECT DELETE ENDALL IDCHANGE NEWTAPE REWIND SKIPTIL	5-6 5-6.5 5-6.2 5-6.3 5-6.5 5-6.4 5-6.9 5-6.7 5-6.1 5-6.2 5-6.4
SYSFLID	 ENDALL FIND LIST LOCSENT REWIND SENTINEL	5-10 5-10.6 5-10.3 5-10.4 5-10.1 5-10.1 5-10.2
SYSTRACE	 ENDALL *id*	5-7 5-1.1 5-7.1
TACLTC	 COM ENDALL	5-8 5-8.0 5-1.1
TACSERV	 ADD ADDPROG COPY COPYTIL CORRECT DELETE DELPROG ENDALL ENDPROG LOCATE LOCFLAD LOCSENT LOCSEQ NEWTAPE NOSEQ REPLACE REWIND REWINDLO SENTINEL SEQ WRTSENT	5-9 5-9.12 5-9.4 5-9.5 5-9.5 5-9.6 5-9.16 5-9.7 5-9.8 5-9.18 5-9.9 5-9.19 5-9.9 5-9.19 5-9.3 5-9.20 5-9.20 5-9.10 5-9.10 5-9.11 5-9.22 5-9.11

APPENDIX B

PHILCO 2000 CHARACTER CODES

Philco Character	Octal Code	Hollerith Punch	Console Typewriter	
			Upper	Lower
0	00	0	0	?
1	01	1	1	<
2	02	2	2	;
3	03	3	3	e
4	04	4	4	$
5	05	5	5	@
6	06	6	6	(
7	07	7	7	>
8	10	8	8	'
9	11	9	9	#
@	12	8-2 ①	(See LC 5) ②	
=	13	8-3	=	"
;	14	8-4	(See LC 2) ②	
≡	15	8-5 ①	(See LC &) ②	
&	16	8-6 ①	&	≡
'	17	8-7	(See LC 8) ②	
+	20	12	+	*
A	21	12-1	A	a
B	22	12-2	B	b
C	23	12-3	C	c
D	24	12-4	D	d
E	25	12-5	E	e
F	26	12-6	F	f
G	27	12-7	G	g
H	30	12-8	H	h
I	31	12-9	I	i
n ③	32	12-8-2 ①	CAR. RET.	
.	33	12-8-3	.	:
)	34	12-8-4	(See LC %) ②	
%	35	12-8-5 ①	%	)
?	36	12-8-6 ①	(See LC 0) ②	
"	37	12-8-7 ①	SHIFT UPPER	

Philco Character	Octal Code	Hollerith Punch	Console Typewriter	
			Upper	Lower
-	40	11	-	/
J	41	11-1	J	j
K	42	11-2	K	k
L	43	11-3	L	l
M	44	11-4	M	m
N	45	11-5	N	n
O	46	11-6	O	o
P	47	11-7	P	p
Q	50	11-8	Q	q
R	51	11-9	R	r
⅂	52	11-8-2 ①	TAB	TAB
$	53	11-8-3	(See LC 4) ②	
*	54	11-8-4	(See LC +) ②	
<	55	11-8-5 ①	(See LC 1) ②	
#	56	11-8-6 ①	(See LC 9) ②	
⊔	57	11-8-7 ①	SHIFT LOWER	
Δ ③	60	Blank ⑤	SPACE	
/	61	0-1	(See LC -) ②	
S	62	0-2	S	s
T	63	0-3	T	t
U	64	0-4	U	u
V	65	0-5	V	v
W	66	0-6	W	w
X	67	0-7	X	x
Y	70	0-8	Y	y
Z	71	0-9	Z	z
|	72	0-8-2 ①	STOP CODE	
,	73	0-8-3	,	⊔
(	74	0-8-4	(See LC 6) ②	
>	75	0-8-5 ①	(See LC 7) ②	
:	76	0-8-6 ①	(See LC .) ②	
e ③	77	0-8-7 ①	DELETE CODE	

CONTROL CHARACTERS

Character	Console Typewriter	High-Speed Printer	Punched-Card System
n	Carriage Return	Null	Null
⅂	Tab	Cond. Stop ④, Prints ⅂	Cond. Stop ④, End of Block
Δ	Space	Space	Blank Column ⑤
|	Stop Code	Abs. Stop ④, Prints |	Abs. Stop ④, Punches 0-8-2
e	Delete Code	End of Line	End of Card
"	Shift to upper case	Prints "	Punches 12-8-7
⊔	Shift to lower case	Prints ⊔ (logical OR)	Punches 11-8-7

NOTES

① These codes can be punched on the keypunch by multiple punching.

② The octal codes for these Philco Characters are illegitimate on the Console Typewriter. To type these characters from the computer, first transmit a shift to lower case (octal code 57), then transmit code for Philco character indicated in parenthesis.

③ These characters are printed by the Line Printer when in Write - All mode only.

④ These characters are recognized only when the second character in a block in off-line non-data select operation; otherwise second function is performed.

⑤ A switch on the Punched-Card Controller allows blank columns to be read as Philco Character 0 or Δ

APPENDIX C

ALPHABETICAL LISTING OF CONSOLE TYPEWRITER TYPE-OUTS

TYPE-OUT	TYPE *	ORIGINATING PROGRAM	PAGE	EXPLANATION
ADD * WITH NO ID	E	RPLC	5-6	Both an *id* parameter of an ADD * control instruction *and* a dummy PMAX card heading the ABS card deck following the ADD * instruction are missing. Control is transferred to 1XCONER.
ALL CARDS OF FIRST BLK DISAGREED-DEPRESS ADVANCE IF OK	E	TACLTC	5-8	The program executed a halt because none of the twelve cards of the first block of Tape t_1 agreed with their counterparts on Tape t_2. The program will resume its comparison if the Advance Bar is pressed.
BAD CARD # *n*	E	RPLC	5-6	A correction instruction is illegal, where *n* is the sequence number of the instruction, as counted by the SYSRPLC routine, starting from one with each correct instruction. This correction instruction is ignored. Each time the Advance Bar is pressed, the remaining correction instructions are typed out until a new octal location is encountered in the Location field of an instruction, or a new SYSRPLC instruction is encountered. Normal action will be continued if the Advance Bar is pressed again.
BAD CHECKSUM	E	RPLC	5-6	The checksum on one of the cards in the ABS program deck following an ADD * control instruction does not agree with the computed sum. Control is transferred to 1XCONER.
BAD SEC WORD	E	COBOL	3-3.3	One of the compiler phases cannot be loaded during a COBOL compilation, thus indicating a faulty system tape.

*Either Normal or Error Type-Out

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
BAD SECTION WORD	E	RPL	3-4.1	The RPL section just loaded is longer or shorter than specified in the section's control word. Control is transferred to 1ERRDMP (page 4-4).
		ANALYZER RPLC	5-3 5-6	A format error was detected in the RPL input. Control is transferred to 1NXTCRD (page 4-19) for ANALYZER; to 1XCONER (page 4-18) for SYSRPLC.
BAD TAPE	E	RPLC	5-6	A non-recoverable error was detected on a designated tape. All tapes used by SYSRPLC are rewound and the program must be manually terminated. This type-out may occur if a sentinel block is not encountered.
CANNOT FIND JOB *id*	E	JOBSRCH	3-8.2	Indicates that a JOB card with ENDINPUT as its first parameter has been encountered prior to the requested JOB card specified by *id*. Control is transferred to 1SYSIN (page 4-3).
CARD ID	N	TACLTC	5-8	The *mode* parameter of a COM control instruction was either an F (Console Typewriter) or a B (both Typewriter and Tape 5) and differences were detected during the card comparisons. The Identity and Sequence field of the cards from Tape t_1 that differed from Tape t_2 are listed beneath CARD ID.
CHKSUM	E	ANALYZER	5-3	An ABS input card does not have a proper checksum, but it is processed and operation is resumed.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
CHECKSUM ERROR *data*	E	REL ABS	3-4.3 3-4.2	A discrepancy in checksums was found by the SYS loader, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
CLOBDMP	N	Catastrophic Dump	3-7	The SYS catastrophic dump (CLOBDMP) is being executed.
CLOBDMP?	N	Catastrophic Dump	3-7	SYS is about to be initialized; the computer has halted. If a catastrophic dump is desired, the operator should press the Advance Bar. If a catastrophic dump is not desired, the operator should place Toggle Switch 23 in the OFF position and press the Advance Bar.
CLOCK FAILURE	E	CLOCK JOB	3-8.4 3-2.1	The Accounting Clock was interrogated but was unavailable. If this action was initiated by the CLOCK instruction, the instruction is ignored.
COMMON *n*	N	SEGMENT	3-5.1	The program(s) just segmented contain *n* words of Common storage.
COMP. ERRORS	E	TAC ALTAC	3-3.1 3-3.2	One or more serious compilation errors were detected.
COMPARISON COMPLETE	N	TACLTC	5-8	The comparisons specified by the parameters of a COM control instruction have been completed.
CONTROL LINE COMMAND NOT COM	E	TACLTC	5-8	A control instruction which was neither COM nor ENDALL was encountered by the TACLTC routine. Control is transferred to 1XCONER (page 4-18).

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
CONT. LINE ERR.........	N	1XCONER	4-18	An illegal command or illegal parameter (of most control instructions) was detected in a control instruction being interpreted by SYS. An illegal parameter command may also have been detected in the AIDE, ANALYZER or BINDEL Service Routines. Control is transferred to 1ERRDMP (page 4-4).
CONTROL ERROR	E	SYSFLID	5-10	An illegal SYSFLID control instruction was submitted. Control is transferred to 1ERRDMP (page 4-4).
date, time	N	JOB CLOCK	3-2.1 3-8.4	The date and time from the Accounting Clock have been typed in the form: MM-DD HH-MM.T, where MM-DD is the month and day, and HH-MM.T is the hour (per 24-hour day), the minute and tenth of a minute.
DELETION FINISHED, 6 IS BACKUP	N	BINDEL	5-4	The action of service routine BINDEL is completed, and a copy of the original programs on Tape 4 has been placed on Tape 6. The Advance Bar should be pressed to obtain the next job.
DELETION ROUTINE FLEXO?	N	BINDEL	5-4	Enables the operator to set Toggle Switch Zero to ON, if output via the Console Typewriter is desired. The Advance Bar should be pressed to continue computer operation.
DESIGNATED TAPE UNIT IS GREATER THAN 15	E	TACLTC	5-8	The t_1 or t_2 parameter of a COM control instruction specified a number greater than 15. Control is transferred to 1XCONER (page 4-18).
DIFFERENCES	N	TACLTC	5-8	The *mode* parameter of a COM control instruction was a T (tape output only), and differences were detected during the card comparisons. The differences are written onto Tape 5.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
DUMP TABLE FULL	E	DUMP	3-7.1	More than ten DUMP instructions were given within one job. All dump calls over ten are ignored.
DUMP ERR	E	DUMP	3-7.1	An Illegal dump parameter was encountered by SYS. The dump card is ignored.
EALTAC	E	ALTAC	3-3.2	The ALTAC compiler detected one or more source language errors that would cause generation of an incorrect object program. The TAC assembly phase is bypassed, and control is returned to SYS via 1NXTCON.
ECOBOL	E	COBOL	3-3.3	The COBOL compiler detected one or more source language errors that would cause generation of an incorrect object program. The TAC assembly phase is bypassed, and control is returned to SYS via 1NXTCON.
E-SNAP	E	SNAPGEN	6-3	An error was detected by SNAPGEN during compilation. If errors were found in subsequent SNAPGEN calls, the type-out is not repeated. Coding will be generated, which will enable each call containing an error to be bypassed at run time. For each snap error detected, the error condition will be printed on the Code-Edit, along with the statement, THE REMAINING CHARACTERS IN THIS CALL ARE *xx...x* where the *x*'s indicate the remaining characters, if any, in the Address and Remarks field.
END OF JOB	N	1ENDJOB	4-6	A job has been terminated.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
END OF TAC LANGUAGE TAPE COMPARISON	N	TACLTC	5-8	An ENDALL control instruction was encountered by TACLTC, and control is transferred to 1NXTCON (page 4-7).
ERRDMP 0 ERRDMP 2 ERRDMP 3	N	1ERRDMP	4-4	A programmed transfer of control to Location Zero, 1ERRDMP, or 1SUBERR was executed, or a program malfunction was detected (either by the program or operator) and control was transferred to Location Zero, 1ERRDMP, or 1SUBERR (pages 4-2, 4-4, and 4-5).
EXCEEDS MEMORY *data*	E	REL	3-4.3	The program to be loaded (with subroutines) exceeds memory capacity, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
FINIS	N	AIDE	5-2	The operation requested by a COPY, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine has been completed. No errors or inequalities were detected during the COMPB or COPYCOMP comparisons.
FORM *nn*	E	DUMPCON	2-3.4	An illegal conversion code has been submitted to the DUMPCON routine, where *nn* is the illegal code in octal. Control is transferred to 1ENDJOB.
FORM OF LOAD IS NOT RPL, ABS, OR REL	E	1INTLD	4-20	Occurs if neither Bit 45, Bit 46, nor Bit 47 has been set to one when the internal loading feature of SYS is used. Control is transferred to 1ERRDMP (page 4-4).
format ANALYZE *id*	N	ANALYZER	5-3	A program with a format of *format* and an ID of *id* is about to be analyzed.

TYPE-OUT		ORIGINATING PROGRAM	PAGE	EXPLANATION
FOUND JOB	N	JOBSRCH	3-8.2	Indicates that the specified JOB card has been located. The name of the job is then typed on the Console Typewriter and the job is initiated.
ID *id* ABS CDS *n* ON TAPE *t*	N	WRTABS	3-6.11	An ABS program designated by *id,* containing *n* decimal cards, has been written onto Tape *t.*
ID *id* RPL BLX *n* ON TAPE *t*	N	WRTRPL	3-6.8	An RPL program designated by *id,* containing *n* decimal blocks, has been written onto Tape *t.*
id MISSING	E	COBOL	3-3.3	One of the compiler phases was not correctly identified during a COBOL compilation, thus indicating a faulty system tape.
		ANALYZER	5-3	An RPL or ABS program specified by *id* was not encountered during a forward and backward search of Tape *t.* The current ANALYZE control instruction is ignored.
		SYSFLID	5-10	A specified sentinel block was encountered prior to a designated program *id* during execution of a FIND control instruction. Control is transferred to 1ERRDMP (page 4-4).
id NOT HERE	E	RPL ABS REL LOCTACL LOCRPL	3-4.1 3-4.2 3-4.3 3-6.9 3-6.10	A program specified by *id* cannot be found on a designated tape. Control is transferred to 1ERRDMP (page 4-4), for RPL, ABS, and REL; for LOCTACL and LOCRPL, control is transferred to 1XCONER (page 4-18).
id RPL *n* BLOCKS	N	SEGMENT	3-5.1	A program has just been written on tape in RPL format, identified as *id* and containing *n* blocks.
ILLEGAL BLK CNT	E	AIDE	5-2	A decimal number larger than 19,000 has been specified in a *nbp* parameter of a COPY, COMPF, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine. SYSAIDE will immediately return control to SYS.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
ILLEGAL CARD	E	ANALYZER	5-3	A card in an ABS input deck is not in ABS format. Control is transferred to 1XCONER (page 4-18) if input is from Tape 8; otherwise, control is transferred to 1NXTCRD (page 4-19).
ILLEGAL CARD *data*	E	ABS REL	3-4.2 3-4.3	A card unacceptable to the SYS loader has been encountered, where *lata* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
ILLEGAL 1ST CARD *data*	E	REL	3-4.3	The first card of a program to be loaded was not a PMAX or TUG absolute card, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
ILLEGAL CSA *data*	E	REL ABS	3-4.3 3-4.2	An attempt was made to load a program into an illegal memory address, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
ILLEGAL INPUT TAPE NO.	E	ALTAC	3-3.2	An attempt was made to use Tape 6 as an input tape. Control is transferred to 1ERRDMP.
		COBOL	3-3.3	One of the following occurred during a COBOL compilation: • An attempt was made to use Tape 6 as an input tape. • An attempt was made to use Tape 9 as an input tape *and* CONBITS 19 is set to one. • An attempt was made to use Tape 10 as an input tape *and* CONBITS 20 is set to one. Control is returned to SYS via 1ERRDMP.
		REPORT	3-3.4	An attempt was made to use Tape 10 as an input tape during a REPORT compilation, *and* CONBITS 20 is set to one. Control is returned to SYS via 1ERRDMP.
ILLEGAL LOAD ADDRESS	E	RPL	3-4.1	An attempt was made to load a program into memory occupied by SYS. Control is transferred to 1ERRDMP (page 4-4).

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
ILLEGAL MODIFIER *data*	E	REL	3-4.3	An unacceptable address modifier has been encountered by the SYS loader, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
ILLEGAL OUTPUT PARAMETER	E	TACLTC	5-8	The *mode* parameter of a COM control instruction is not T, F, or B. Control is transferred to 1XCONER (page 4-18).
ILLEGAL SUBLIB TAPE NO.	E	REL	3-4.3	Tape 1, 5, or 8 (if input via magnetic tape) was specified as a binary library tape in a REL control instruction. Control is transferred to 1ERRDMP (page 4-4).
ILLEGAL T OR O	E	READF READB WRITE	3-6.3 3-6.4 3-6.5	The *t* parameter is 16 or greater, or the *addr* parameter is non-octal.
		PROGTAPE	3-8.8	The *t* parameter is 16 or greater.
ILLEGAL TAPE NUMBER	E	AIDE	5-2	Either an alphabetic character or a decimal number larger than 15 was entered as a *from, to,* t_1, or t_2 parameter of a COPY, COMPF, COMPB, or COPYCOMP control instruction within the SYSAIDE service routine. SYSAIDE will immediately return control to SYS.
INPUT TAPE IS CODE, CANNOT TRANSFER IN IMAGE	E	DATA	5-5	An illegal conversion request was made by the TAPE instruction. The *mode* parameter is BINARY or IMAGE where input data is in Code Mode. Control is transferred to 1XCONER (page 4-18).
JOB CARD INTERCEPTED	E	1NXTCRD	4-19	A JOB card was intercepted by the 1NXTCRD subroutine. A post-mortem dump (refer to 1ERRDMP, page 4-4) will occur, following which the job specified by the intercepted JOB card will be processed. This type-out may also occur via a BINDEL or DATA service routine.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
JOB *id*	N	JOB	3-2.1	The job specified by *id* is ready to be executed.
LOAD ERR	E	1INTLD	4-20	The internal load function did not receive the proper indication of the type of program to be loaded (RPL, ABS, or REL). Control is transferred to 1ERRDMP.
LOGICAL TAPE UNIT *t* IN LOCAL. CORRECT AND DEPRESS ADVANCE	E	TACLTC	5-8	Tape *t* is in local operation. The program will resume its comparison if the tape is placed in automatic mode and the Advance Bar is pressed.
MACHERR	E	COBOL	3-3.3	A machine or compiler error was detected during a COBOL compilation. The computer halts following this type-out and the operator must manually return control to SYS.
MASTER ORIGIN *addr*	N	MASTER	3-5.2	A Master segment has been loaded into memory and its origin has been stored in MASORG as an octal *addr*.
NEWTAPE CARD NOT SOON ENOUGH	E	RPLC	5-6	A control instruction other than a REWIND was encountered prior to a NEWTAPE instruction by the SYSRPLC service routine. Control is transferred to 1XCONER (page 4-18).
NO DIFFERENCES		AIDE	5-2	The operation requested by a COMPF control instruction has been completed, and no errors or inequalities were detected.
		TACLTC	5-8	The *mode* parameter of a COM control instruction was a T, and no differences were detected during the card comparisons.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
NO LIB	E	COBOL	3-3.3	The COBOL library, required by a COBOL source program, is not mounted on the correct tape.
NO MASTER	E	SEGMENT	3-5.1	A master parameter is present in a SEGMENT control instruction, but no MASTER control instruction has been encountered. Control is transferred to 1XCONER (page 4-18).
NO ROCK	E	AIDE	5-2	A possible error in the SYSAIDE program was detected. An error dump will be provided by SYS. (This type-out should not normally occur.)
NO WRITE RING ON LOGICAL TAPE 5. CORRECT AND DEPRESS ADVANCE.	E	TACLTC	5-8	Tape 5 does not have a Write Ring. The program will resume its comparison, if a Write Ring is placed on Tape 5 and the Advance Bar is pressed.
NONRECOVERABLE TAPE ERROR	E	TACLTC	5-8	A PROC-detected non-recoverable tape error occurred. Control is transferred to 1XCONER (page 4-18).
NOT HERE	E	RPLC 1INTLD	5-6 4-20	The program designated by *id* could not be found during a search of Tape *t*. Control is transferred to 1ERRDMP (page 4-4).
NUMBER OF BLOCKS TO BE COMPARED IS EQUAL TO 0	E	TACLTC	5-8	The *nbp* parameter of a COM control instruction specified 0. Control is transferred to 1XCONER (page 4-18).

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
NUMBER OF BLOCKS TO BE COMPARED IS GREATER THAN 19000	E	TACLTC	5-8	The *nbp* parameter of a COM control instruction specified a number greater than 19,000. Control is transferred to 1XCONER (page 4-18).
PROG. XFERRED TO PROGTAPE, T5	N	TAC ALTAC	3-3.1 3.3.2	The compiled program was transferred to the program tape and the information for binary cards was transferred to Tape 5.
PROG. NOT XFERRED TO PROGTAPE, T5	N	TAC ALTAC	3-3.1 3-3.2	The compiled program was not transferred to the program tape and information for binary cards was not transferred to Tape 5 because of compilation errors or IBIT settings.
PROG. XFERRED TO PROGTAPE ONLY	N	TAC ALTAC	3-3.1 3-3.2	The compiled program was transferred to the program tape. Information for binary cards was not transferred to Tape 5.
PROG. XFERRED TO T5 ONLY	N	TAC ALTAC	3-3.1 3-3.2	Information for binary cards of a compiled program was transferred to Tape 5 but the program was not transferred to the program tape.
PROGTAPE = *t*	N	PROGTAPE	3-8.8	A PROGTAPE instruction has been executed and all subsequent programs within the current job will be transferred to Tape *t* following compilation subject to normal IBIT settings (refer to page 3-3.1.1).
READY	N	RPL ABS REL	3-4.1 3-4.2 3-4.3	A program has just been loaded and is ready for execution.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
REWIND NOT ACCEPTED AFTER TIO	E	TACLTC	5-8	A PROC-detected error has occurred. Control is transferred to 1ERRDMP (page 4-4).
RPLC	N	RPLC	5-6	The SYSRPLC service routine has been initiated. No operator action is necessary.
SNAP ERR. AT = *p addr*	E	SNAP	3-7.2	Illegal parameter(s) are contained in a SNAP control instruction, where *p* is the label and *addr* is the octal address of the instruction which initiated the SNAP routine. The SNAP call containing the illegal parameter is ignored.
SPROCKET OR PARITY ERROR, DEPRESS ADVANCE TO ROCK TAPE 5 MORE TIMES	E	TACLTC	5-8	A parity or sprocket error was detected and five error correction attempts failed. Each time the Advance Bar is pressed, five additional correction attempts will be made. If the error remains uncorrected, control can be transferred manually to SYS.
SYMBOL REDEFINED *data*	E	REL	3-4.3	A program name symbol has been redefined at load time. Control is transferred to 1ERRDMP (page 4-4).
SYS *n* VERSION	N	1SYSIN	4-3	SYS has been initialized, where *n* is the version of SYS being used.
TAC COMPILATION BYPASSED	N	COBOL	3-3.3	IBIT 36 is set to one. The COBOL compiler produced a TAC program, but bypassed the TAC compilation phase. The TAC program is on Tape 6. Control is returned to SYS via 1NXTCON.
TACLTC-TAC LANGUAGE TAPE COMPARISON	N	TACLTC	5-8	The TACLTC service routine has been initiated.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
TACS E*n*	E	TACSERV	5-9	An error has been detected by the TACSERV service routine, where *n* indicates the type of error as described on page 5-9.26.
TAPE *t* FAULTY	E	TAC ALTAC ANALYZER DATA SYSFLID	3-3.1 3-3.2 5-3 5-5 5-10	A non-recoverable error was encountered on Tape *t*. Further action is dependent upon the operator.
TAPE *t* IN LOCAL	E	TAC ALTAC ANALYZER DATA SYSFLID	3-3.1 3-3.2 5-3 5-5 5-10	Tape *t* is in local operation. The operator should place Tape *t* in automatic mode and press the Advance Bar to continue.
TAPE *t* ROCKED 5	E	TAC ALTAC ANALYZER DATA SYSFLID	3-3.1 3-3.2 5-3 5-5 5-10	A parity or sprocket error persisted on Tape *t* during five successive retries. The operator may press the Advance Bar to continue retrying.
TAPE *t* WR RING	E	TAC ALTAC ANALYZER DATA SYSFLID	3-3.1 3-3.2 5-3 5-5 5-10	There is no Write Ring on Tape *t*. To continue the operator may insert a Write Ring and press the Advance Bar.
TAPE NOT RECOGNIZED	E	DATA	5-5	The first card supplied to the DATA routine does not contain the word TAPE in the Command field. Control is transferred to 1XCONER (page 4-18).
TAPE SELECTED FOR DATA ILLEGAL	E	DATA	5-5	A decimal number other than 0, 3, 6, 7, or 9 through 15 was specified as Tape *t*. Control is transferred to 1XCONER (page 4-18).

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
TAPES FOR COMPARISON HAVE SAME TAPE NO.	E	TACLTC	5-8	The t_1 and t_2 parameters of a COM control instruction have specified the same tape. Control is transferred to 1XCONER (page 4-18).
TOO MANY SYMBOLS *data*	E	REL	3-4.3	The NAME/SYMBOL list of the relocatable loader has been filled to capacity, where *data* indicates Columns 1-8 of the illegal card. Control is transferred to 1ERRDMP (page 4-4).
TOTAL BLKS COPIED *n*	N	RPLC	5-6	The SYSRPLC service routine has been terminated and the total number of blocks transferred to the output tape is indicated by *n*.
U. N. A. *t*	E	AIDE	5-2	SYSAIDE attempted to issue an order to Tape *t* which was not available. If the Advance Bar is pressed, SYSAIDE will wait until operator intervention makes Tape *t* available and then will continue normal operation.
UNDEFINED SYMBOL	E	REL SEGMENT	3-4.3 3-5.1	The program just loaded contains an undefined symbol. For REL, control is transferred to 1ERRDMP (page 4-4); for SEGMENT, control is transferred to 1ENDJOB (page 4-6).
UNIT NOT AVAILABLE	E	RPLC	5-6	A designated tape is not available, or there is no Write Ring on the output tape. The operator may correct the condition and press the Advance Bar to continue.

TYPE-OUT	TYPE	ORIGINATING PROGRAM	PAGE	EXPLANATION
UNIT 4 SELECTED FOR COMPARISON DEPRESS ADVANCE IF OK	E	TACLTC	5-8	The t_1 or t_2 parameter of a COM control instruction specified Tape 4. The program executed a halt. Because Tape 4 is the User's Program Tape, it is highly probable that its comparison may not be desired. If Tape 4 is to be compared, the Advance Bar should be pressed.
UNIT (1, 5, OR 8) SELECTED FOR COMPARISON	E	TACLTC	5-8	The t_1 or t_2 parameter of a COM control instruction specified Tape 1, 5 or 8 (illegal tapes). (Tape 8 is illegal if control input is via Tape 8.) Control is transferred to 1XCONER (page 4-18).
WRAPUP?	N	JOB	3-2.1	A JOB card is read with toggle switch *n* set, where *n* is the switch used by the installation for wrapup (usually 25).

APPENDIX D

THE GENERAL SYSTEM TAPE

SYS E

The General System Tape contains those programs in RPL, ABS, and TAC language format needed in the Philco Operating System. Its contents are listed as follows:

SYS IMAGE, RPL, AND ABS PROGRAMS

TITLE	PROGRAM	BLOCKS
SYS E	(SYS IMAGE)	40
00000000	(SENTINELS)	4
SYSJOBSΔΔΔ100862	RPL	11
DUMPCONΔΔΔ121862	RPL	11
PREPTACII2120662	RPL	10
SYSALTACΔ8KΔ0962	RPL	36
SYSALTAC16KΔ0962	RPL	36
SYSALTAC32KΔ0962	RPL	36
SYSTAC2ΔΔΔ120662	RPL	40
SYSPASS2TAC21262	RPL	29
DATA ΔΔΔΔΔΔ070362	RPL	3
TACSERVSΔΔ060762	RPL	9
SYSAIDEΔΔΔ121361	RPL	11
SYSRPLCΔΔΔ100862	RPL	12
ANALYZERΔΔ092462	RPL	12
SYSFLIDΔΔΔ080162	RPL	8
CHECK8ΔΔΔΔ062062	RPL	4
BINDELΔΔΔΔ012562	RPL	11
SYSTRACEΔΔ092062	RPL	9
JOBSRCHΔΔΔ121361	RPL	1
SYSPLUMΔΔΔ121362	RPL	15
TACLTCΔΔΔΔ010562	RPL	8
LOCATORΔΔΔ061562	RPL	2
COMPACTΔΔΔ101962	RPL	4
SYSALT2ΔΔ8KΔ0504	RPL	37
SYSALT2Δ16KΔ0504	RPL	37
SYSALT2Δ32KΔ0504	RPL	37
SYSTAC2ΔΔΔ120662	RPL	40
SYSPASS2TAC21262	RPL	29
CLOBBERΔΔΔ111362	ABS	4
ISYSΔEΔΔΔΔ121762	ABS	24
IISYSΔEΔΔΔ121762	ABS	19
ZZZZZZZZ	(SENTINEL)	1
TOTAL		590

TACL PROGRAMS

TITLE	PROGRAM	BLOCKS
CLOBBERΔΔΔ111362	TACL	39
ISYSΔEΔΔΔΔ121762	TACL	362
IISYSΔEΔΔΔ121762	TACL	289
SYSJOBSΔΔΔ100862	TACL	30
DUMPCONΔΔΔ121862	TACL	58
PREPTACII2120662	TACL	81
SYSALTACΔ8KΔ0962	TACL	667
SYSALTAC16KΔ0962	TACL	667
SYSALTAC32KΔ0962	TACL	667
SYSTAC2ΔΔΔ120662	TACL	683
SYSPASS2TAC21262	TACL	479
DATA ΔΔΔΔΔΔ 070362	TACL	24
TACSERVSΔΔ060762	TACL	148
SYSAIDEΔΔΔ121361	TACL	101
SYSRPLCΔΔΔ100862	TACL	160
ANALYZERΔΔ092462	TACL	192
SYSFLIDΔΔΔ080162	TACL	93
CHECK8ΔΔΔΔ062062	TACL	37
BINDELΔΔΔΔ012562	TACL	51
SYSTRACEΔΔ092062	TACL	130
JOBSRCHΔΔΔ121361	TACL	11
SYSPLUMΔΔΔ121362	TACL	240
TACLTCΔΔΔΔ010562	TACL	72
LOCATORΔΔΔ061562	TACL	32
COMPACTΔΔΔ101962	TACL	60
ZZZZZZZZ	(SENTINEL)	1
	TACL TOTAL	5374
	RPL TOTAL	590
	TOTAL BLOCKS ON TAPE (including sentinel blocks)	5964

DESCRIPTION OF CODE-EDIT TAPE

The Code-Edits appear in the same order and with the same ID's as programs in TAC language format listed on the General System Tape. Each Code-Edit on the tape is preceded by an index block which may be printed in Data Select 9. The index blocks are provided as an aid to positioning the tape to a specific Code-Edit. Each index block contains a listing of the next ten Code-Edits on the tape and is terminated with a conditional stop. Following the conditional stop, the tape is positioned at the beginning of the Code-Edit appearing at the top of the list. As these blocks are printed using Data Select 9, there is no interference with the printing of the Code-Edits which use Data Select 0.

Each time an index block is printed, the list is moved ahead one ID. To print a particular Code-Edit, the operator should continuously print the index blocks in Data Select 9 until the desired ID appears at the top of the list. The Code-Edit may then be printed using Data Select 0.

As an example, an index block, when printed, may appear in the following format:

THE NEXT TEN CODE-EDITS ARE

```
SYSJOBS       062662
DUMPCON       062462
PREPTACII 2070362
SYSALTAC8K    0504
SYSALTAC16K 0504
SYSALTAC32K 0504
SYSTAC2       041162
SYSPASS2TAC20462
DATA          070362
TACSERVS      060762
```

If printing is continued using Data Select 9, the next index block will follow, similar to the first except that DUMPCON will be moved to the top of the list. The next Code-Edit following will be added as the tenth ID.

To print the Code-Edit of DATA, the operator should continue to print the index blocks using Data Select 9 until the ID DATA appears at the top of the list. This will be the ninth index block in this case. The operator should then switch to Data Select 0 and commence printing the Code-Edit of DATA (and the remainder of the Code-Edits on the tape if desired). If it is desirable to skip additional Code-Edits, Data Select 9 should be used until the tape is properly positioned.

PRODUCTION OF INSTALLATION SYSTEM

A binary deck of the program *zz*MS in ABS format (where the *z*'s indicate the current version of SYS) is provided for installation use to produce a particular System tape from the master. Installation options for 8,192 (8K), 16,384 (16K), or 32,768 (32K) word memory and/or Accounting Clock may be selected under toggle control by *zz*MS as described below. The System may also be preset to assume the MAGTAPE, IMAGE mode (instead of FLEXO mode) upon initialization, thus eliminating the typing of CONIN IMAGE each time the System is initialized.

The sentinels of zeroes within the third and fourth blocks following the SYS Image portion (page 2-2.0) of the new System tape (listed on page D-1) are changed by zzMS to indicate the System version produced. (Refer to the example within System Identities, below.) The *zz*MS program also provides a SYSFLID listing of the new System tape. The version of the System produced is indicated in the sentinels within the listing.

Whenever the program *zz*MS is run, toggle settings are requested via the Console Typewriter as indicated in the following table:

TOGGLE	SYSTEM OPTION
0	ON for Accounting Clock OFF for no Accounting Clock
1	ON for CONIN IMAGE. (Initialized input OFF for CONIN FLEXO. mode)
8, 16, or 32	Memory size, indicated by <u>one</u> of the three toggles set to ON. If more than one of these toggles is set, or if an illegal toggle is set, TOGS SET WRONG will be typed on the Console Typewriter. The operator should then reset the toggles properly, and press the Advance Bar to continue.

The General System Tape must be mounted on Unit 1 with a scratch tape on Unit 3. The program copies the General System Tape from Tape 1 to Tape 3, making the internal changes for selected options and the program selections as it goes. At program completion, the General System Tape is rewound, the installation System tape on Unit 3 is rewound with lockout and the computer halts. Additional installation System tapes may be made by mounting a new scratch tape on Unit 3 and pressing the Advance Bar.

System Identities

The 16-character version identity of any particular System is determined by the selection of toggle options (as described previously). The System base identity, contained with the first eight characters, is SYS*zz*. The full identity, which is the version type-out during initialization (refer to 1SYSIN, page 4-2) is determined by the tables below:

VERSION TYPE-OUT
SYS *zz*ΔΔΔΔ*nn*KΔΔ*xy*
where the *z*'s indicate the current version of SYS, the *n*'s indicate machine size, *x* indicates whether CONIN IMAGE or CONIN FLEXO is selected, and *y* indicates whether or not a clock is present.

TOGGLE OPTIONS SELECTED	CHARACTERS POSITION IN ID	CHARACTER
8K	11th and 12th	8K
16K	10th through 12th	16K
32K	10th through 12th	32K
CONIN FLEXO	15th	F
CONIN IMAGE	15th	M
CLOCK	16th	C
NO CLOCK	16th	Δ

Example: A System with a 32,768-word memory and an Accounting Clock, set to assume MAGTAPE, IMAGE mode, will have the version identity: SYSΔZ ΔΔΔΔ32KΔΔMC. The four sentinel blocks for this particular System, which follow the SYS Image portion on the SYSΔZ master tape, appear on the SYSFLID listing as:

```
SENTINEL     00000000
SENTINEL     00000000
SENTINEL     0000SYSZ
SENTINEL     00032KMC
```

Console Typewriter Normal Type-Outs

NO.	TYPE-OUT	EXPLANATION
1	SET TOGS - 8, 16, OR 32 FOR MACH. SIZE - 0 UP FOR CLOCK - 1 UP FOR CONIN IMAGE ASSUMED	Set toggles to indicate desired options, press Advance Bar.

CONTINUED

Console Typewriter Normal Type-Outs (Continued)

NO.	TYPE-OUT	EXPLANATION
2	TAPE 3 IS SYS TAPE	This type-out will be followed by a HLT. Either mount a new scratch on Unit 3 and press the Advance Bar to produce another System, or transfer control to 1ENDJOB (page 4-6).

Console Typewriter Error Type-Out

NO.	TYPE-OUT	EXPLANATION
1	TOGS SET WRONG	Reset toggles properly, press Advance Bar to retry.

Example

The following procedure will produce the required System for a 32,768-word memory computer with an Accounting Clock, and initialize SYS to the MAGTAPE, IMAGE mode of input:

1. Mount tapes (General System Tape on Unit 1 scratch tape on Unit 3).

2. Initialize SYS*zz* (on the General System Tape) and execute the following job from Tape 8:

```
L   Location   Command   Address and Remarks

               JOB       NEWSYS
               ABS       *,,GO
                 }
                         (ABS deck of zzMS)
```

3. Set Toggle Switches 0, 1 and 32 to ON when *zz*MS requests toggle settings via the Console Typewriter.

4. Dismount the tape from Unit 3 when the System halt occurs. (Refer to HLT, Remark 3, page 3-8.5.)

APPENDIX E

SYS OPERATING PROCEDURES

In order to provide operators with pertinent information concerning SYS, this interim appendix is presented until a complete section on SYS Operating Procedures is released.

Appendix E contains instructions for the initialization of SYS and execution of the SYS catastrophic dump (CLOBDMP). The information concerning the two subjects is complete.

INITIALIZATION OF SYS

In order to initialize the Philco Operating System (SYS), the operator should perform the following operations:

1. Mount the magnetic tapes according to the assignments listed below:
   - Tape 0 — Unassigned
   - Tape 1 — SYS Program Tape
   - Tape 2 — SYS Intermediate Tape (Write-Enabled)
   - Tape 3 — Compiler Scratch Tape (Write-Enabled)
   - Tape 4 — User's Program Tape (Write-Enabled)
   - Tape 5 — SYS Output Tape (Write-Enabled)
   - Tape 6 — Compiler Scratch Tape (Write-Enabled)
   - Tape 7 — TAC Library Tape
   - Tape 8 — SYS Input Tape
   - Tapes 9-15 — Unassigned
2. Place the computer into STEP mode.
3. Rewind Tape 1.
4. Read one block from Tape 1 into Memory Location Zero.
5. Check to see that Toggle Switch 23 is <u>not</u> set.*
6. If SYS is to be initialized to the standard mode of input (page 4-3), do not set Toggle 47; if SYS is to be initialized to the FLEXO mode of input, set Toggle 47.
7. Execute a JMPL to Memory Location 1.
8. Place the computer into RUN mode.
9. Press the Advance bar.
10. SYS *n* VERSION will be typed on the Console Typewriter, where *n* is the version of SYS being used at a particular installation.

---

* If, when trying to initialize SYS, Toggle 23 is set, CLOBDMP? will be typed on the Console Typewriter and the computer will halt (see Action 8, page E-4). The operator at this point should place Toggle 23 into the OFF position and then press the Advance bar. SYS will then be initialized without executing the catastrophic dump.

11. SYS then calls for a control instruction from the mode of input indicated in step 6.

12. If the System is operating in any input mode other than FLEXO, SYS will proceed automatically; if the System is operating in FLEXO mode, the operator should enter the control instructions via the Console Typewriter.*

13. If the System is operating in the FLEXO mode of input, and it is desired to change the mode to magnetic tape or paper tape, the operator should enter the new source of control information via the Console Typewriter as follows (where (t) indicates tab settings and (c) indicates carriage return settings):

    - If magnetic tape is desired:

```
                              CODE
(t)   (t)   CONIN   (t)        or      (c)
                              IMAGE
```

    - If paper tape is desired:

```
(t)   (t)   CONIN   (t)   PAPER   (c)
```

    For a further explanation of CONIN, refer to page 3-2.2.

* If a typing error should occur, the entire instruction may be reentered by pressing the STOP CODE key. In response to this action, SYS issues a carriage return and requests a new type-in via the white light signal.

EXECUTION OF A SYS CATASTROPHIC DUMP

In order to execute a SYS catastrophic dump (CLOBDMP), assuming MAGTAPE input mode, the operator should perform the following actions:

1. Place the computer into STEP mode.
2. Record the D and JA Registers.
3. Write one block from Location Zero onto Tape 2.
4. Backspace Tape 8 one block.
5. Rewind Tape 1.
6. Read one block from Tape 1 into Location Zero.
7. Set Toggle Switch 23.
8. Execute a JMPL to Location 1, which will cause CLOBDMP? to be typed on the Console Typewriter.
9. Place the Computer into RUN mode.
10. Press the Advance bar.

At this point in the program, the catastrophic dump automatically performs the following operations:

- CLOBDMP is typed on the Console Typewriter.
- Seven blocks are written onto Tape 5.
- The remaining three blocks of the catastrophic program are read from Tape 1 into memory.
- Tape 4 is rewound.
- The information saved on Tapes 2 and 5, plus the remainder of memory, is written onto Tape 2 in post-mortem dump format.
- SYS is reloaded from Tape 1 and the information on Tape 2 is converted to the specified format and is placed on Tape 5 edited for output.
- SYS is then initialized automatically.

INDEX