

UNCLASSIFIED

AD 277 414

*Reproduced
by the*

**ARMED SERVICES TECHNICAL INFORMATION AGENCY
ARLINGTON HALL STATION
ARLINGTON 12, VIRGINIA**



UNCLASSIFIED

NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

ENCORES FOR ACT I SYSTEM FOR LGP-30 DIGITAL COMPUTER COMPILER

Alan H. Rich

Propagation Branch
Sound Division

June 1, 1962



JUL 5 1962
62-61-1
ASTIA
A

U. S. NAVAL RESEARCH LABORATORY
Washington, D.C.

CATALOGED BY ASTIA
AS AD NO. 277414

277 414

CONTENTS

Abstract	ii
Problem Status	ii
Authorization	ii
INTRODUCTION	1
PURPOSE	1
ALTERATIONS TO THE COMPILER	2
THE OPERATIONS TABLE	2
"CAST OF CHARACTERS WITH REFERENCES ON REQUEST"	2
"UNDERSTUDIES"	4
"OUTPT" - "ADDR" - "ROUND" SUBROUTINE PACKAGES	4
PACKAGING	5
SUMMARY	6
REFERENCES	6

ABSTRACT

The Algebraic Compiler and Translator I System for the LGP-30 digital computer has been modified and expanded for greater versatility. A location symbols table is generated at a new location, providing space for one of two decimal program and references printout routine. A 63-symbol operations table includes non-input typewriter operation symbols, breakpoint symbols, a compatible hex or alpha-numeric output symbol, a binarization of address symbol, a flexible program input symbol, an integer-trim symbol, and a trace-one-intermediate-result symbol. Two alternate subroutines, located in unused space at the time of computing, utilize program input routine.

PROBLEM STATUS

This is an interim report on the problem; work is continuing.

AUTHORIZATION

NRL Problem S01-01
Project RR 004-07-41-5250

Manuscript submitted April 6, 1962.

ENCORES FOR ACT I SYSTEM FOR LGP-30 DIGITAL COMPUTER COMPILER

INTRODUCTION

Modifying and supplementing subroutines have been written for the LGP-30 digital computer compiler system, Act I (1,2). Called "Encores," the subroutines include compiler modifications which cause the symbol table to be stored and punched out in 6000-6163 after the object program. This modification provides space for either "Cast of Characters with References on Request" or "Understudies," special decimal printout routines, to be stored in 5800-5963. Headings for each of these subroutines are printed by programs in 3932-3963, stored following an expanded operations table. The additional subroutines "outpt," "addr," and "round" fit in 6240-6263 and provide compatible hexadecimal or alphanumeric output, binarization of addresses at a q of 29, and rounding off the fractional portion of numbers at a q of 29, respectively.

PURPOSE

One reason for expanding the already well-integrated Act I system was to help overcome semantics barriers in order that both programmers and scientists unfamiliar with computer terminology would be able to communicate with each other and with the computer without translation of terms outside the computer. The desire for this close relationship developed when an analog computer data display and tape punch system was designed by the author for use in field experiments conducted by the Ocean Sound Propagation Section. Of course, data analysis is only one application of Act I and Encores.

The function of Cast of Characters is to print out the object program in decimal, with optional inclusion of symbols, constants, and operations. The purpose of Understudies is to punch out the object program in compatible decimal instead of hexadecimal in order that it may be moved, for instance, to allow the first eight tracks to be protected in installations having record-inhibit switches. The operations table was expanded to provide non-input typewriter output controls, breakpoint instructions, optional trace of selected locations, a variety of meanings for "-" program input code, and the three additional subroutine symbols "outpt," "addr," and "round."

The Ocean Sound Propagation Section has used a system in the past which includes a convenient alphanumeric output subroutine. Following the policy of improving communications among users, the author added a form of Herring's program (3).

The compatible hexadecimal "outpt" routine is intended to provide more rapid access to punched tape peripheral storage, i.e., to add a link via input and output devices without changing existing subroutines in order to render them compatible. The use of Act III compatible output subroutines was considered; however, necessary rearrangement of the subroutine packages seemed an unwarranted complication.

The "addr" subroutine causes decimal addresses input via the source program to be binarized under control of the object program and program input routine for manipulation in source program terminology.

The "round" operation masks off the 30th and 31st bits of words. It might find use in accurately unpacking fixed point numbers by dividing by a power of 10, rounding, then multiplying them by the same power of 10, at a q of 29.

No "z" delays are employed in these subroutines, permitting a nonstopping, high-speed-punch output mode of operation. Action, however, is slow enough for the typewriter output mode to be equally satisfactory.

ALTERATIONS TO THE COMPILER

In order that the symbol table be generated at 6000 and that nothing be recorded or erased in 5800-5963, the following alterations to the compiler are necessary:

Lo+0012	xh6000	Initialize symbol table
0103	u0104}	Bypass adding xz0200 to symbol Lo
0303	u0304}	
1038	xz6000	Symbol table Lo
1452	b1038	Inclusion in punchout

When using the altered compiler, dimensions are counted from location 6000 instead of 5800. For this reason the first "dim" statement should contain a number 128 greater than with the standard Act I.

In order that depression of the transfer control button will shift control to headings and, subsequently, to the special decimal memory printout routines rather than again to the compiler immediately following compiling, the 80xt4000 instruction in location 5444 is changed to 80xt3932.

THE OPERATIONS TABLE

The symbol table for Act I and Encores appears in Fig. 1. The resulting operations table compiled by Prelude to Act I and punched by Act I appears in Fig. 2.

No distinction was made between floating point and fixed point calling sequences for logarithm subroutines in order to provide more space in tracks 36 and 37. The left operand should be a number; but if floating log-in subroutine is employed, the number is not used.

A statement number in the source program should follow "-pir." When unprotected program input routine 10.4 or 10.4L is in memory, the object program stores the location of that statement in the address of the "t" instruction in 0007. Going to program input routine and inputting "-" and seven characters will transfer control to the statement with the seven characters in the accumulator.

"Trace" may be used with or without the transfer control button depressed during compiling. Its intended use is to select printout of intermediate results while computing with the transfer control button depressed.

"CAST OF CHARACTERS WITH REFERENCES ON REQUEST"

The Cast of Characters subroutine is located in 5800-5963 and is under control following the output of its heading program in 3932-3963. Block diagrams of the subroutine appear in Figs. 3 and 4.

After the compiler prints out program and statement locations there is a stop in 5443. If no decimal memory printout is desired at this point, the operator may transfer to another location, or he may turn on the punch, release the transfer control if on, and start to punch the object program at 0300, the symbol table at 6000, and the constants table at 6200.

If a decimal object program printout is desired, the transfer control button should be depressed and the computer started. The output still is to the typewriter, and all break-point switches are released. Subsequently, if the transfer control is left down, only the program will be printed, omitting the heading and references. If it is released, the heading and references will be included with the object program printout.

An example of the printout is shown in Fig. 5, followed by the coding of Cast of Characters in Fig. 6. The program in 3932-3963 employs the Cast of Characters alphanumeric subroutine to type out supplementary information, repairs the alphanumeric routine exit, then transfers to 5800 to initialize decimal printout at location 0300.

When the Encores compiler is in memory, a "-" program input routine instruction will transfer to 5802 with the hex address in the accumulator. This address will be the first location printed in decimal. If the address is the same as or greater than the last program location, only one printout cycle will follow. During normal Cast of Characters cycling, the last location and instruction "z0000" will be omitted, and the computer will stop, allowing the operator to choose what he would rather do next. He may either start the punch, use the program in memory, or alter the program first.

The Cast of Characters portion of the subroutine is optimized. The delay instruction following the carriage return is not necessary unless the latter instruction is a tabulation into a long, automatic carriage return, or a stop is desired before punching. When the Reference on Request portion of the subroutine is not to be used, "xz3000" is not needed as a constant and may be changed to "u1805;" otherwise, 5832 may be changed to "xs6331," Lf, instead of 3000. All temporary storage is within the subroutine. Therefore printout of track 63 contents is possible; however, an attempt to print out contents of tracks 58 or 59 damages the subroutine.

The transfer control may be depressed at any time to abort reference printouts if desired. When it is released, an instruction having an address of 6000-6163 is interpreted as one referring to a symbol. Following printout of the location and instruction, the symbol is typed. If the word found in 6000-6163 happens to contain a non-input (e.g., "0800") or jam code (e.g., "4800"), that character is bypassed, preventing uncontrolled action, damage, or stoppage. When the computer encounters an instruction having an address of 6200-6239, the word in that location is typed as a five-digit decimal constant at 29. Negative numbers or numbers greater than 99999 will not stop the computer but may contain nondecimal characters. Instruction addresses 3000-5963 are first treated as subroutine bridge locations. If the command is not "u," as it normally would be following an "r0310" instruction, a "u" - meaning unidentified - is typed. If the command is "u," the table of transfer addresses at q of 30 in track 37 is searched. If the address is not found, a "u" is printed. When it is found, the symbol in track 36 having the same sector is typed as the operation symbol. For examples, the following instructions would cause the resulting typeouts:

p3200 u	(stop code, not "u" instruction)
z3200 u	(breakpoint, not "u")
r0305	(address not 3000 or more)
u5000 u	(not transfer address)
b6001 temp.	(symbol)
r0310	(address not 3000 or more)
u4900 asin	(operation symbol)
s6208 04963	(constant)

For future reference, the typewriter punch may be turned on before transferring to Cast of Characters, thereby including the decimal interpretation on the object program tape. Since the program is incapable of punching a conditional stop and the first input word is an eight-character "v" code, the printout will be ignored by the input routines.

"UNDERSTUDIES"

An optional decimal memory printout program may be loaded in 5800-5963 with its headings program in 3932-3963. Understudies (Fig. 7) is similar to Cast of Characters in that it prints locations and instructions in decimal and includes printout of symbols referred to by the object program; however, here the similarity ends.

The Understudies subroutine is headed by loading limits and instructions. The object program, properly modified, can be moved up to the initial region location designated by "dim" statements. The maximum start fill instruction therefore is

region Li - [program Lf - (program Li - 1)].

The Understudies subroutine does not modify any addresses, so the modifier must be three tracks smaller than the start fill instruction. If subroutines are used with the object program, all return addresses must be changed to program Lo + 0010, the N temporary storage addresses must be changed to program Lo + 0012, and trace subroutine must return to Lo + 0005. Subroutines may be altered by loading a tape of start fill plus instruction sets after the subroutines to be used are loaded and /Lo of program is stored by program input routine.

The Understudies program does not print out constants, operations, or "u." Following carriage return and printout of location, each object program instruction is tested. If the command bits are all zero ("Z"), then ",0000001'" and the word in hex is printed with a conditional stop. The check sum printout routine of the compiler is employed to output 1 at 2, 1 at 30, breakpoint delay, and stop instructions, since all happen to be positive and contain "0" bits in their command portions. "p" instructions and instructions having addresses greater than object program Lf are typed with (a) leading spaces to clear the remainder of the location out of the accumulator on reloading, (b) an x to prevent modifying, and (c) conditional stop. An instruction having an address 6000-6163 is followed by the symbol in that address. All other instructions are assumed to be modified and are preceded by spaces and no "x." After typing the final instruction, "xz0000," control is immediately transferred to the compiler, which punches the symbol table and constants table on the typewriter in hexadecimal, each with a check sum.

The object program is unchanged in memory, so it may be used immediately or, if desired, punched out in hex at 0300 by transferring to location 5445. Understudies uses the address printout routine in the compiler to type locations and one space, but uses its own address printout routine to type addresses and conditional stop. Hence, interrupting the program cannot alter Act I.

If the object program is to be moved in order that it can be used with protected program input routine, it should be remembered that "outpt," "addr," and "-code" operations then will not work. The first requires subroutines which use constants in unprotected program input routine; the second must "r-u" to program input routine binarization subroutine; and the third must alter an inhibited instruction in order to function.

"OUTPT"- "ADDR"- "ROUND" SUBROUTINE PACKAGES

Since the compiler directly stores symbols of up to five characters at a q of 30 by n-multiplying the six-bit input word, it is convenient to store alphanumeric symbols using

the source program. Furthermore, non-input typewriter codes and delays may be assembled by the compiler if appropriate symbols appear in the operations table. The author reviewed a POOL* program by Mr. Herring for alphanumeric output and condensed it for this subroutine package.

Symbol "outpt" for alphanumeric words differs from the alphanumeric output in the Cast of Characters subroutine by omitting protection against non-input codes (Fig. 8). Starting at the left, six bits at a time are shifted into the track portion of a p() instruction. p0000 instructions are bypassed, being interpreted as missing character codes of stored symbols having less than five characters. An example of source program and result is:

```
outpt'temp.' '
2'print'temp.' '          temp. 662.15 F
ucase'outpt' F' lcase' cr' '
```

If an alphanumeric symbol is not to be replaced by a number in memory before reloading the object program, it should not be identical to the symbol used to represent the number storage location in 6000-6163. A period or space, for instance, might differentiate one from the other.

The "addr" and "round" subroutines help delay the exit from either "outpt" subroutine so that either high-speed punch or typewriter output can be used. The "addr" subroutine is simply a transfer to the binarizing section of 10.4 or 10.4L program input routine and return. Constants and fixed point numbers in the object program are at a q of 29, whereas program input routine codes are at a q of 31. Therefore transfer is made from the "addr" subroutine location to a point beyond the order to n-multiply by 1 at 29. The "round" subroutine contains one extract instruction to mask off bits 30 and 31. Since its transfer address is listed in the operations table, Cast of Characters can include printout of the operation as it can with "addr" and "outpt" operations.

The "addr" and "round" subroutines follow the "outpt" subroutine at the same transfer addresses when combined with the compatible hex "outpt" subroutine (Fig. 9). Four bits at a time are shifted into a p() instruction, the hex character identification bit is added, and eight characters - forcing previous contents out of the accumulator - and a conditional stop are printed.

PACKAGING

Act I and Encores have been packaged along with debugging programs, bootstrap with check sum, and supplementary programs to form the following twelve-tape library in the possession of the writer:

- I Prelude, Operations Table, Compiler, Cast of Characters
- II Bootstrap II with Σ , Program Input Routine 10.4L, Index Instructions and Constants, "-" code to Cast of Characters
- III Symbol Table
- IV Alphanumeric/Hexadecimal Subroutines, Bootstrap I
- V Loader and Linkages, Trace-Float-Unfloat, Floating Decimal, Fixed Point Input-Output Subroutines
- VI Floating Point Subroutines
- VII Fixed Point Subroutines
- VIII Program Input Routine 24.2
- IX Decimal Memory Printout and Hex Punch
- X U. of Mich. Trace
- XI React
- XII Understudies

* An LGP-30 user's organization.

React is a one-track assembly (Fig. 10) for use without the compiler to help write efficient programs employing Act I subroutines.

SUMMARY

Changes have been made to improve readability and increase the versatility of Act I. No difficulties have been encountered other than remembering to increase the first "dim" statement number by 128 and to refrain from using subroutines requiring 10.4 or 10.4L PIR when protected PIR is used.

REFERENCES

1. Morrisseau, K. C., "Programming the LGP-30 Computer Using Act I," (Code 447) U. S. Navy Bureau of Ships, Washington, D. C.
2. Royal McBee Corp., "LGP-30 Act I Compiler," Bulletin No. S-520-R1, Port Chester, N. Y.
3. Herring, R. L., "Alphanumeric Output for Act I," POOL, The LGP-30 Users Organization, Feb. 7, 1961, Pasadena, Calif.

```

;0'blo'hro'hio''
+1'blo'sro'hio''
-1'blo'sro'hio''
x2'blo'nro'mca'0307'hio''
/2'bca'0308'dro'mlo'hio''
when'0''
less'3'blo'sro''
grt'3'bro'slo''
equal'3'blo'sro'tja'3'aca'314''
trn'0'tro''
abs'3'bro'tja'2'uja'3'cca'306'sca'306'hio''
read'3'rca'0310'uta'5700'hro''
print'3'blo'hca'0312'bro'rca'0310'uta'5800''
stop'0'zca'0''
iter'sub'
f+'1'blo'hca'0312'bro'rca'0310'uta'5200'hio''
f-'1'blo'hca'0312'bro'rca'0310'uta'5300'hio''
fx'2'blo'hca'0312'bro'rca'0310'uta'5400'hio''
f/2'blo'hca'0312'bro'rca'0310'uta'5328'hio''
finp'3'rca'0310'uta'5502'hro''
fprt'3'bro'rca'0310'uta'5700''
flo'3'blo'hca'0312'bro'rca'0310'uta'5137'hio''
unflo'3'blo'hca'0312'bro'rca'0310'uta'5100'hio''
cr'3'pca'1600'zca'0000''
tab'3'pca'2400'zca'0000''
ret'0'rro''
use'0'uro''
bring'0'bro''
add'0'aro''
subtr'0'sro''
mult'0'mro''
rmult'0'nro''
div'0'dro''
extrt'0'ero''
hold'0'hro''
clear'0'cro''
stadd'0'yro''
rdhex'3'cca'306'pca'0090'ica'0000'hro''
sin'3'bro'rca'0310'uta'4908'hio''
cos'3'bro'rca'0310'uta'4915'hio''
atan'3'bro'rca'0310'uta'4939'hio''
sqrt'3'bro'rca'0310'uta'4920'hio''
asin'3'bro'rca'0310'uta'4945'hio''
acos'3'bro'rca'0310'uta'4953'hio''
exp'3'bro'rca'0310'uta'4926'hio''
log'3'bro'rca'0310'uta'4933'hio''

ln'3'blo'hca'0312'bro'rca'0310'uta'4954'hio''
log'3'blo'hca'0312'bro'rca'0310'uta'4958'hio''
outpt'3'bro'rca'0310'uta'6240''
addr'3'bro'rca'0310'uta'6256'hio''
round'3'bro'rca'0310'uta'6258'hio''
-pir'0'bro'yca'0007''
trace'3'bro'rca'0305'uca'5000''
ucase'3'pca'0800'zca'0000''
lcase'3'pca'0400'zca'0000''
color'3'pca'1200'zca'0000''
bksp'3'pca'2000'zca'0000''
cdstp'3'pca'3200'zca'0000''
bp4'3'zca'0400''
bp8'3'zca'0800''
bp16'3'zca'1600''
bp32'3'zca'3200''

```

Fig. 1 - "Encores" symbol table

```

v0402400'
0001q'00016'0000q'0004q'00026'01w632g2'0034gqvr'5j6kr'
4gk53j8j'5f6g2'722vr'006jgjr'426r2jkr'01qkg1j2'008kg29r'7g4u'
01516'0150q'0154q'01526'01522j2'015426kr'54346'52jkk346'
01r9r'5g48r'1g2kr'53qjr'0r6r2jkj'72rrr'7g48g69r'00qk235r'
32qk235r'2f0gq'4g3kr6kr'018j632r'6r34g19r'7g6v2rrr'1rrq32jq'7f8g2'
6g1vr'01jkgjg2'01qv46kr'01jvr8g2'01jag1vr'012jg3j2'0114g3j2'00332'
0k1kj'474kg0kr'01jrr9r'1g1k2jrr'003j289r'5r6v3rjr'53rv3qjr'0krv3qjr'
6g18k19r'002qkqj2'6rrvg6j2'0g0r4'0g0j4'002j2334'002j2714'w0000000'
2rrq24j92'v03v2500'
03200000'07206000'0720j000'0g312000'0g31r000'01w1f000'0w10j000'0w122000'
0w326000'0202q000'0q530000'0q23kk80'0w443k00'0204j000'008kg29r'7g48r'
0754vr00'0754vr00'0g54w000'0g54vr8'0q23kg84'0q247j80'0v54w9jr'0v54w980'
0q15r000'0q15q000'02062000'02064000'02022000'02008000'0200q000'02066000'
02014000'0201j000'02068000'02002000'0206r000'0206j000'0q36q000'0q353890'
0q35389q'0q3538jq'0q3538r8'0q3538kr'0q3538v6'0q3538g4'0q3538j2'0v54w8qj'
0v54w8w4'0q247w50'0q353v70'0q353v74'021'6000'0q27r000'0q180000'0q184000'
0q188000'0q18j000'0q190000'0q094000'0q096000'0q098000'0q09r000'33vr7k3r'
v04q2600'
10000'010j0000'018j0000'10000'010q0000'018j0000'10000'010w0000'
018j0000'10000'01060000'01q7031j'018j0000'01q10320'01050000'70000'
018j0000'01010000'w0000'10000'010w0000'01jg000j'01qv0338'010g0000'
01010000'01jg0008'01jr000j'01qk0318'01qv0318'018j0000'01q30328'800r0000'
010j0000'10000'01qj0330'01010000'01q30328'800r0000'01q00000'10000'
01qj0330'01010000'01q30328'800r0000'018j0000'01q81000'01q00000'01q81800'
01q00000'01030000'010r0000'01070000'01090000'010k0000'01020000'01qk0318'
01q80000'01q40000'010j0000'01010000'01q2001j'01010000'01q30314'01qf3200'
01q80800'01q00000'01q80400'01q00000'01q80j00'01q00000'01q81400'01q00000'
01q82000'01q00000'01q00400'01q00800'01q01000'01q02000'285j07q6'

```

Fig. 2 - "Encores" operation table

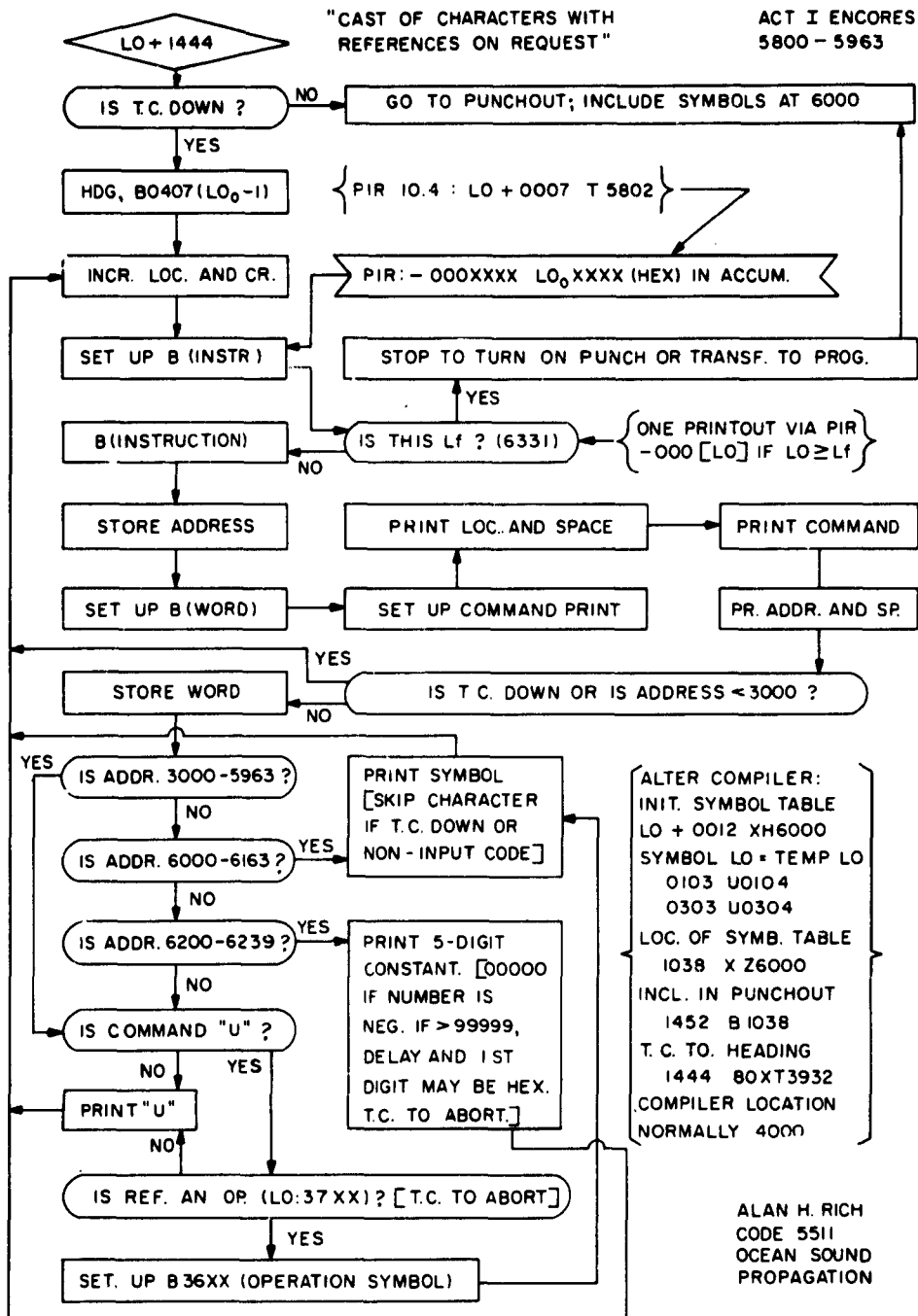


Fig. 3 - "Cast of Characters" block diagram

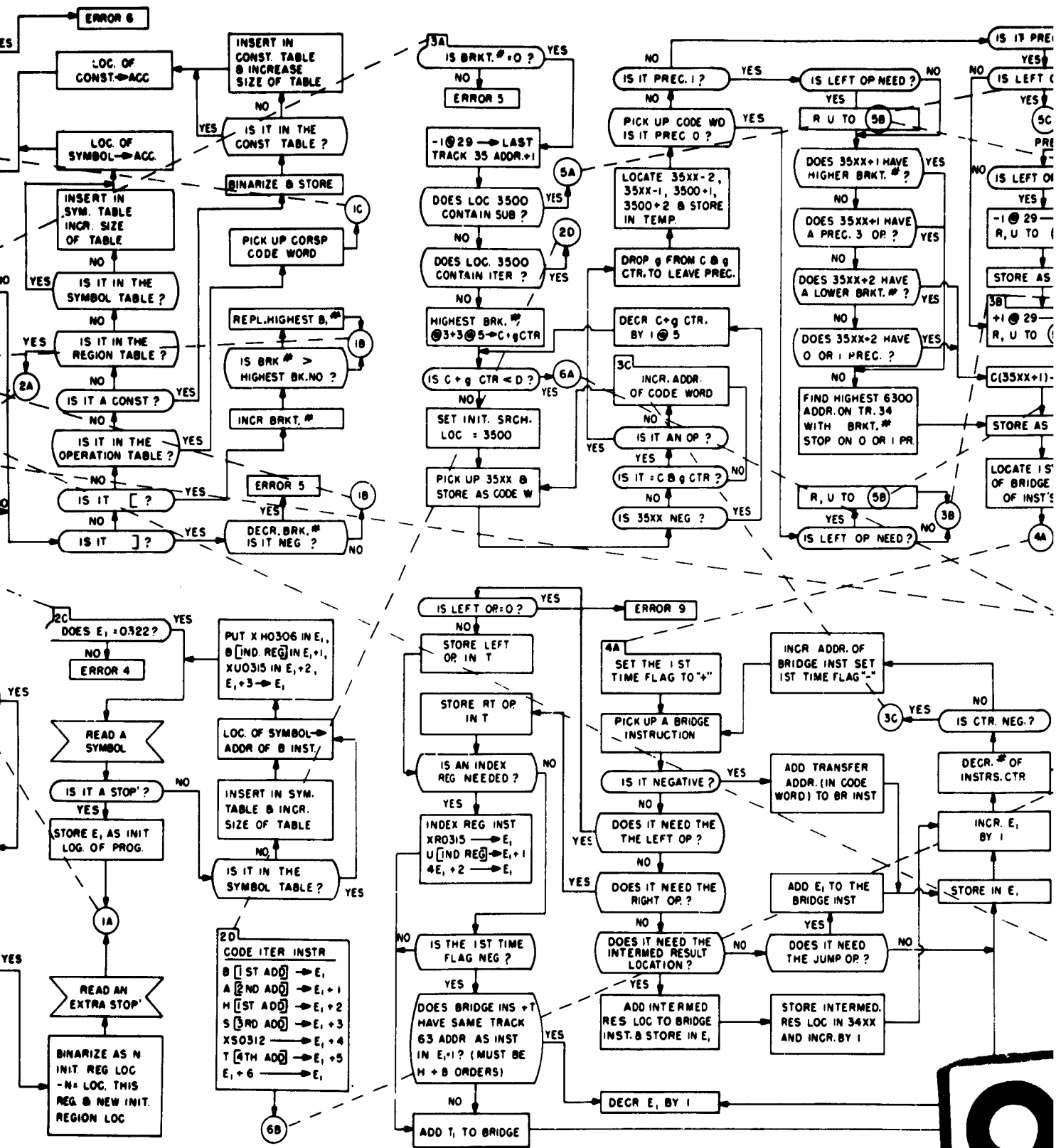
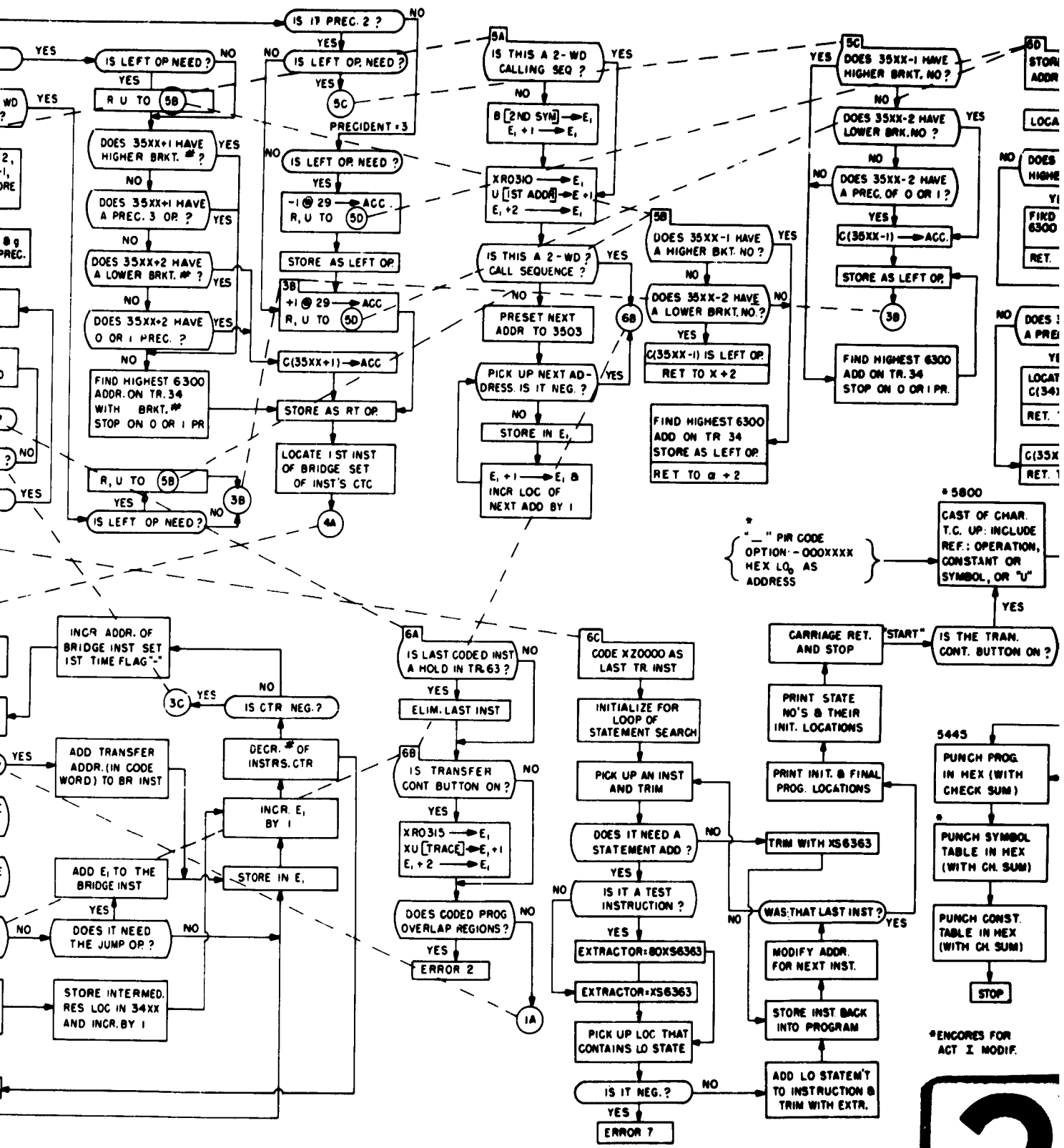
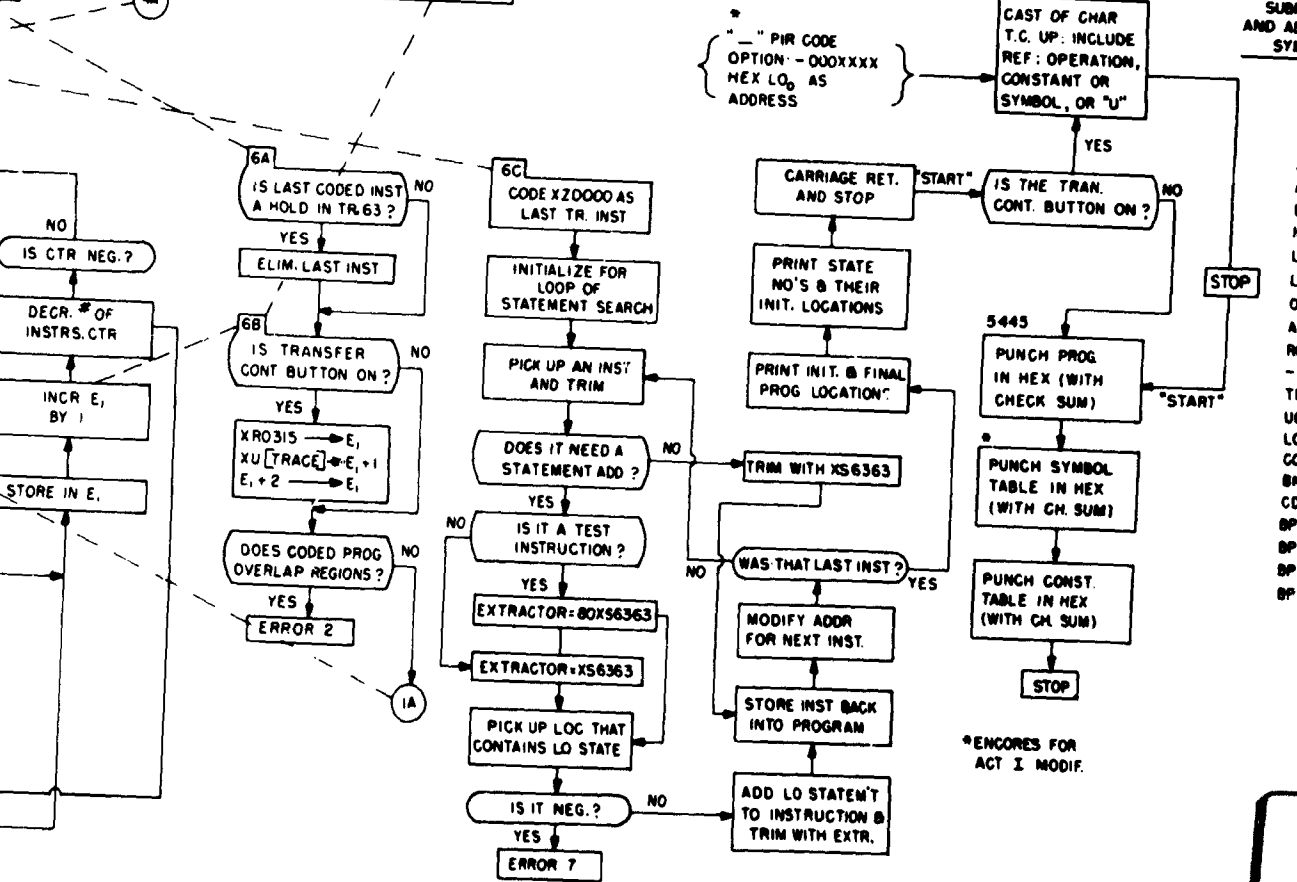
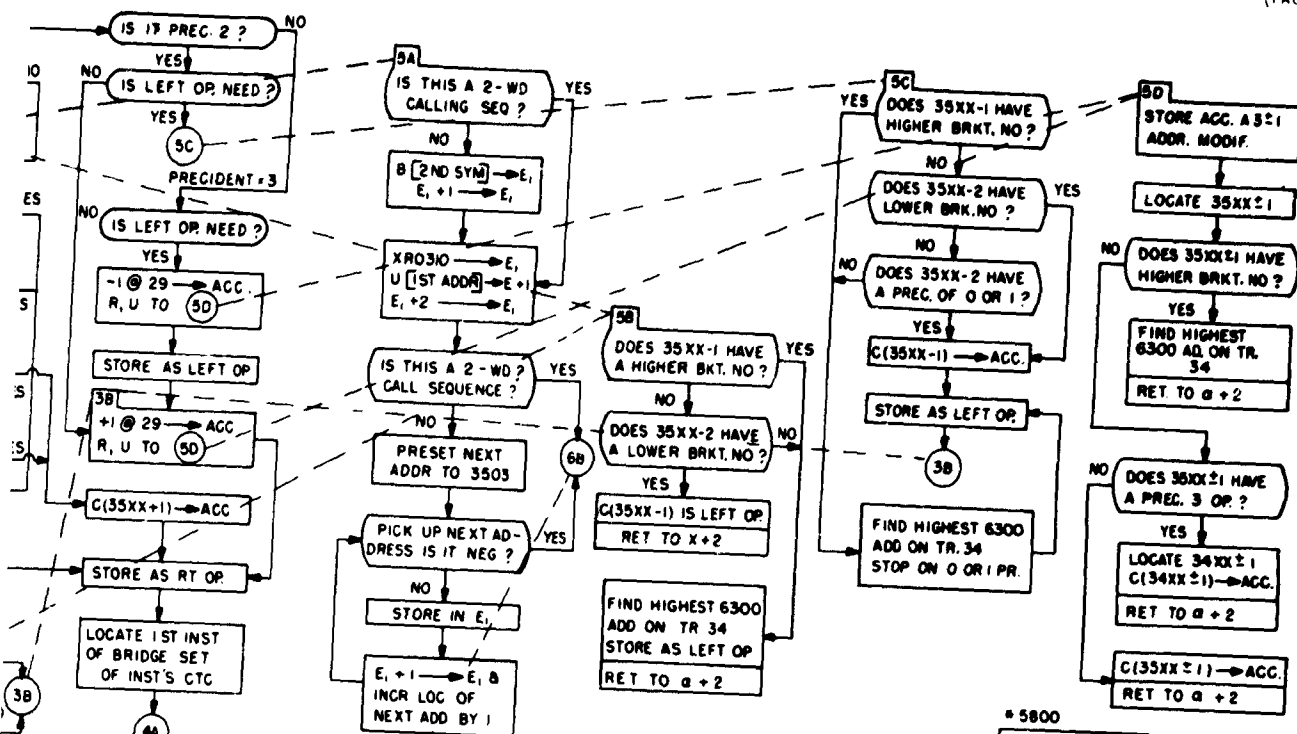


Fig. 4 - Altered "Act I" block diagram



Act I" block diagram



* " PIR CODE OPTION - 000XXXX HEX LO AS ADDRESS

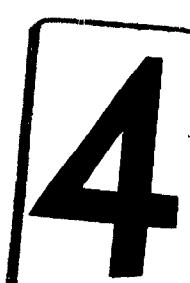
* 5800

CAST OF CHAR T.C. UP: INCLUDE REF: OPERATION, CONSTANT OR SYMBOL, OR "U"

SUBROUTINE AND ADDITIONAL SYMBOLS

- SIN
- COS
- ATAN
- SORT
- ASIN
- ACOS
- EEXP
- IOEXP
- LN
- LOG
- OUTPY
- ADDR
- ROUND
- PIR
- TRACE
- UCASE
- COLOR
- BKSP
- CDSTP
- BP 4
- BP 8
- BP 16
- BP 32

*ENCORES FOR ACT I MODIF.



.0004000	0324 r0310
dimcomp800	0325 u5900 read
slcr	0326 h6000 a
reada	0327 b6200 00004
4flob;fa	0328 h0312
tab	0329 b6000 a
readb	0330 r0310
3flob;fb	0331 u5137 flo
tab	0332 h6001 fa
fprt[sqrt[fa+fb]]	0333 p2400
uses1	0334 z0000
	0335 r0310
i 0322 r 0356	0336 u5900 read
	0337 h6002 b
	0338 b6201 00003
a01 0322	0339 h0312
symb 6000+	0340 b6002 b
cons 6200+	0341 r0310
amp w ref	0342 u5137 flo
if tran on	0343 h6003 fb
0307 l at2	0344 p2400
0314 lat30	0345 z0000
	0346 b6001 fa
0300 b0315	0347 h0312
0301 u0318	0348 b6003 fb
0302 z0300	0349 r0310
0303 u0305	0350 u5200 f+
0304 z0001	0351 r0310
0305 u0305	0352 u4920 sqrt
0306 z0000	0353 r0310
0307 z0000	0354 u5700 fprt
0308 z0001	0355 u0322
0309 z0000	v0390300'
0310 u0644	1033j'r0348'00300'r0314'00004'r0314''20000000'
0311 z0000	00004''r06g0''''00002'q033j'
0312 z0000	j0308'r0300'q0310'20314'10318'r0308'81000''
0313 z0000	30328'r3g00'j3j00'13q00'j0330'13j00'30328'r3394'
0314 z0000	j3j04'81800''30328'r3g00'j3j08'13q04'j0330'
0315 a0315	13j08'30328'r3394'j3j0j'81800''13j04'j0330'
0316 h0302	13j0j'30328'r3400'30328'r3150'30328'r3900'r0358'
0317 u0300	'2143669r'v0053j00'
0318 a0304	00072'01572'0000r'0150r'www'wwwj'00002fw8'vc033q00'
0319 y0305	00010'0000j'wwwwwwj'0000001j'v0183qr0'
0320 b0306	01rj6qj2'wwwwwwj'q1740q30'70994'r0728'r116'q3700q30'q2320q30'
0321 u0302	0000j'q3kg0q30'12g44'709w'02w80'63kj8'r1r84'13kk0'
0322 pl600	r1r78'11814'26k7885.'26k7885r'26k7885r'26k7885r'q3q30'r3390'
0323 z0300	28548r46'

Fig. 5 - "Cast of Characters" printout


```

,0004000          0323 ,0000001' 00000000'
dimcomp800       0324 r0310'
slcr             0325 xu5900'
reada           0326 xh6000'a
4flow;fa        0327 xb'200'
tab             0328 h0312'
readb           0329 xb6000'a
3flob;fb        0330 r0310'
tab            0331 xu5137'
fprt[sqrt[fa'+fb]] 0332 xh6001'.fa
uses1           0333 xp2400'
i 0322 f 0350.  0334 ,0000001' 00000000'
0335 r0310'
0336 xu5900'
0337 xh6002'b
PUNCH ON:       0338 xb6201'
;max4:59 ,0300 less 0339 h0312'
0300 b0315'     0340 xb'002'b
0301 u0318'     0341 r0310'
0302 ,0000001' 00000300' 0342 xu5137'
0303 u0305'     0343 xh6003'.fb
0304 ,0000001' 00000004' 0344 xp2400'
0305 u0305'     0345 ,0000001' 00000000'
0306 ,0000001' 00000000' 0346 xh6001'.fa
0307 ,0000001' 20000000' 0347 h0312'
0308 ,0000001' 00000004' 0348 xb'.003'.fb
0309 ,0000001' 00000000' 0349 r0310'
0310 xu0644'    0350 xu7200'
0311 ,0000001' 00000000' 0351 r0310'
0312 ,0000001' 00000000' 0352 xu4920'
0313 ,0000001' 00000000' 0353 r0310'
0314 ,0000001' 00000002' 0354 xu5700'
0315 a0315'    0355 u0322',0000001'v0053j00'
0316 h0302'    00072'01572'0000f'0150f'wwwwwwj'00002fv8'v0033q00'
0317 u0300'    00010'0000j'wwwwwwj'0000001j'v0183qr0'
0318 a0304'    01fj6qj2'wwwwwwj'q1740q30'70994'f0728'f1168'q3700q30'q2320q30'
0319 y0305'    0000j'q3kg0q30'12g44'709w0'02v80'63kj8'f1f84'13kk0'
0320 b0306'    f1f78'11814'26k7885f'26k7885f'26k7885f'26k7885f'q3q30'f3390'
0321 u0302'    26'hw2w4'
0322 xp1600'
    
```

Fig. 7 - "Understudies" printout

```

;0006240'
/0006200' 40 xc0306' word (OUTPT)
41 b0063' -5 at 29
42 xc0312' counter
43 xb0306' word
44 m0061' 1 at 17
45 e0060' xz6300
46 y0049' p[ ]
47 xa0029' 1 at 29
48 t0050' no character?
49 pz000' alpha character
50 xb0306' word
51 xn0135' 1 at 25
52 xc0306' word
53 xb0312' counter
54 xa0029' 1 at 29
55 t0042' 5 characters:
56 xr0063' delay (ADDR)
57 xu0051' binarize
58 e0062' ROUND at 29
59 xu0310' exit
60 xz6300'
,0000003' 61 4000'
62 wwwwwwj'
63 wwwwwwqj'

;0000135' xz0016' patch
.0000000'

```

Fig. 8 - Alphanumeric
"outpt"- "addr"- "round"
subroutine package

Fig. 9 - Compatible hex
"outpt"- "addr"- "round"
subroutine package

```

;0006240'
/0006200' 40 xc0306' word (OUTPT)
41 b0063' -8 at 29
42 xc0312' counter
43 xb0306' word
44 xm0156' 1 at 18
45 e0060' xz6000
46 a0061' xz0200
47 y0048' p[ ]
48 pz000' hex character
49 xb0306' word
50 xn0135' 1 at 27
51 xc0306' word
52 xb0312' counter
53 xa0029' 1 at 29
54 t0042' 8 characters?
55 xp3234' cond. stop
56 xr0063' delay ADDR
57 xu0051' binarize
58 e0062' ROUND at 29
59 xu0310' exit
60 xz6000'
61 xz0200'
,0000002' 62 wwwwwwj'
63 wwwwwwq0'

;0000135' xz0004' patch
.0000000'

```