IDENTIFICATION: TRACE I

AUTHOR: P. Jarvie, PBCC

ACCEPTED: February 7, 1961

PURPOSE: To interpret and execute instructions of an object program (the program to be traced) and type out specified instructions of the object program, along with their locations and the register contents.

This routine will usually be employed as a debugging aid in checking out new routines.

RESTRICTIONS:

1. The Octal Utility Package (PBCC Cat. No. 0001A) must be in memory.

2. IAM, MLX, MCL, BSO, BSI will not be correctly interpreted when the sector number of the instruction is the same as either the location or the next location.

3. The undefined commands (OP codes 27, 54, 74, 76) are not recognized as errors and will be interpreted incorrectly.

4. A TRU command (37) without a sequence tag is interpreted as a NOP (24).

5. If the object program causes the Flexowriter to go into upper case, all subsequent Trace print-out will stay in upper case until a lower case is given by the object program.

6. The Breakpoint switch is used (see paragraph 4 of "USE").

7. Whenever a Trace print-out occurs, the contents of the last two sectors of line 05 are lost.

RESTRICTIONS:

8. Instruction sequences based on critical timing considerations (such as READ sequences) will not necessarily follow the same pattern when run under Trace control as when run under machine control. I.E.: it fouls up I/O.

SPACE REQUIRED:

The Trace routine occupies all but the last two sectors of a command line (254 sectors). The Octal Utility Package is assumed to be in line 01, and the contents of the last two sectors of line 05 are lost whenever a Trace print-out occurs. No other memory, including the short line (00) and the Index Register, is disturbed.

TIMING:

The amount of time required to trace a program is approximately

$$.15n + 7n_t \text{ seconds}$$

Where

n = Total number of instructions (implicit) in the object program.

n_t = Number of instructions "tagged" (see USE) for print-out.

If the Breakpoint switch (see USE) is depressed when tracing, the time can be reduced to approximately

$$.15n + 2.7n_t \text{ seconds}$$

USE:

1. Trace is loaded into a selected command line, with the line choice determining which one of six Trace tapes is to be used (one tape for each of lines 02 thru 07) ; i.e., Trace is <u>not</u> relocatable and a tape must be prepared for each command line. The annotated listing (Appendix E) indicates by an XX in the line field where the selected command line number is to be inserted.

2. Once Trace is loaded, there are three items that

USE (cont.): the routine requires in order to interpret an object program:

a) α, the location of the first instruction of the object program.
b) C_1, an extract (mask) parameter.
c) C_2, a compare parameter.

Item (a) must always be input. Items (b) and (c) must be input if selective print-out is desired.

A print-out occurs whenever an instruction, I, is such that

$$\overline{C_1}\, I = C_2$$

In this case, we say the instruction is "tagged" for print-out.

α is input to Trace, by means of the Octal Utility Package, in the form LDB α (no sequence or index tag) and is stored in sector 000 of Trace.

C_1 and C_2 are already in Trace so that every instruction is tagged; i.e.,

$$C_1 = -7777777$$
$$C_2 = +0000000$$

If C_1 and C_2 are to be input for a selective print-out, C_1 is stored in sector 141 of Trace, while C_2 is stored in sector 172 of Trace.

USE (cont.):

With the object program in memory and Trace in a command line, initialized as indicated above, tracing can begin by transferring (via the Octal Utility Package) to sector 000 of Trace.

3. Two examples of the use of Trace are as follows:

a) If the first instruction in the object program is in line 05, sector 102, then

LDB $\alpha \equiv$ 102 0605; $\longrightarrow$ sector 000 of Trace.

If it is desired to print only at all unconditional transfers, then,

C_1 = 377 0077I $\longrightarrow$ sector 141 of Trace.
C_2 = 000S3700; $\longrightarrow$ sector 172 of Trace.

b) To use T_2 (the unused bit in minimal memory machines) as a Trace tag, then

C_1 = 377S7737I $\equiv$ -7777775
C_2 = 000 0040; $\equiv$ +0000002

4. Two print modes are possible for tagged instructions, depending on the position of the Breakpoint switch.

Breakpoint normal (up) will provide a full print-out as follows:

C/R Location Instruction (A) (B) (C) (Index)

a) Location print-out is in command format of the form S LDB L, where only the sector and line numbers have any meaning.

USE (cont.):

b) Instruction print-out is in command format.
c) A, B, C, register print-out is in data format, in true word image form, and shows the status of the registers after execution of the instruction.
d) Index register print-out is in command format, where only the line number portion has any meaning, and shows the status of the register after execution of the instruction.

Breakpoint on (depressed) will omit the register print-out; i.e., only the following will be printed:

C/R   Location   Instruction

5. To stop Trace and gain Octal Utility Package keyboard control, depress the Enable switch and the I key, then insert the special 33-frame tape into the reader and hit the F key. This tape restores sectors 046 and 140 of line 01. When the tape has been read in, the Octal Utility Package will have resumed its normal functions.

6. To restart Trace, the procedure beginning at step 2 of USE should be followed. If it is desired to restart at the point where Trace was stopped or, in general, if it is desired to start Trace with specified register settings (except for the Index register), the following should be input via the Octal Utility Package:

A Register Setting ⟶ Sector 320 of Trace.
B Register Setting ⟶ Sector 317 of Trace.
C Register Setting ⟶ Sector 322 of Trace.
Overflow Flag ⟶ Sector 314 of Trace.
(Where 0 implies overflow,
and $\neq$ 0 implies no overflow.)

METHOD:

An instruction is picked up and examined to determine the first print-out. General linkages are set for the several instruction classes, and the operation code is then filtered to determine the execution mode and possible next instruction (s). Registers are then restored, as well as the overflow condition, and execution takes place in either an actual or dummy location, with the instruction addresses and tags suitably modified. Post execution information is saved, as well as next instruction control for branch commands. The Breakpoint switch and a flag are examined to determine the second print-out. Finally, next-instruction logic, based on the sequence tag, modifies the initial Trace pick-up command before repeating the cycle.

The method by which the execution mode and the next instruction are determined, is based on the tables in Appendix C.

The Trace print-out is based on the presence of the Octal Utility Package in line 01. Two instructions of the utility routine are modified by Trace in order to provide a return link to Trace. Actual printing is done by the Octal Utility Package under control of Trace. A special tape is provided to restore the two modified instructions when terminating Trace.

APPENDIX A

TRACE I
SUMMARY OF USE

1. Load Octal Utility Package and Trace.

2. Input by means of Octal Utility Package:

   a) LDB $a \longrightarrow$ sector 000 of Trace. → ααα Δ06XX;
      Where a = location of 1st instruction in program to be traced.

   b) If selective Trace print-out is desired, also input

      $C_1 \longrightarrow$ sector 141 of Trace.
      $C_2 \longrightarrow$ sector 172 of Trace.

3. Transfer to sector 000 of Trace.

4. When a tagged instruction comes up, Breakpoint switch normal (up) will cause full print-out; whereas Breakpoint switch depressed will cause a partial print-out.

5. To stop Trace and use the Octal Utility Package,

   a) Depress the Enable switch and hit the I key.
   b) Insert special tape into the reader and hit the F key.

6. To resume tracing,

   a) Preset registers (if desired).

      A setting $\longrightarrow$ sector 320 of Trace.
      B setting $\longrightarrow$ sector 317 of Trace.
      C setting $\longrightarrow$ sector 322 of Trace.
      Overflow Flag $\longrightarrow$ sector 314 of Trace.

      Flag $=$ 0 for overflow.
      Flag $\neq$ 0 for no overflow.

   b) Follow procedure beginning at Step 2 above.

APPENDIX B

TRACE I
SAMPLE PROBLEM

Insert B-1 is a sample program which, when traced, illustrates the input-output aspects of Trace. The sample program was stored in line 04, while Trace was in line 03.

Insert B-2 is a portion of a full trace of the program; i.e., all instructions were tagged.

Insert B-3 illustrates a selective trace; it is desired to output only at those instructions where the line number is 11.

For this requirement:

C_1 = 377S7700I ≡ -7777603
C_2 = 000 0011; ≡ +0000041

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM SAMPLE OBJECT PROGRAM — PAGE 2 OF 4

PROGRAMMER P. JARVIE — DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
10004	101S0504;	LDA	SET INDEX TO LINE 04
10204	103S1137;	STA	
10404	105 7500I	TOF	TURN OVERFLOW OFF
10504	107 0500I	LDA	Ao ⟶ A
10604	107S4300;	CLB	0 ⟶ B
11004	111S1611I	DPA	Ao + Δ
11304	115 7500I	TOF	TO END IF OVERFLOW
11404	110S3700I	TRU	ADD ANOTHER Δ
11504	110S0011I	HLT	END: RESTART WILL CONTINUE
10104	+0000010	CONST	4 FOR INDEX REGISTER
10704	+7777775	CONST	TO PRESET A (Ao)
11104	-1111111	CONST	TO INCREMENT (Δ)
11204	+0000000	CONST	

TRACE I -- INSERT B-1
SAMPLE OBJECT PROGRAM

```
00003$100 0604;
00003.
100 0604;101S0504;+0000010+0000000+0000000111S5103;
102 0604;103S1137;+0000010+0000000+0000000111S5104;
104 0604;105 7500I+0000010+0000000+0000000111S5104;
105 0604;107 0500I+7777775+0000000+0000000111S5104;
106 0604;107S4300;+7777775+0000000+0000000111S5104;
110 0604;111S1611I+7777775-1111111+0000000111S5104;
113 0604;115 7500I+7777775-1111111+0000000111S5104;
114 0604;110S3700I+7777775-1111111+0000000111S5104;
110 0604;111S1611I+7777776+2222222+0000000111S5104;
113 0604;115 7500I+7777776+2222222+0000000111S5104;
114 0604;110S3700I+7777776+2222222+0000000111S5104;
110 0604;111S1611I+7777776-3333333+0000000111S5104;
113 0604;115 7500I+7777776-3333333+0000000111S5104;
114 0604;110S3700I+7777776-3333333+0000000111S5104;
110 0604;111S1611I+7777777+4444444+0000000111S5104;
113 0604;115 7500I+7777777+4444444+0000000111S5104;
114 0604;110S3700I+7777777+4444444+0000000111S5104;
110 0604;111S1611I+7777777-5555555+0000000111S5104;
113 0604;115 7500I+7777777-5555555+0000000111S5104;
114 0604;110S3700I+7777777-5555555+0000000111S5104;
110 0604;111S1611I-0000000+6666666+0000000111S5104;
113 0604;115 7500I-0000000+6666666+0000000111S5104;
115 0604;110S0011I-0000000+6666666+0000000111S5104;
110 0604;111S1611I-0000000-7777777+0000000111S5104;
113 0604;115 7500I-0000000-7777777+0000000111S5104;
114 0604;110S3700I-0000000-7777777+0000000111S5104;
110 0604;111S1611I-0000001-1111110+0000000111S5104;
113 0604;115 7500I-0000001-1111110+0000000111S5104;
114 0604;110S3700I
110 0604;111S1611I
113 0604;115 7500I
114 0604;110S3700I
110 0604;111S1611I
113 0604;115 7500I          } BP DEPRESSED HERE
114 0604;110S3700I
110 0604;111S1611I
113 0604;115 7500I
114 0604;110S3700I
110 0604;111S1611I-0000003-5555554+0000000111S5104;
113 0604;115 7500I-0000003-5555554+0000000111S5104;
114 0604;110S3700I-0000003-5555554+0000000111S51 4;
110 0604;111S1611I-0000004+6666665+0000000111S5104;
113 0604;115 7500I-0000004+6666665+0000000111S5104;
114 0604;110S3700I-0000004+666IF <---- Terminated
```

Insert B-2 FULL TRACE PRINT-OUT

```
00003$100 0604;
14103$377S77001
17203$000 0011;
00003.
110 0604;111S1611I+7777775-1111111+0000000111S5104;
110 0604;111S1611I+7777776+2222222+0000000111S5104;
110 0604;111S1611I+7777776-3333333+0000000111S5104;
110 0604;111S1611I+7777777+4444444+0000000111S5104;
110 0604;111S1611I+7777777-5555555+0000000111S5104;
110 0604;111S1611I-0000000+6666666+0000000111S5104;
115 0604;110S0011I-0000000+6666666+0000000111S5104;
110 0604;111S1611I-0000000-7777777+0000000111S5104;
110 0604;111S1611I-0000001-1111110+0000000111S5104;
110 0604;111S1611I-0000002+2222221+0000000111S5104;
110 0604;111S1611I-0000002-3333332+0000000111S5104;
110 0604;111S1611I-0000003+4444443+0000000111S5104;
110 0604;111S1611I-0000003-5555554+0000000111S5104;
110 0604;111S1611I-0000004+6666665+0000000111S5104;
110 0604;111S1611I-0000004-7777776+0000000111S5104;
110 0604;111S1611I-0000005-1111107+0000000111S5104;
110 0604;111S1611I-0000006+2222220+0000000111S5104;
110 0604;111S1611I-0000006-3333331+0000000111S5104;
110 0604;111S1611I-0000007+4444442+0000000111S5104;
110 0604;111S1611I-0000007-5555553+0000000111S5104;
110 0604;111S1611I-0000010+6666664+0000000111S5104;
110 0604;111S1611I-0000010-7777775+0000000111S5104;
110 0604;111S1611I-0000011-1111106+0000000111S5104;
110 0604;111S1611I-0000012+2222217+0000000111S5104;
110 0604;111S1611I-0000012-3333330+0000000111S5104;
110 060IF
```

Terminated

Insert B-3. SELECTIVE TRACE PRINT-OUT

APPENDIX C

COMMAND CLASSIFICATION TABLES

Table C-1 shows the next instruction classification as programmed in Trace. It is assumed that an instruction (of the form s OP 1 in sector S of line L) when sequenced tagged will find its next instruction according to Table C-1. It is further assumed that any instruction which is not sequenced tagged will find its next instruction at the next sequential location (at sector S+1 of line L) unless a branch condition is satisfied, in which case, the next instruction, again, is according to Table C-1.

Table C-2 shows how commands are grouped by Trace for the various execute modes.

Table C-1

COMMAND SEQUENCING
CLASSIFICATION

LOCATION OF NEXT INSTRUCTION	COMMANDS
L s	00 (HALT), 20 to 26, (27), 30 to 33, 60 to 73, (74), (76), 77
L s+1	00 (MAC), 01, 02, 04 to 06, 10 to 12, 14, 15, 40 to 53, (54), 55 to 57
L s+2	03, 07, 13, 16, 17
l s	34 to 37, 75

Where

s = sector address of command

l = line address of command

L = line location of command

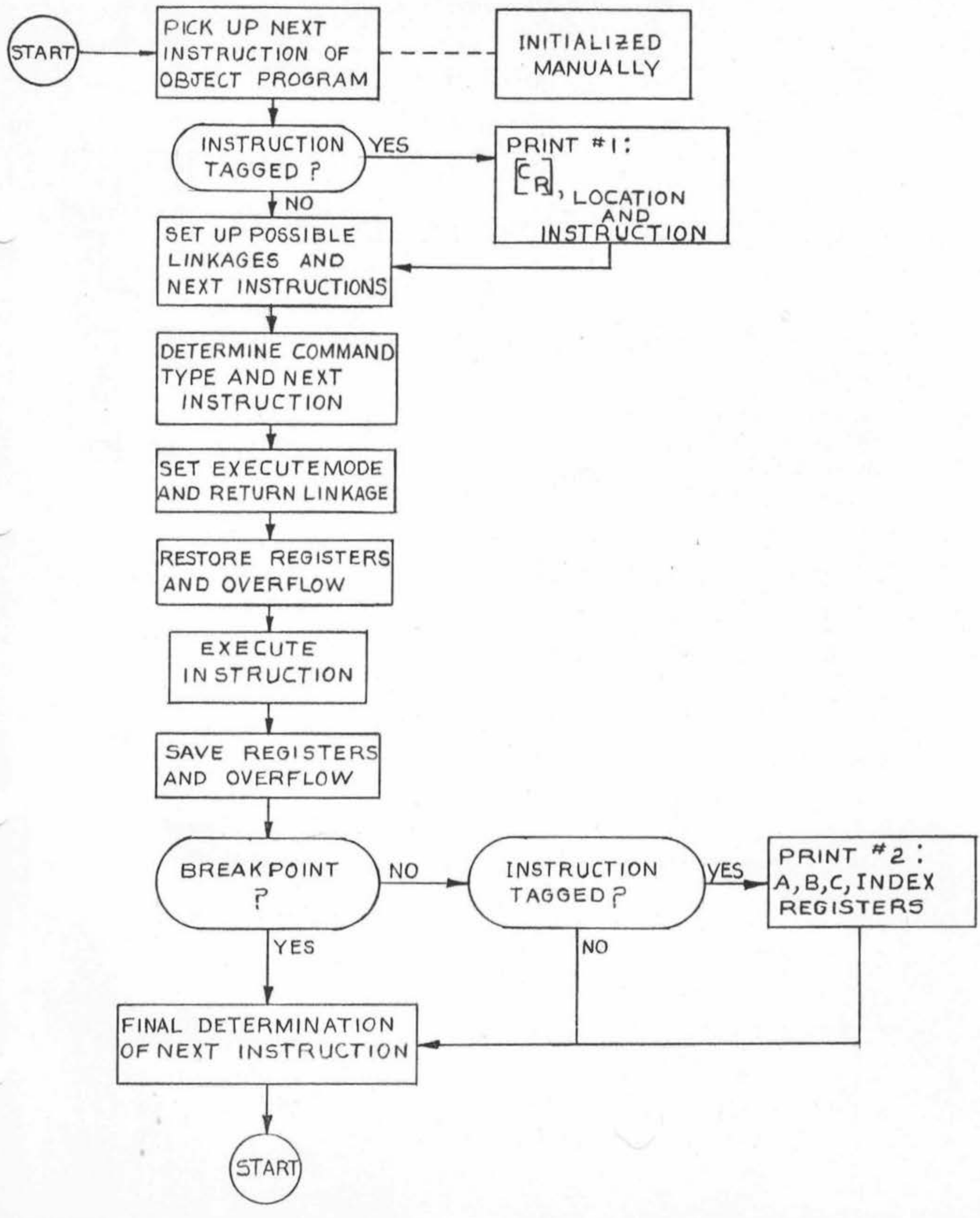
Table C-2

COMMAND EXECUTE MODE
CLASSIFICATION

COMMAND GROUP	COMMANDS	EXECUTE MODE
I	25, 26, (27), 71, 72, 73, (74)	Leave indexed, arbitrarily seq. tag, execute in actual location.
II	00, 20 to 23, 30 to 33 60 to 67	Leave indexed, strip of seq. tag, execute in actual location.
III-1	(76), 77	Leave indexed, strip of seq. tag, modify sector operand, execute in dummy location.
III-2	37	Simulate
III-3	34, 35, 36, 75	Strip of seq. tag and index, modify sector and line operands, execute in dummy location.
III-4	01 to 17, 24, 40 to 57, 70	Leave indexed, strip of seq. tag, execute in dummy location.

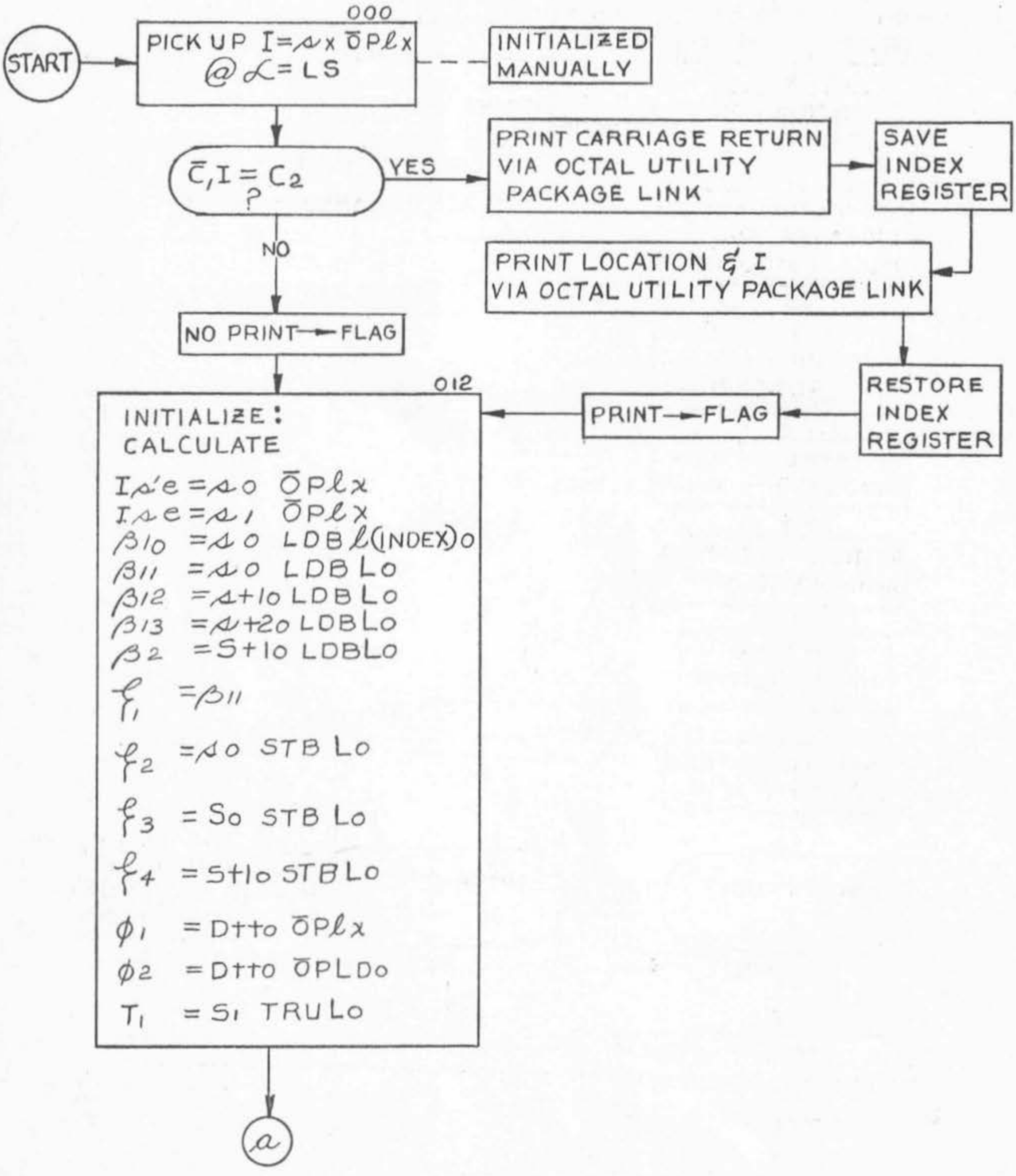
TRACE I APPENDIX D

GENERAL FLOW



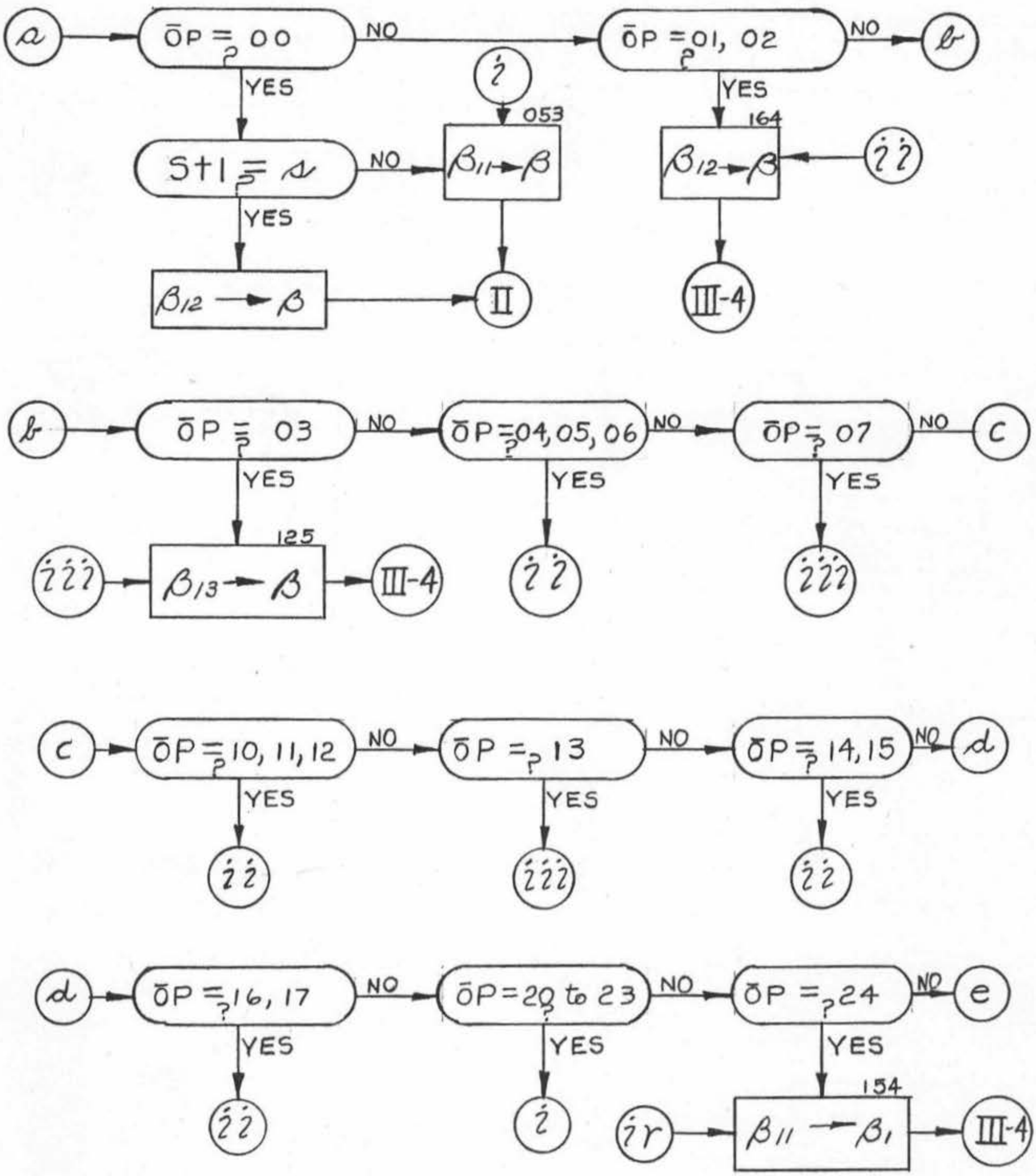
TRACE I APPENDIX D

DETAILED FLOW

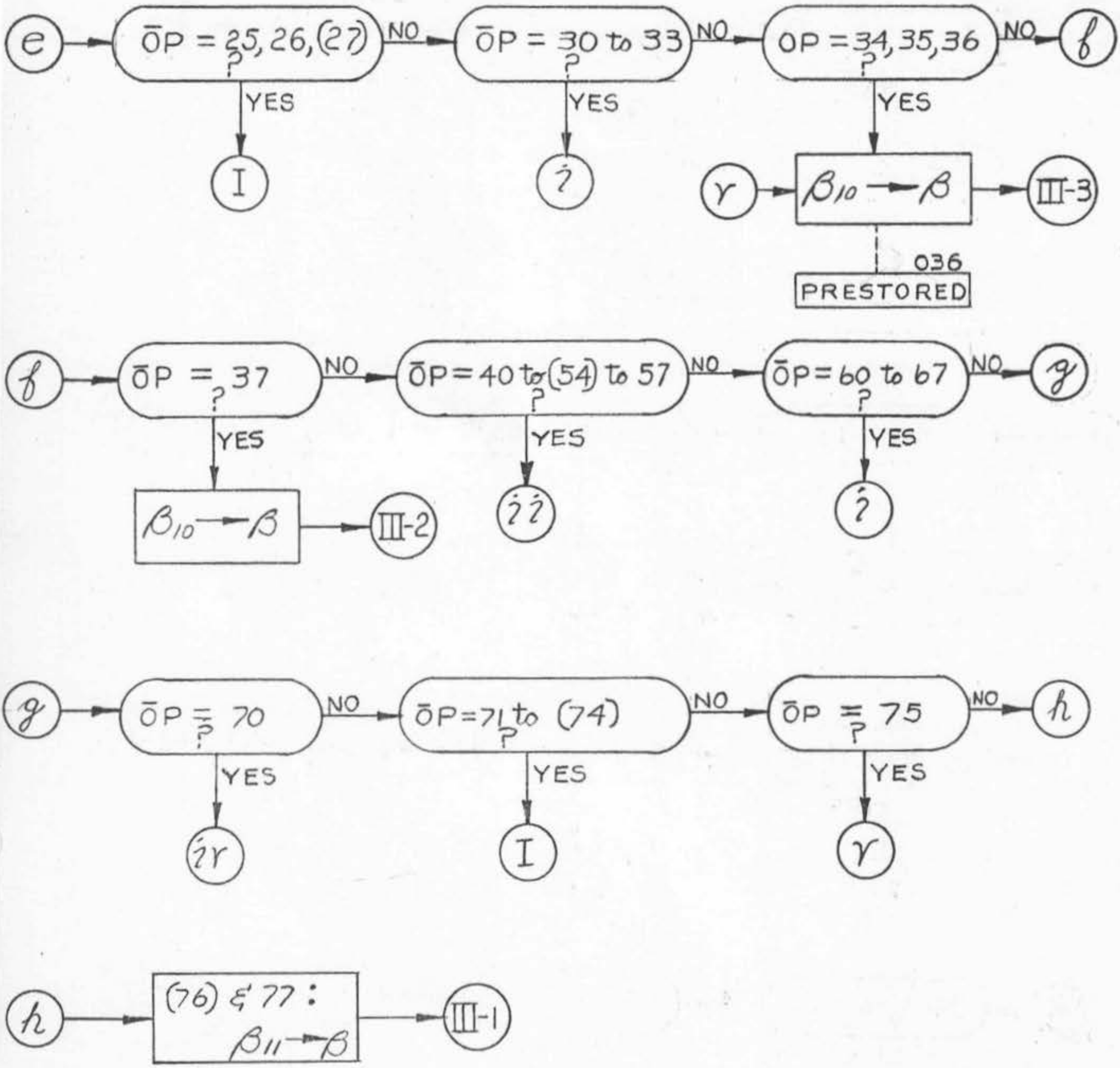


TRACE I

[D]ETAILED FLOW

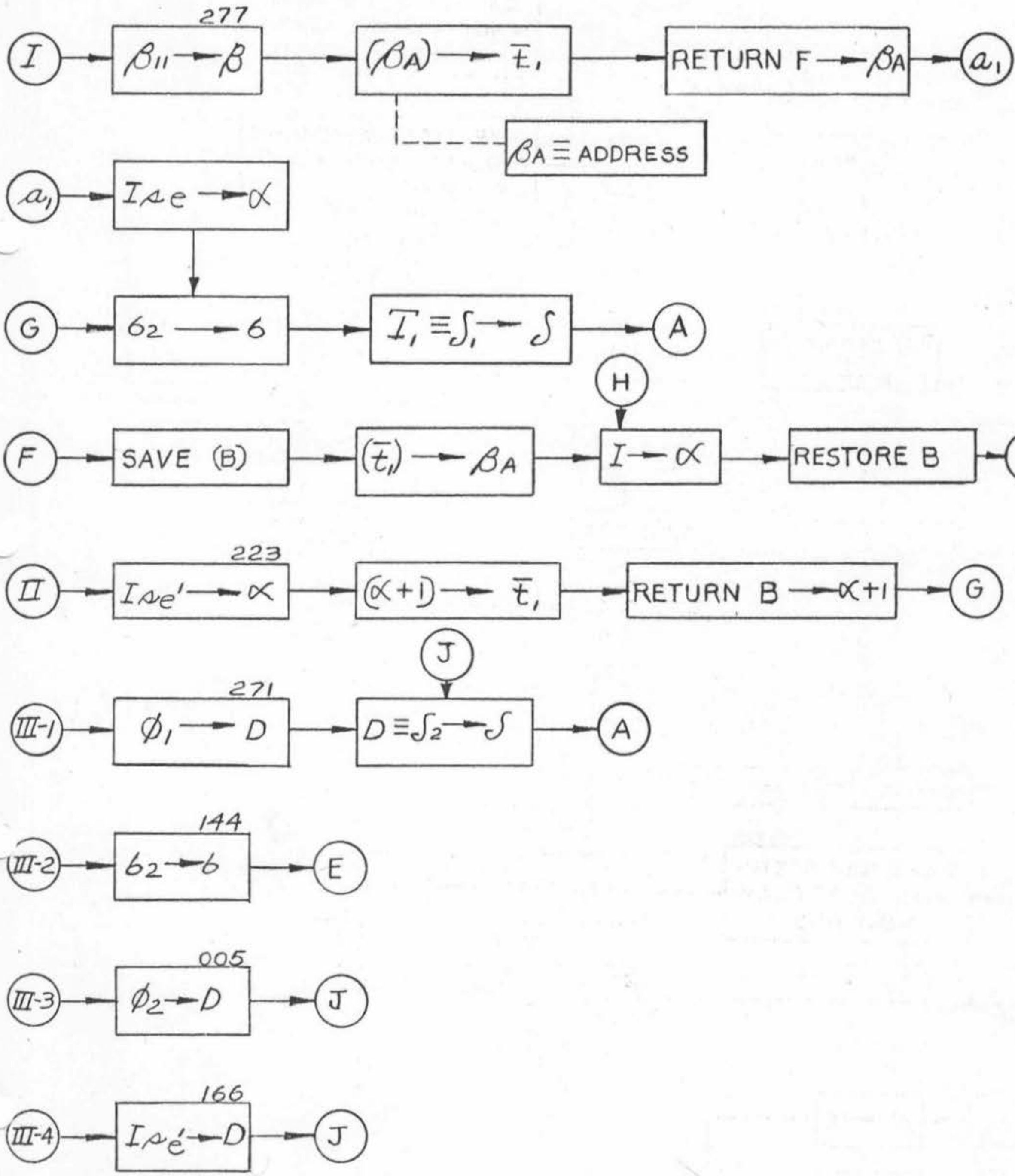


TRACE I
DETAILED FLOW



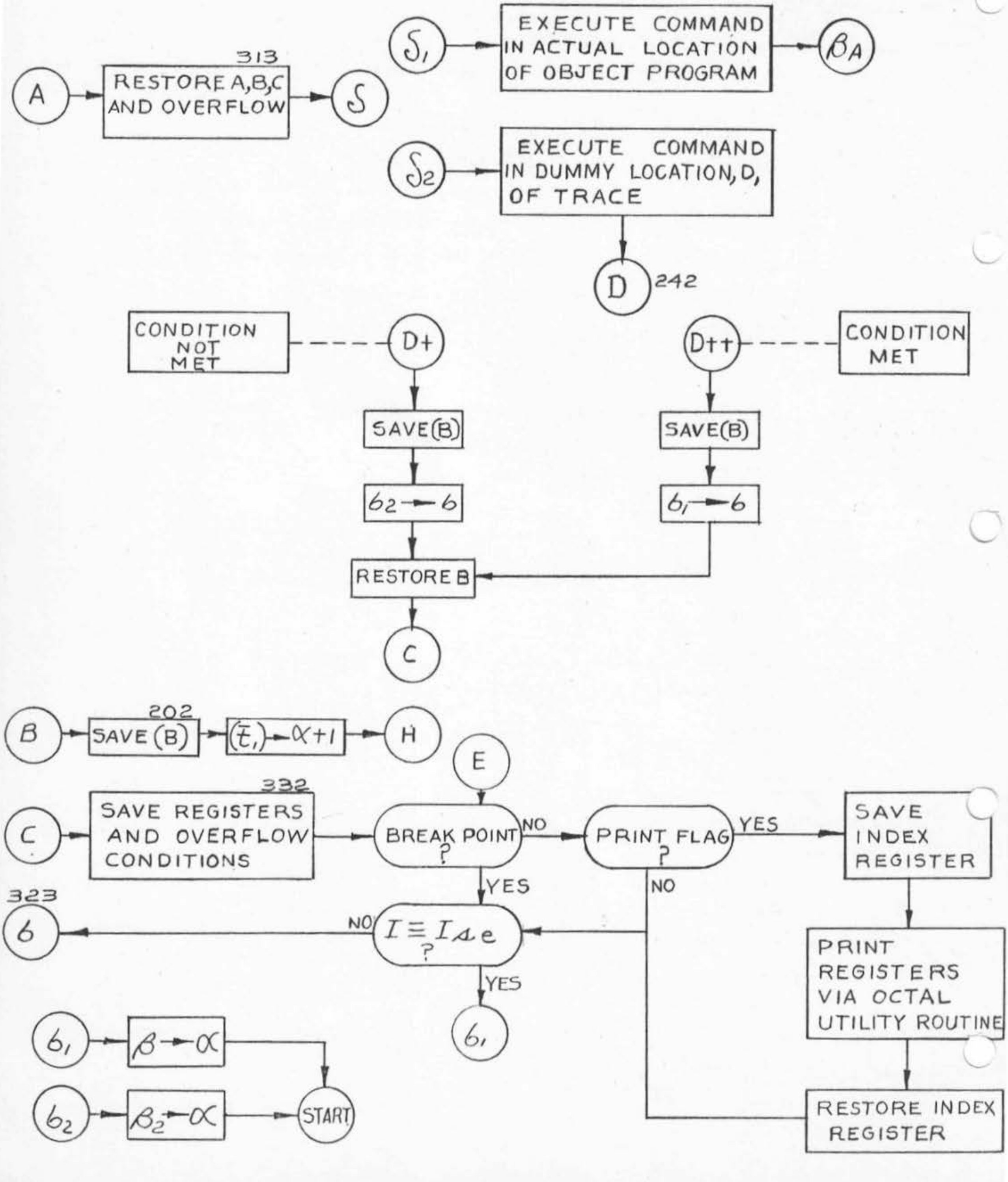
TRACE I

DETAILED FLOW



TRACE I

DETAILED FLOW



APPENDIX E

TRACE I LISTING

In the Trace listing, the presence of an XX indicates where the line number is to be inserted for Trace to be filled in that line. Since Trace is not relocatable, a seperate tape is required for each line. In addition, a special tape has to be prepared in order to restore the normal functions of the Octal Utility Package. This tape has only 33 frames and can be contained on the same tape as the Trace routine itself.

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE

DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
000XX	[000] 06[00];	LDB	PICK UP NEXT INSTRUCTION (I)
001	002S12XX;	STB	SAVE AS CURRENT INSTRUCTION
002	[ ]	TEMP	FOR CURRENT INSTRUCTION
003	141 47XX;	EXF	FORM $\overline{C}$, I
004	005S0300;	ROT	BRING TO A
005	335 06XX;	LDB	BRING IN $\emptyset_2$ = D++$_{00}$OP LD$_0$
006	272S37XX;	TRU	GO TO COMMON STORE
007	172 56XX;	CAM	COMPARE WITH C_1
010	175 75XX;	TOF	GO PRINT IF COMPARE
011	302 05XX;	LDA	NEG. WORD FOR NO PRINT FLAG.
012	002 06XX;	LDB	PICK UP I
013	014S47XX;	EXF	STRIP OF SEQ. TAG
014	000S0000;	CONST	+0020000
015	016S13XX;	STD	STORE $I_{\Delta e'}$ PRINT FLAG
016	[ ]	TEMP	FOR $I_{\Delta e'}$
017	[ ]	TEMP	FOR PRINT FLAG
020	016 05XX;	LDA	$I_{\Delta e'} \to A$
021	014 14XX;	ADD	ARBITRARILY SEQ. TAG. AND SAVE
022	120 11XX;	STA	
023	025S2200;	RSI	BRING LINE NUMBER TO B (IF INDEX, BRING INDEX)
024	025S0637;	LDB	
025	024 36XX;	TBN	
026	030S2100;	LSD	
027	000 0077I	CONST	+0000177
030	016 04XX;	LDC	BRING IN $\mathcal{A}_0$ AND ℓ (INDEX)$_0$
031	032S46XX;	AOC	
032	377S0000;	CONST	-7760000
033	000 04XX;	LDC	BRING IN LDB
034	035S46XX;	AOC	
035	000S7700;	CONST	+0037600
036	040S12XX;	STB	$\beta_{10} \to \beta$
	[ ]	TEMP	FOR REPLACED INSTRUCTION

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE

DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
040XX	[ ]	TEMP	FOR LOCATION OF NEXT INSTRUCTION
041	027 46XX;	AOC	BRING IN L_o
042	300 12XX;	STB	$\beta_{11} = \mathcal{T}_1 \rightarrow$
043	044S0300;	ROT	$\beta_{11} \rightarrow A$, $I_{Ae'} \rightarrow B$, $I_{Ae} \rightarrow C$
044	131 05XX;	LDA	BRING IN $\beta_{12} = \mathcal{A} + 1_o$ $LDBL_o$ GO TO II
045	223S37XX;	TRU	
046	001 1501;	SUB	FORM β_{12}
047	131 11XX;	STA	
050	001 1501;	SUB	FORM $\beta_{13} = \mathcal{A} + 2_o$ $LDBL_o$
051	151 11XX;	STA	
052	053S0300;	ROT	$I_{Ae'} \rightarrow A$, $I_{Ae} \rightarrow B$, $\beta_{13} \rightarrow C$
053	300 05XX;	LDA	BRING IN β_{11} GO TO II
054	223S37XX;	TRU	
055	001 1501;	SUB	FORM $\beta_2 = S+1_o LDBL_o$
056	225 11XX;	STA	
057	060S14XX;	ADD	MAKE STB
060	000 0400;	CONST	+0001000
061	231 11XX;	STA	$\rightarrow \mathcal{T}_4 = S+1_o$ $STBL_o$
062	204 11XX;	STA	
063	300 05XX;	LDA	TO MAKE $\mathcal{T}_2 = \mathcal{A}_o$ $STBL_o$
064	060 14XX;	ADD	
065	304 11XX;	STA	
066	326 11XX;	STA	
067	000 05XX;	LDA	TO MAKE $\mathcal{T}_3 = S_o$ $STBL_o$
070	060 14XX;	ADD	
071	306 11XX;	STA	
072	330 11XX;	STA	
073	224 11XX;	STA	
074	075S14XX;	ADD	MAKE $T_1 = S_1 TRU L_o$
075	000S2500;	CONST	
076	077S04XX;	LDC	BRING IN $D{+}{+}{}_o^{...}LD_o$
077	134 00XX;	CONST	

pb Packard Bell Computer
PB 250 PROGRAM LISTING

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE

DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
100XX	032 46XX;	AOC	FORM ϕ_1, = D $++_o$ OP ℓ_x AND T_1
101	102S13XX;	STD	
102	[ ]	TEMP	FOR ϕ_1
103	[ ]	TEMP	FOR $T_1 \equiv \delta$ TRANSFER, AND INDEX
104	027 46XX;	AOC	FORM ϕ_2 = D $++_0$ OPLD$_0$
105	335 12XX;	STB	
106	116S2200;	RSI	SCALE OP AS INTEGER
107			NOT USED
110			
111	250 07XX;	LDP	TO PRINT LOCATION
112	111 04XX;	LDC	
113	347S37XX;	TRU	
114	364 07XX;	LDP	TO PRINT INSTRUCTION
115	112S37XX;	TRU	
116	03~~2~~3 4701;	EXF	CLEAN OP $\longrightarrow$ C
117	120S0200;	IBC	
120	[ ]	TEMP	FOR I_{AC} AND (B)
121	123S2100;	LSD	OP - 1 $\longrightarrow$ C
122	[ ]	TEMP	FOR INDEX REG. PRINT
123	220 34XX;	TCN	MAC TEST
124	127S2100;	LSD	OP - 3
125	151 05XX;	LDA	BRING IN β_{13}
126	165S37XX;	TRU	GO TO COMMON STORE
127	164 34XX;	TCN	$1 \leq$ OP ≤ 2
130	132S2100;	LSD	OP - 4
131	[ ]	TEMP	FOR β_{12}
132	125 34XX;	TCN	OP = 3
133	137S2100;	LSD	OP - 7
134	120 12XX;	STB	D ++ (CONDITION MET)
135	254 06XX;	LDB	
136	245S37XX;	TRU	
137	164 34XX;	TCN	$4 \leq$ OP ≤ 6

PROBLEM TRACE I
PROGRAMMER PHIL JARVIE DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
140XX	142S2100;	LSD	OP - 10
141	377S7777I	CONST	PARAMETER C_1
142	125 34XX;	TCN	OP = 7
143	147S2100;	LSD	OP - 13
144	215 06XX;	LDB	
145	262 12XX;	STB	III - 2 SEQUENCE
146	342S37XX;	TRU	$\sigma_2 \rightarrow \sigma$
147	164 34XX;	TCN	10 ≤ OP ≤ 12
150	152S2100;	LSD	OP - 14
151	[ ]	TEMP	FOR β_{13}
152	125 34XX;	TCN	OP = 13
153	156S2100;	LSD	OP - 16
154	300 05XX;	LDA	BRING IN β_{11}
155	165S37XX;	TRU	GO TO COMMON STORE
156	164 34XX;	TCN	14 ≤ OP ≤ 15
157	162S2100;	LSD	OP - 20
160	370 07XX;	LDP	PRINT (B)
161	346S37XX;	TRU	
162	125 34XX;	TCN	16 ≤ OP ≤ 17
163	170S2100;	LSD	OP - 24
164	131 05XX;	LDA	ïï SEQUENCE
165	040 11XX;	STA	
166	016 06XX;	LDB	III - 4 SEQUENCE
167	272S37XX;	TRU	
170	053 34XX;	TCN	20 ≤ OP ≤ 23
171	173S2100;	LSD	OP - 25
172	[ ]	CONST	C_2 PARAMETER
173	154 34XX;	TCN	OP = 24
174	200S2100;	LSD	OP - 30
175	265 05XX;	LDA	
176	140 1101;	STA	PRINT CARRIAGE RETURN
177	136S3701;	TRU	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
200XX	277 34XX;	TCN	25 ≤ OP ≤ 27
201	206S2100;	LSD	OP - 34
202	120 12XX;	STB	
203	037 06XX;	LDB	Ⓑ SEQ. : RESTORE <u>NEXT</u> INST.
204	[ ]	STB	T_4
205	327S37XX;	TRU	
206	053 34XX;	TCN	30 ≤ OP ≤ 33
207	213S2100;	LSD	OP - 37
210	040 05XX;	LDA	
211	000 11XX;	STA	σ_1 SEQ. : NEXT INST.
212	000S37XX;	TRU	
213	005 34XX;	TCN	34 ≤ OP ≤ 36
214	216S2100;	LSD	OP - 40
215	362S37XX;	TRU	σ_2 SWITCH TRANSFER
216	144 34XX;	TCN	OP = 37
217	240S2100;	LSD	OP - 60
220	300 05XX;	LDA	
221	225 56XX;	CAM	MAC TEST
222	044 75XX;	TOF	
223	016 06XX;	LDB	
224	[ ]	STB	II SEQ. T_3
225	[ ]	LDB	β_2
226	037 13XX;	STD	
227	230S06XX;	LDB	
230	202S37XX;	TRU	
231	[ ]	STB	T_4
232	307S37XX;	TRU	
233	103 05XX;	LDA	RETURN FROM PRINT
234	000 1137;	STA	SAVE INDEX REG.
235	237 2100;	LSD	SETUP " PRINT INDEX"
236	122 11XX;	STA	
	[ ]	TRU	RETURN TO PRINT CONTROL

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM TRACE I —

PROGRAMMER PHIL JARVIE — DATE 1-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
240XX	164 34XX;	TCN	40 $\leq$ OP $\leq$ 57
241	252S2100;	LSD	OP - 70
242	[ ]	TEMP	D FOR EXECUTE IN DUMMY LOCATION
243	120 12XX;	STB	
244	215 06XX;	LDB	
245	262 12XX;	STB	D+ (CONDITION NOT MET)
246	120 06XX;	LDB	
247	332S37XX;	TRU	
250	+00000XX;	CONST	LOCATION PRINT CONSTANTS
251	114S37XX;	TRU	
252	053 34XX;	TCN	60 $\leq$ OP $\leq$ 67
253	255S2100;	LSD	OP - 71
254	210S37XX;	TRU	σ_1 SWITCH TRANSFER
255	154 34XX;	TCN	OP = 70
256	263S2100;	LSD	OP - 75
257	002 05XX;	LDA	
260	277 2200;	RSI	FINAL TEST FOR NEXT INSTRUCTION
261	210 36XX;	TBN	
262	[ ]	TRU	σ SWITCH
263	277 34XX;	TCN	71 $\leq$ OP $\leq$ 74
264	266S2100;	LSD	OP - 76
265	111S37XX;	TRU	RETURN FOR C.R. PRINT
266	005 34XX;	TCN	OP = 75
267	300 05XX;	LDA	OP > 75, $\therefore \beta_{11} \rightarrow \beta$
270	040 11XX;	STA	
271	102 06XX;	LDB	
272	242 12XX;	STB	
273	274S06XX;	LDB	III - 1 SEQ. , $\delta_2 \rightarrow \delta$
274	242S37XX;	TRU	δ_2 SWITCH TRANSFER
275	323 12XX;	STB	
276	313S37XX;	TRU	START OF I SEQ.
277	300 05XX;	LDA	

pb Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE

DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
300XX	[ ]	LDB	$\zeta_1 \equiv \beta_{11}$
301	037 13XX;	STD	I SEQ.
302	303S06XX;	LDB	ALSO NEGATIVE WORD
303	324S37XX;	TRU	RETURN FOR I
304	[ ]	STB	ζ_2
305	120 06XX;	LDB	
306	[ ]	STB	ζ_3
307	215 05XX;	LDA	
310	262 11XX;	STA	G
311	103 06XX;	LDB	
312	323 12XX;	STB	$\delta_1 \rightarrow \delta$
313	314S05XX;	LDA	A SEQ.
314	377S7777I	TEMP	FOR OVERFLOW FLAG ($\neq 0$ OFF, 0 ON)
315	061 5601;	CAM	SET OVERFLOW
316	317S07XX;	LDP	RESTORE A & B
317	[ ]	TEMP	FOR (B)
320	[ ]	TEMP	FOR (A)
321	322S04XX;	LDC	RESTORE C
322	[ ]	TEMP	FOR (C)
323	[ ]	TRU	δ SWITCH
324	120 12XX;	STB	
325	037 06XX;	LDB	
326	[ ]	STB	F SEQ. , ζ_2
327	~~2~~002 06XX;	LDB	
330	[ ]	STB	ζ_3
331	120 06XX;	LDB	
332	317 13XX;	STD	
333	322 10XX;	STC	
334	335S4300;	CLB	C SEQ.
335	[ ]	TEMP	FOR ϕ_2
336	341 75XX;	TOF	
337	340S06XX;	LDB	

PROBLEM TRACE I

PROGRAMMER PHIL JARVIE

DATE 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
340XX	233 75XX;	TOF	RETURN LINK FROM PRINT, ALSO $\neq$ 0
341	314 12XX;	STB	END OF C SEQ.
342	257 7735;	TES	
343	017 05XX;	LDA	E SEQ.
344	257 35XX;	TAN	
345	366 07XX;	LDP	PRINT OUT NO. 2
346	350 04XX;	LDC	ANY POS. WORD
347	237 11XX;	STA	PRINT SUB-ROUTINE
350	144 0537;	LDA	SAVE INDEX
351	103 11XX;	STA	
352	340 05XX;	LDA	PLANT RETURN LINK
353	046 1101;	STA	
354	346 3401;	TCN	COMMAND FORMAT
355	310S3701;	TRU	DATA FORMAT
356	372 07XX;	LDP	PRINT (C)
357	346S37XX;	TRU	
360	374 07XX;	LDP	PRINT (INDEX)
361	112S37XX;	TRU	
362	225 05XX;	LDA	σ_2 SEQ.
363	211S37XX;	TRU	
364	+00002XX	CONST	INSTRUCTION PRINT CONSTANTS
365	012S37XX;	TRU	
366	+00320XX	CONST	(A) PRINT CONSTANTS
367	160S37XX;	TRU	
370	+00317XX	CONST	(B) PRINT CONSTANTS
371	356S37XX;	TRU	
372	+00322XX	CONST	(C) PRINT CONSTANTS
373	360S37XX;	TRU	
374	+00122XX	CONST	(INDEX) PRINT CONSTANTS
375	257S37XX;	TRU	
376			NOT USED
377			

Packard Bell Computer

PB 250 PROGRAM LISTING

PROBLEM: SPECIAL 33 FRAMES FOR TRACE

PAGE 1 OF 1

PROGRAMMER: P. JARVIE

DATE: 2-13-61

LOCATION	INSTRUCTION	SYMBOLIC OP CODE	REMARKS
04601$	136 7501;	TOF	
14001$	~~200~~ 307 S3701;	TRU	
W			