

JAN 29 1960

SDC

FIELD NOTE A WORKING PAPER

M. Blauer
FN - 10 - 205

The views, conclusions, and recommendations expressed herein do not necessarily reflect the official views or policies of either the Air Force or the System Development Corporation.

Although this working paper contains no classified information it has not been cleared for open publication by the Department of Defense. Open publication, wholly or in part, is prohibited without prior approval of the System Development Corporation.

AUTHOR
K. J. Theting R. L. Small
K. J. Theting R. L. Small
APPROVED
M. Blauer
M. Blauer

(Produced under System Development Corporation sub-contract No. 202 issued by International Electric Corporation in performance of contract AF-30(635)-11583)

PAGE 1 OF 42 PAGES

SYSTEM DEVELOPMENT CORPORATION • 2500 COLORADO AVENUE • SANTA MONICA, CALIFORNIA



PROGRAM DESIGN SPECIFICATION FOR
THE DATA PROCESSING PROGRAM (JDSYZ) --
A SUB-PROGRAM OF THE JOVIAL INTERPRETER SYSTEM



PROGRAM DESIGN SPECIFICATION FOR
THE DATA PROCESSING PROGRAM (JDSYZ) ---
A SUB-PROGRAM OF THE JOVIAL INTERPRETER SYSTEM

TABLE OF CONTENTS

Introduction	Page 3
Environment	Page 5
Input	Page 10
Functions	Page 14
Output	Page 27
Sample Output Pages	Page 28
Message Printouts	Page 33
Flow Diagram	Page 36

PROGRAM DESIGN SPECIFICATION FOR
THE DATA PROCESSING PROGRAM (JDSYZ) ---
A SUB-PROGRAM OF THE JOVIAL INTERPRETER SYSTEM

INTRODUCTION

The Data Processing Program (hereinafter referred to as JDSYZ) is the last program in the Interpreter System to operate and is designed to furnish the programmer with meaningful printouts to aid in debugging a program coded in JOVIAL. JDSYZ will analyze the table and item settings to see if they were altered by the JOVIAL-coded program in the manner specified by the programmer in his expected results deck. All data will then be output in a manner reflecting the results of this analysis. JDSYZ, in addition, will print a trace of the Interpreter SET operations for each item entry found to be in error or, at the option of the programmer, will print a full trace of all Interpreter SET and TRANSFER operations in the order in which they occurred. Finally, a printout of all ILT entries will be provided to aid in checking the logic of the JOVIAL-coded program. A brief summary of the operations of JDSYZ follows.

Two of the four input tapes used by JDSYZ are read in when the program first operates. One tape contains the Baby Compool made up by JABCZ, the Assemble Baby Compool Program, and the initial values output from JSTRZ, the Data Simulation Program. The second tape is also output from JSTRZ and contains the expected values.

The description of each table, item, and value entry in Baby Compool is then printed in a heading type format. Below each heading appears the result of comparisons made of the initial, final, and expected data values pertaining to the respective Compool entry.

The third tape is read in if discrepancies were discovered between the expected and final results or if the programmer requested a full trace via JTCPZ, the Test Control Program. The tape is output by a subroutine internal to JOLLZ, the Second Pass of the Interpreter, and contains all SET operations in the order performed by JOLLZ and, if a full trace, also contains all TRANSFER operations in their order of occurrence. In the automatic trace mode, JDSYZ, during the previous data comparison, will have saved all entries which had final settings differing from their expected results. Utilizing this information, JDSYZ will print the "ILT SET" information for each difference found. In the full trace mode, JDSYZ will print all SET and TRANSFER information in its order on the input tape.

The fourth tape, an output of JALLZ, Interpreter First Pass, contains the following tables generated by JALLZ:

1. SLT - Statement Label Table
2. ILT - Intermediate Language Table
3. CON - Constant Table
4. SUB - Subscript Table
5. VAT - Variable Table
6. SWT - Switch Table

Utilizing these tables, JDSYZ will translate each ILT entry to prose form, and starting with ILT 0 will printout all ILT entries.

ENVIRONMENT

Data Tables

Initial		Expected		Final	
∅	Table A Control Word	∅	Table A Control Word	∅	Table A Control Word
1	Item X Entry ∅	1	Item X Entry ∅	1	Item X Entry ∅
2	Item X Entry 1	2	Item X Entry 1	2	Item X Entry 1
3	Item Y Entry ∅	3	Item Y Entry ∅	3	Item Y Entry ∅
4	Item Y Entry 1	4	Item Y Entry 1	4	Item Y Entry 1
n-1	Table B Control Word	n-1	Table B Control Word	n-1	Table B Control Word
n	Item Z Entry ∅	n	Item Z Entry ∅	n	Item Z Entry ∅

The Initial Table contains the initial conditions, while the Expected Table contains the results which the programmer expects to appear in the Final Table.

The location in core and the contents of the Final Table are determined by the Interpreter Program, while the Initial and Expected Tables are made up by JSTRZ and read into locations dynamically assigned by JDSYZ.

The tables set or used by the Object Program and contained in the three data tables will be parallel tables of fixed length. Program-table control words and item entries are in the same relative positions within each data table, as shown in the condensed version above.

ILT Tables

See FN-LO-201 for detailed format of each internal table. This document also describes and illustrates the various formats of an ILT entry.

Baby Compool Format -- Table A

Only Parallel Tables are described.

Control Word contains:

- NENT: Number of entries in Compool
- NWDS: Number of words in Compool including the control word

NENT	NWDS
COMTAG (Table)	
TAG TAB SET VAR TYP TYP USE DEF	WDSTRY NENTRY
ABSADD	
COMTAG (Item)	
TAG SIG SET VAR TYP NED USE DEF	ITCODE PNTPOS ADDSTA NBITS NSTATI
TABREF	ABSADD
COMTAG (Value)	
TAG SIG SET VAR TYP NED USE DEF	ITCODE PNTPOS NBITS
VALUE	

<u>TAG</u>	<u>BITS</u>	<u>DEFINITION</u>
COMTAG	0-35	Hollerith tag left justified
TAGTYP	0,1	0 = Item 1 = Table 2 = Value
TABTYP	2	0 = Fixed length 1 = Variable length
SIGNED	2	0 = Unsigned item 1 = Signed item
SETUSE	3,4	0 = Null 1 = Set 2 = Used 3 = Both
VARDEF	5,6	1 = Variable defined by the program 2 = Variable defined by the Master Compool 3 = Compool definition overridden by program
WDSTRY	7-17	Number of blocks in this table
NENTRY	18-35	Number of words per block
ABSADD	18-35	Absolute address of table control word, or absolute address of first register containing this item.
PNTPOS	18-29	Mixed fractions. The number of bits to the right of the binary point.
ADDSTA	18-29	Relative address in Table B of the block of registers containing the legal statuses used by this program.
NBITS	30-35	Number of bits in item including the sign bit.
NSTATI	30-35	Number of statuses for status-coded items.
ITCODE	14-17	The item coding: 0 = ST 1 = FP 2 = FI 3 = BH 4 = AD 5 = MX
VALUE	0-35	The constant described by this value entry.
TABREF	0-17	The relative address in Table A of the first word of the table entry which describes the table this item belongs to.

Table of Differences (Expected vs. Final)

TILT

Number of words following (2's complement)	
Item A	Comtag
Entry	3
Entry	7
Entry	13
Entry	17
Item D	Comtag
Entry	1
Entry	5
Entry	12

For the first difference occurring during comparison of an item's data, both the name of the item in Hollerith and the present entry number of the item are stored in TILT. For following differences of this item, the entry number only is stored.

Processing Table

PRO

Expected result
Final result
Initial setting
Entry number
Expected result
Final result
Initial setting
Entry number

PRO is the storage table in which Status, BH, or AD data is saved until it can be written on the DLO tape. The core allocation for PRO will be computed by JDSYZ. PRO will not exist if the data tables occupy all of core following JDSYZ; no data will be saved during comparison.

Communication Registers

TABREG

OP	DECREMENT	TAG	ADDRESS
	Number of words in Final Table		Starting address of Final Table

Set by Interpreter First Pass, used by JDSYZ to allocate core for Baby Compool, Initial and Expected Tables.

COMREG

OP	DECREMENT	TAG	ADDRESS
▨	Number of words in Baby Compool	▨	Starting address of Baby Compool

Set by JABCZ. JDSYZ will reset with a new starting address.

MATTBL

	OP	DECREMENT	TAG	ADDRESS
	▨	Number of words in CON Table	▨	Absolute address of CON Table
+1	▨	Number of words in SUB Table	▨	Absolute address of SUB Table
+2	▨	Number of words in VAT Table	▨	Absolute address of VAT Table
+3	▨	Number of words in SWT Table	▨	Absolute address of SWT Table
+4	▨	Number of words in ILT Table	▨	Absolute address of ILT Table
+5	▨	Number of words in SLT Table	▨	▨

Used by JDSYZ during the processing of ILT entries for ILT print.

TRAP1 and TRAP2

TRAP1
TRAP2

Set by JTCPZ and JOLLZ.

TRAP1 = \emptyset + TRAP2 $\neq \emptyset$, automatic trace.
 TRAP1 + TRAP2 both $\neq \emptyset$, full trace.
 TRAP1 + TRAP2 both = \emptyset , no trace (set by JOLLZ only).

Sense Indicators

	14	
--	----	--

B14 set by JSTRZ, JDSYZ

∅ = No expected results
1 = Expected results

1	2	3	4		20	21	22	23	24	25	26	27
---	---	---	---	--	----	----	----	----	----	----	----	----

Indicators set/used internally by JDSYZ and restored to zero when control is returned to JTCPZ.

- B1 - ∅ = No arithmetic or exponent operator.
1 = Arithmetic or exponent operator.

- B2 - ∅ = (1) Comparison - No automatic trace.
 (2) ILT Print - No "SET TO" operator.
1 = (1) Comparison - Automatic trace.
 (2) ILT Print - "SET TO" operator.

- B3 - ∅ = Right-half term not processed.
1 = Right-half term processed.

- B4 - ∅ = No special subscripted variable.
1 = Special subscripted variable.

- B2∅ - ∅ = No heading printed for arithmetic-type item.
1 = Heading printed for arithmetic-type item.

- B21 - ∅ = No item tag in TILT Table.
1 = Item tag in TILT Table.

- B22 - ∅ = Entry to be printed.
1 = Entry to be saved in PRO Table.

- B23 - ∅ = (1) Comparison - No expected results for the item.
 (2) Trace - No Compool search.
1 = (1) Comparison - Expected results for the item.
 (2) Trace - Compool search.

- B24 - ∅ = Do not convert data to BCI.
1 = Convert data to BCI.

- B25 - ∅ = Not AD coded item.
1 = AD coded item.

- B26 - ∅ = Not FP coded item.
1 = FP coded item.

- B27 - ∅ = PRO available for this item.
1 = PRO not available for this item.

INPUT

Tape C1

File 1 - Record 1	Identification of SLT Table
	Number of Words in SLT Table
Record 2	SLT Table
Record 3	End of File
File 2 - Record 1	Identification of First ILT Table
	Number of words in First ILT Table
Record 2	First ILT Table (Relative Locations)
Record 3	End of File
File 3 - Record 1	Identification of CON Table
	Number of words in CON Table
Record 2	CON Table
Record 3	End of File
File 4 - Record 1	Identification of SUB Table
	Number of words in SUB Table
Record 2	SUB Table
Record 3	End of File

File 5 - Record 1	Identification of VAT Table
	Number of words in VAT Table
Record 2	VAT Table
Record 3	End of File
File 6 - Record 1	Identification of SWT Table
	Number of words in SWT Table
Record 2	SWT Table
Record 3	End of File
File 7 - Record 1	Identification of 2nd ILT Table
	Number of words in 2nd ILT Table
Record 2	2nd ILT Table
Record 3	End of File
File 8 - Record 1	Identification of BABY COMPOOL
	Number of words in BABY COMPOOL
Record 2	BABY COMPOOL
Record 3	Identification of Initial Conditions
	Number of words in Initial Conditions
Record 4	Initial Conditions

Tape D2

File 1 - Record 1

Expected Results Identification (E)
Number of words in Expected Results

Record 2

Expected Results

Tape D1

File 1	S	DECREMENT	TAG	ADDRESS
1	-	to where in ILT - 2's compl.		from where in ILT - 2's compl.
2	+	relative location of sub- script - 2's complement.		ILT present Loc. - 2's compl.
Record 1	3	Value stored into data tables (Binary)		
	4	Hollerith name of item		
	5	+		Value of subscript
	↓	~~~~~		
Record n	n	~~~~~		

In the above example, word one or record 1 illustrates a snap* of an ILT Transfer, note that the sign is negative. Words two through five contain a snap of an ILT store operation, note that in word two the sign is positive. ILT Transfer snaps will appear only when a full trace is called for. All snaps on the tape will be randomly distributed, insofar as the type of snap is concerned. No record will exceed 100 words and all records will be in FILE 1.

Card Input

No cards are introduced directly by JDSYZ. There exists a card input via JTCPZ which requests a full trace. JTCPZ communicates this information to JDSYZ via TRAP1, TRAP2.

* The term "snap" refers to a "snapshot" or recording of a particular ILT operation performed by the Interpreter Second Pass (JOLLZ).

FUNCTIONS

Read In

- A. Compute Storage allocations, using TABREG and COMREG.
1. Initial Conditions Table Address =
Final Table Address - NWDS in Final
 2. Expected Results Table Address =
Final Table Address - 2(NWDS in Final)
 3. Baby Compool Table Address =
Final Table Address - 2(NWDS in Final)-NWDS in Baby Compool
 4. Storage capacity of PRO =
Baby Compool Table Address - PRO Table Address.
- B. Select tapes and read into core
1. Read in Baby Compool from C1
 2. Read in initial conditions and expected results simultaneously
from C1 and D2 respectively.
- C. Error checks provided:
1. Baby Compool identification incorrect. Log comment, return control
to JTCPZ.
 2. Baby Compool tape parity on read-in. Log comment, return control
to JTCPZ.
 3. Initial conditions identification incorrect. Log comment, indicate
no initial data and proceed normally.
 4. Initial conditions tape parity on read-in. Log comment, proceed
as in 3 above.
 5. Expected results identification incorrect. Log comment, indicate
no expected data (SI Bit 14 = \emptyset), proceed normally.
 6. Expected results tape parity on read-in. Log comment, proceed
as in 5 above.
 7. EOT, EOF, tape unit unassigned. Log comment, halt and transfer to
B above.
- D. Test for automatic trace.
If TRAP1 = \emptyset , TRAP2 $\neq \emptyset$, set Sense Indicator Bit 2 = 1. This will cause
JDSYZ to save in the TILT Table those entries having differing Final vs.
Expected data.

Disassemble Table Data

The data for the first (next) table heading is found in Table A of the Baby Compool. Store the contents of the three word entry into the table heading image. Start a new page and write as a line on the DLO tape the following information:

- A. Hollerith name of table.
- B. Table type - fixed or variable length.
- C. Number of blocks (WDSTRY).
- D. Number of words per block (NENTRY).
- E. Absolute location of table within Final Table (ABSADD).
- F. Set or used by program being tested.
- G. Method by which table was defined (program, Master Compool, Compool Override).

Disassemble Item Data

Using the information from the table data disassembly, find the first item within Table A of Baby Compool associated with the data table disassembled. Using the following information from Table A, load the Item Image and write as a line on DLO tape.

- A. Hollerith name of item.
- B. Type of coding (BH, ST, AD, FI, FP, MX).
- C. Absolute location of item within Final Table (ABSADD).
- D. Set or used by program being tested.
- E. Method by which item was defined (program, Master Compool, Compool Override).
- F. Number signed or unsigned (FP, FI, MX only).
- G. Number of bits occupied by number (FI, MX, FP only).
- H. Number of statuses (ST only).

Actual legal status names are written on the following line, from Table B.

Set Up to Process Data Tables for the Item

Before entering the subroutine PROC, which will process the table data, set indicators and perform modifications shown below whenever they apply.

- A. SI Bit 22 - When set on, indicates a BH, ST, or AD coded item.
- B. SI Bit 24 - When set on, indicates AD, FI, FP, MX data which must be converted to BCI.
- C. SI Bit 25 - When set on, indicates AD coded item.
- D. SI Bit 26 - When set on, indicates FP coded item.
- E. Set number-of-entry counter in PROC.
- F. If FP, MX, or FI, modify conversion routine (CONV).
 - 1. FP - LRS 27 stored in CONV will be used to split exponent and fraction.
 - 2. MX - LRS (PNTPOS) stored in CONV will be used to split integer and fraction.
 - 3. FI - LRS \emptyset stored in CONV, used to denote no fraction to convert to BCI.

Process Table Data for Item Using PROC

A. For each entry in Initial, Expected and Final Tables

1. If no Expected Table (SI Bit 14 = \emptyset) or Expected for entry equals $+\emptyset$, insert BCI blank into Expected Image or PRO, skip to A6. A negative zero in Expected Results Table indicates Final should = $+\emptyset$.
2. If there is an expected result and it is the first one found (SI Bit 23 = \emptyset), for BH, ST, or AD coded items, set Sense Indicator Bit 23 = 1 to indicate expected result print format.
3. Expected = Final, insert BCI asterisk into Expected in Image or PRO, skip to A6.
4. Expected \neq Final, insert BCI value into Expected in Image or PRO. If necessary, first convert to BCI from binary (SI Bit 25 = 1).
5. If an automatic trace was requested, (SI Bit 2 = 1), and this is the first difference found for this item, (SI Bit 21 = \emptyset), store item name (HOLLERITH) in TILT and reset SI Bit 21 = 1. For this and all following differences for the item, the entry number is stored in TILT. A count is kept of the number of differences found and the total printed off-line when all items have been processed.
6. Final = Initial, insert BCI asterisk into Final in Image or PRO, skip to A8.
7. Final \neq Initial, insert BCI value into Final in Image or PRO. First convert if AD, FP, FI, MX coded.
8. Insert BCI value into Initial in Image or PRO. First convert if AD, FP, FI, MX coded.
9. Convert entry number to BCI and store in Image or PRO.
10. If all values = $+\emptyset$, delete entry data from Image or PRO.
11. If entry data in Image, (SI Bit 22 = \emptyset), print one-line DLO.
12. If not last entry for item, get next entry data and repeat A1 through A12. When last entry, go to B or C, whichever applies.

- B. If no data for item - Log comment and return to disassemble next item or table.
- C. Make up output format. When all entries have been analyzed, the program returns to disassemble the next item or table unless the data has been saved in the table, PRO, in which case it proceeds as follows:
 - 1. No expected results for this item (SI Bit 23 = \emptyset) write on DLO tape a four-column heading, followed by four columns of entries and their data.
 - 2. Expected results for this item (SI Bit 23 = 1) write on DLO tape a three-column heading, followed by three columns of entries and their data.

Tracing

After table and item information processing is finished, JDSYZ performs the trace operations in the sequence described below.

- A. See if trace is to be performed.
 - 1. No trace requested (TRAP2 = \emptyset). Skip to ILT Print Routine.
 - 2. Auto trace requested (TRAP2 $\neq \emptyset$, TRAP1 = \emptyset), but no differences discovered by PROC (TILT control word = \emptyset). Skip to ILT Print Routine.
- B. Move Baby Compool inside the environment of JDSYZ, clobbering that portion of JDSYZ no longer needed, and thus increasing the available core space.
- C. Read the snap tape (D-1) into core from Baby Compool Final Address +1 up to and including the end of core. The following read-in returns are provided:
 - 1. Tape Unit unassigned.
Log comment, halt and transfer back to read again.
 - 2. Tape parity
Log comment, set TRAP1 $\neq \emptyset$ to give an incomplete full trace, transfer to D.
 - 3. Snap tape exceeds storage capacity.
Log comment, set TRAP1 $\neq \emptyset$ to give an incomplete full trace, transfer to D.
 - 4. No snaps on snap tape.
Log comment, skip to ILT Print Routine.
 - 5. Normal
Proceed to D.

- D. Determine type of trace to be performed and set controls.
1. Get absolute location of ILT Table from MATTBL⁴ and save for ILT entry number conversion.
 2. Store count of snaps read in into loop controls.
 3. If a full trace, print on DLO tape "FULL TRACE FOLLOWS." Proceed to F.
 4. If an automatic trace, print on DLO tape "AUTO TRACE FOLLOWS." Store count from TILT Table into loop controls.
- E. Perform automatic trace using the data words in TILT as a guide. For each data word in TILT, perform the following:
1. Check TILT data word for item name. If not an item name, skip to item entry routine (E5).
 2. Save item name and print DLO.
 3. Inform Compool Search Routine (SEARCH) that a new item must be decoded (SI Bit 23 = 1).
 4. Print trace headings for this item on DLO tape and return for next TILT word (E.1).
 5. Search snap data table for an item name and subscript value equal to the stored item name and the entry number secured from the TILT Table.
 6. For each matching snap, go to the Snap Processing Routine (F.1) until the entire snap table has been searched.
 7. If no match found, print DLO "NO SET XXXXXX (Item Name) XXXXXX (Entry #)".
 8. If this is the final TILT word, then go to ILT Print Routine; otherwise return for the next TILT word.
- F. Perform full trace by taking each snap in the order found on tape and running it through the Snap Processing Routine below.
1. Using the absolute ILT location given in the first word of the snap, compute the ILT entry number as follows:
$$\frac{(\text{PRESENT ILT ABSOLUTE LOCATION}) - (\text{ILT TABLE ABSOLUTE LOCATION})}{5}$$

The quotient (ignore the remainder) = ILT entry #.

2. If the snap is a four-word set snap (sign of first word positive), skip to F.4. If the snap is a one-word transfer snap (sign minus), convert the absolute ILT location branched to into a relative ILT entry number, using the method in F.1.
3. Set up a one-line transfer snap format utilizing the information from F.1 and F.2 and skip to F.1 \emptyset . Transfer snaps will occur only in a full trace.
4. Set snap. Get subscript letter or number. If a number, convert to BCI.
5. Get subscript value from fourth word of snap, convert to BCI and to octal.
6. If third word in snap $\neq \emptyset$, transfer to F.7; otherwise, set up a one-line format for a subscript set, utilizing information from F.1, F.4 and F.5. Transfer to F.1 \emptyset . Subscript set snaps will occur only in a full trace.
7. If, in the full trace mode, or in auto trace and processing, a new item in the TILT Table (SI 23 = 1), search Compool for item coding and modify conversion routine. Set SI 23 = \emptyset .
8. Get item value from second snap word, converting to BCI if necessary and then convert to octal.
9. Using F.1, F.4, F.7, F.8, set up one-line format for an item set.
- 1 \emptyset . Print one-line snap format DLO; if page is full, print trace headings at top of new page before printing trace information.
11. If this is not the last snap and a full trace, return to process next snap (F.); if an automatic trace, return to match subsequent snaps (E.5).
12. If this is the last snap and a full trace, skip to ILT Print Routine; if an automatic trace and no matches found, log comment and return for next TILT word (E.1). If matches were found, skip a line DLO and return for next TILT word.

ILT Print Routine

This routine makes up a one-line representation of each ILT entry.

A. Read-In

1. Compute a new core address for each of the six Internal Tables by successively subtracting the word count of each Internal Table from 77777. The word counts are found in the MATTBL (+) registers (in permanent core) and the new addresses are saved in these same registers.
2. Using these new addresses, read in the data records of the first six files on tape C-1, checking each ident record. The following returns are provided:
 - a. Ident Error - Log comment, exit to Control Program (JTCPZ).
 - b. Read Parity - Log comment, exit to Control Program (JTCPZ).
 - c. Normal - Proceed.

B. Set up to process ILT Table.

1. The absolute ILT location is used via an index register to locate any one of the five words which comprise an ILT entry.
2. The ILT control word provides a loop control by furnishing the total number of ILT entries.
3. Print ILT output heading DLO.

C. Process ILT Table

For each five-word ILT entry, the following operations are performed.

1. If word 1 of the ILT entry indicates that this ILT entry does not start a new JOVIAL statement, skip to 2. Otherwise, use the subroutine SLT, to search the Statement Label Table. If there is a label for this new statement, store it in Image; if no label, store an asterisk in Image.
2. Convert the ILT entry # to BCD and store in Image.
3. Decode the ILT entry, using the operator in word 1 of the ILT entry as a guide.
 - a. Arithmetic or exponent operator - set SI 1 = 1, store operator in Image. Skip to 3(i).
 - b. Set operator - set SI 2 = 1, store operator in Image. Skip to 3(i).

- c. End or return operator - store operator in Image. Skip to 4.
- d. Stop, GoTo or Term operator - store operator in Image, convert word 5 of ILT entry to a relative decimal address in ILT and store in Image. Skip to 4.
- e. Procedure Declaration, One Output Procedure, or Close Declaration operator - store operator in Image, convert word 2 of ILT entry to a relative decimal address in ILT and store in Image. Skip to 4.
- f. Input Parameter or Output Parameter operator - store operator in Image. Skip to 3(j).
- g. Status Switch Declaration or Numeric Switch operator --
 - (1) Store name of operator in Image.
 - (2) Get number of entries in Switch Table (SWT) from Bits 3-17 of word 2 in the ILT entry. Convert to BCD and store in Image.
 - (3) Get Hollerith label of the switch from words in the ILT entry and store in Image. If a Numeric Switch operator, skip to 4.
 - (4) Guided by the relative position contained in Bits 21-35 of word 4 of the ILT entry, get the Hollerith name of the variable from the VAT Table and store in Image. Skip to 4.
- h. Procedure Call operator - store operator in Image. Pick up the form code, Bits 24-29 of word 1 and store the name of the form in Image. Then pick up the Hollerith label from word 5 and store it in Image. Skip to 4.
- i. Decode left-half term contained in word 1 and word 2 of the ILT entry. The class code in Bits 30-35 of word 1 is examined, and information in word 2 is interpreted according to these possible classes.
 - (1) Constant Class
Bits 21-35 give the relative position of the constant in the CON Table. A subroutine, CONST, is used to pick up the constant, convert it to decimal, and store it in the Image.
 - (2) Subscript Class
Bits 3-17 give the relative position of the subscript. A subroutine, SUBSC, is used to pick up the subscript letter and any associated constant and store them in the Image.

- (3) Subscripted Variable Class
Bits 3-17, the subscript, are decoded in the manner described above. Bits 21-35 give the relative position of the variable in the VAT Table. The variable is picked up and stored in the Image.
- (4) Variable Class
Bits 21-35 are used to pick up the variable as described above.
- (5) Special Variable Class
In this case the form code contained in the first word of the entry, Bits 24-29, are examined. The forms may be any of the following:

ENT
VARIABLE-BIT
VARIABLE-BYTE
VALUE
NWDS
NENT
NWDSSEN

The name of the form is stored in the Image, then the variable is decoded as described above.

The one exception is that if the form is either VARIABLE-BIT or VARIABLE-BYTE, the next ILT entry will contain more information about the variable. The entry is decoded as follows:

The second word, Bits 3-17, contain the relative position of the first bit or byte in the SUB Table. Bits 21-35 give the relative position in the SUB Table of the number of bits or bytes. Both these subscripts are picked up in the manner described for "Subscript Class" (see above) and are stored in the Image.

- (6) Special Subscript Variable Class
This is decoded in the same manner as the above class, except that after the form and variable items are processed, Bits 3-17 are also decoded as described in the "Subscript Class."
- (7) Temporary Storage Class
In this case, the word TEMP is stored in the Image. Bits 21-35 give the relative position in the Temporary Storage Table and are converted and stored in the Image.

- (8) Accumulator Class
In this case, the word ACCUMULATOR is stored in the Image.
- (9) Illegal Class
When the class code does not fall into any of the above categories, the program considers it an illegal code. In this case the word CLASS is stored in Image and the class code is converted to decimal and is stored in the Image.

j. Decode right-half term

The information for this term is contained in words three and four of the ILT entry and, except for that one difference, is processed in the same manner as described for the left-half term.

k. If Input Parameter operator or Output Parameter operator, skip to 4, otherwise decode address. The address output is obtained from the fifth word of the entry. This word is decoded as follows, depending upon sense indicator setting.

- (1) If Sense Indicator #1 or #2 is set, Bits 21-35 represent the relative address, either in temporary storage or of an ILT entry. If the bits represent a temporary storage relative address, the bits are converted to decimal and stored in the Image. If the bits represent an ILT entry, they are divided by five, then converted to decimal and stored in the Image.
- (2) If neither Sense Indicator #1 nor #2 is set, the fifth word contains the relative branch address in ILT for true and false. Bits 3-17 are the true branch address and 18-35, the false branch address. Both these addresses are divided by five, converted to decimal and stored in the Image preceded by the words True and False.

4. Write one-line DLO.

5. If last ILT entry, log comment, exit via Control Program, (JTCPZ), otherwise, return to 1 to process next ILT entry.

Subroutines

A. TRACE or STORE

This subroutine is in the environment of JOLLZ, Interpreter Second Pass, and is used to write transfer and/or set snaps onto tape D-1.

1. If JOLLZ performs a transfer operation and TRAP1 $\neq \emptyset$, the following calling sequence is used:

```
TSX TRACE,4
MZE To where in ILT - 2's Complement,, From where in ILT -
    2's Complement.
```

2. If JOLLZ performs a set operation and TRAP2 $\neq \emptyset$, the following calling sequence is used:

```
n   TSX STORE,4
n+1 PZE Relative Location of Subscript - 2's Complement,,ILT
      present log 2's Complement.
n+2 Value stored into data tables.
n+3 Hollerith name of item ( $\emptyset$  if subscript set)
n+4 MZE or PZE      ,,Value of Subscript
```

3. The subroutine adds either one word or four words from the respective calling sequences to an internal storage table.
4. If the subroutine, after storing the calling sequence, finds that the storage table is full, it writes the contents of the storage onto tape D-1. The subroutine waits until the transfer is completed.
5. End of File

The following calling sequence causes the subroutine to write the contents of the storage table, write an end of file and rewind.

```
TSX TRACE,4
MZE,T, $\emptyset$ 
```

If T = 1 and a parity is discovered, a comment is logged and the routine halts for a change of tape and returns to write an end of file (A.5). This tag is used for the beginning rewind before any words have been written on the tape.

If T = 2 and a parity is discovered, the procedure is the same as below (A.6)

6. Tape Parity or End of Tape

When a parity is discovered, a comment is logged and trace operations are abandoned for the remainder of the test. (TRAP1 and TRAP2 set = \emptyset)

B. WRT

This subroutine, contained in the Helpful Package in permanent core, is used to write on tape for DLO printing. The routine will write from 1 to 120 characters per line, from the beginning location specified in the calling sequence:

m TSX WRT,4
m+1 PZE, MZE, or PTH Location of data, T, # of words

PZE - one line single spaced
MZE - Eject page, reset page and line counters, write heading, write one line single spaced.
PTH - one line double spaced
T = \emptyset - one line, off-line
T = 1 - one line, on-line and off-line
T = 2 - Eject page, reset line count, write heading and write one line. Off-line.

All data written by WRT can be made to appear on-line by depressing SS #3.

C. CONV

This is a general-purpose routine, which is tailored to fit the item being processed. It will convert floating point, fixed integer, mixed fractions and indirect addresses to BCD. The following calling sequence is used:

TSX CONV,4

The Accumulator contains the value to be converted.
The results of the conversions are in OUTPT, OUTPT+1, OUTPT+2.
The functions performed are listed below.

1. Converts AD items to 6 bit octal character format.
2. Converts MX, FI, FP numbers to a two-character decimal characteristic with sign and a twelve-character (or less) decimal fraction with sign.
3. Deletes trailing zeros from the decimal fraction.
4. If MX coded and there is no integer, deletes leading zeros from the fraction.
5. Zero quantities appear as signed zeros.

D. L318

This is a SHARE subroutine which is used to convert from binary to BCI. The following calling sequence is used:

TSX L318,4
PZE Y, D

- Y - is used to specify the lead characters desired. Usually \emptyset or 48 (blank) is used.
D - is used to specify the number of characters desired (up to 12).
The MQ contains the value to be converted.

E. SEARCH

This subroutine searches Baby Compool for an item name to match that contained in the Accumulator. The address of the item will be contained in the Accumulator on the return. If the Accumulator = \emptyset , the item was not found. The following calling sequence is used:

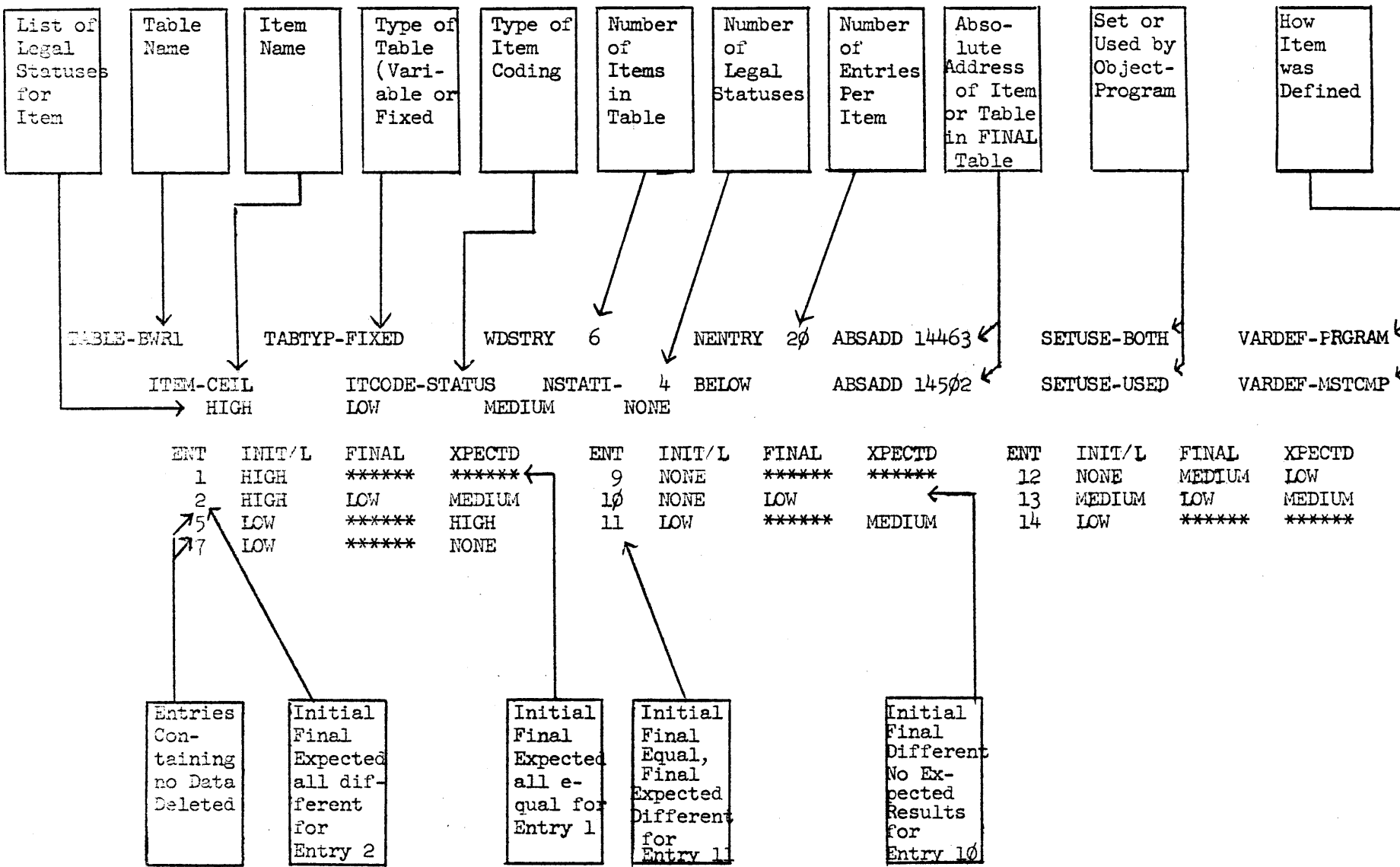
TSX SEARCH,4

OUTPUT

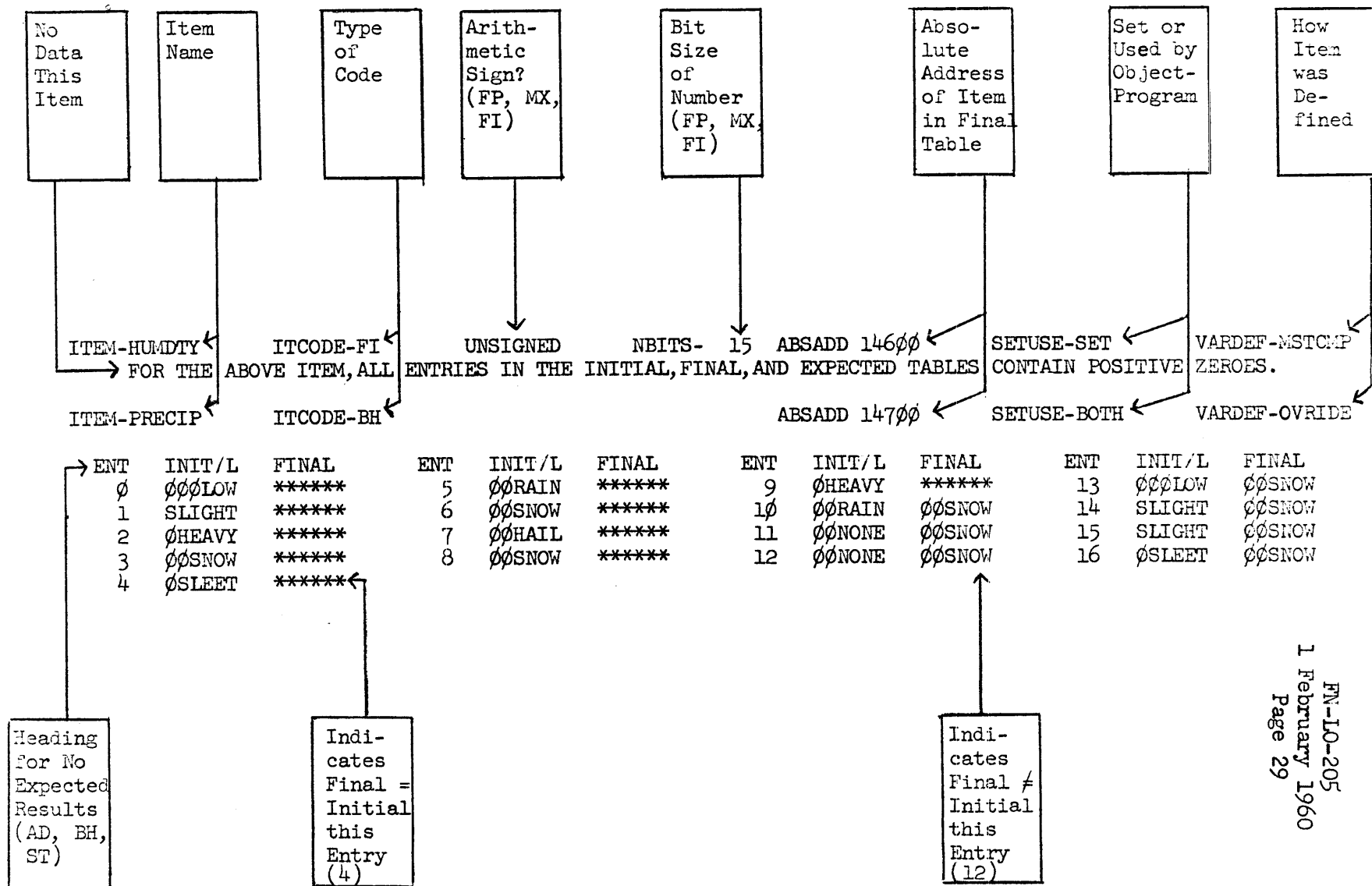
Output Formats

With the exception of the Message Printouts and data headings, all data output is loaded into a twenty word table called IMAGE. Image normally contains BCI blanks and, after each output via WRT, is house kept by loading with BCI blanks.

The following pages contain examples of the various types of output format. The salient features are discussed by means of the arrowed comments.

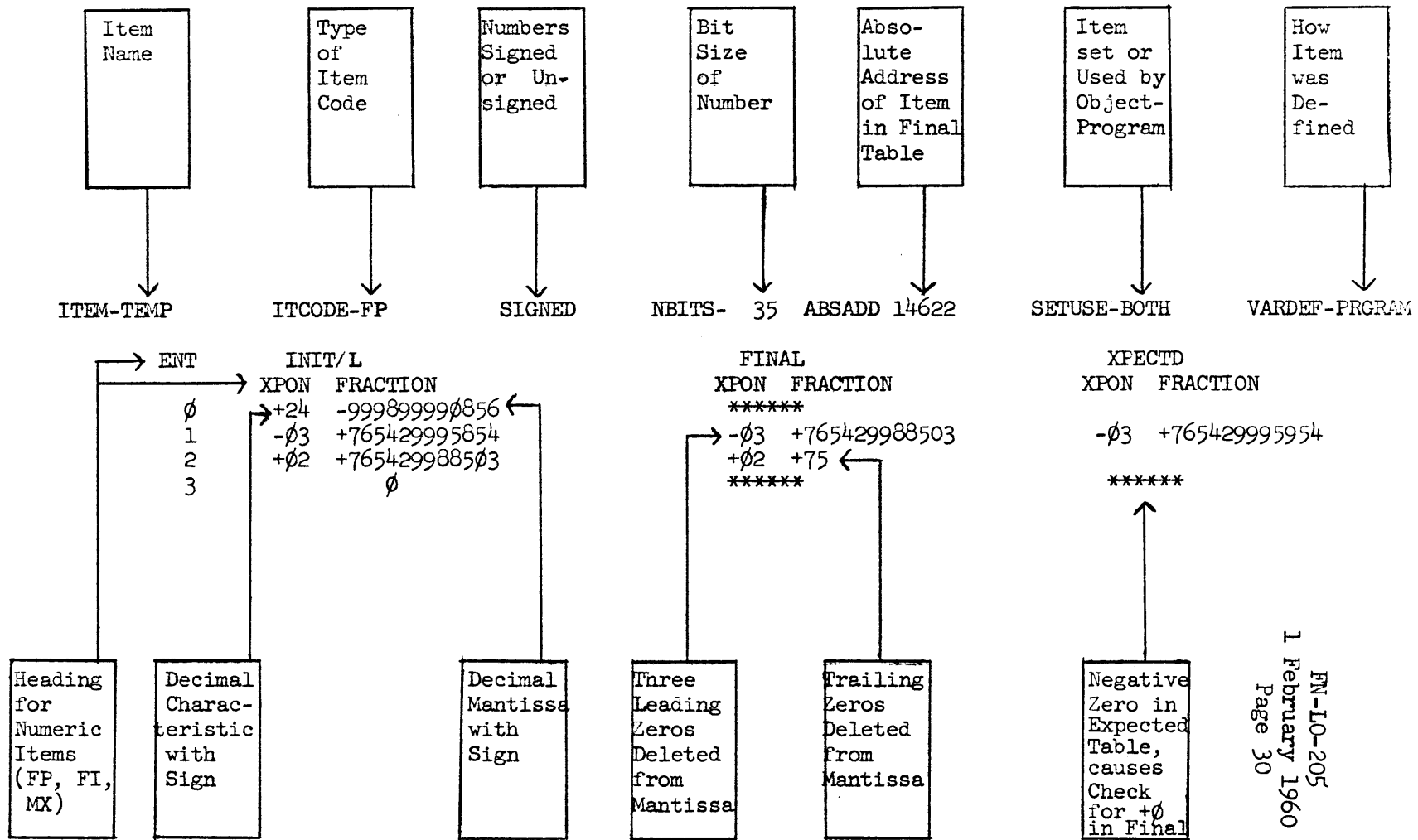


EXAMPLE OF TABLE ENTRY WITH STATUS ITEM AND EXPECTED RESULTS
 FIGURE 1

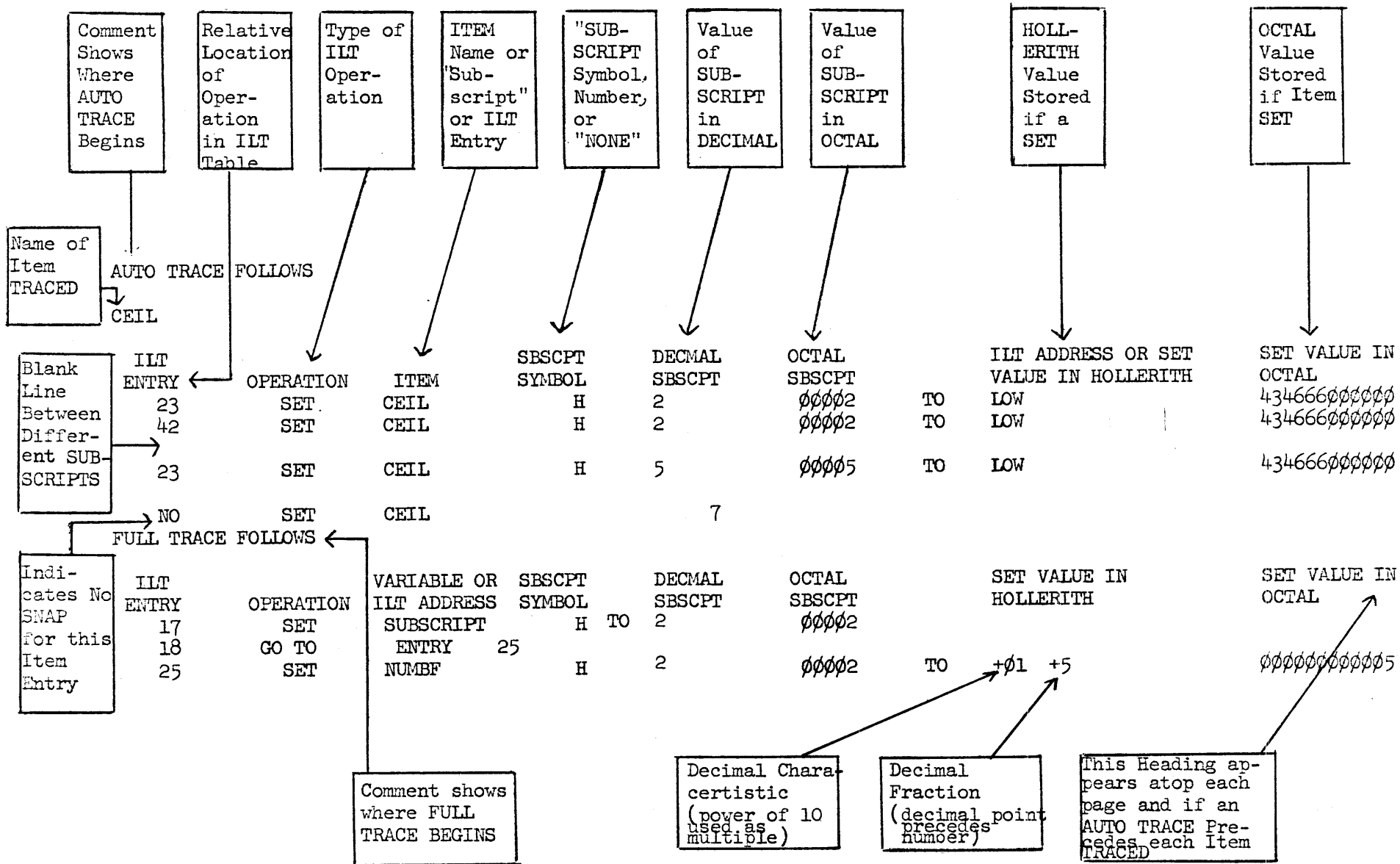


EXAMPLE OF FIXED INTEGER AND BINARY CODED HOLLERITH ITEMS WITH NO EXPECTED RESULTS

FIGURE 2



EXAMPLE OF ARITHMETICALLY-CODED ITEM DATA FORMAT
FIGURE 3



EXAMPLE OF AUTO AND FULL TRACE
FIGURE 4

NEW STATEMENT	ILT ENTRY	LEFT HALF TERM	OPERATION	RIGHT HALF TERM	ADDRESSE (S)
*	∅	SUBSCRIPT X	SET TO	CONST N	
	1	VAR SUB X ±N	PLUS	TEMP N	
*	2		GOTO		ILT NO N
	3	VAR	LS	VAR	TRUE N FALSE N
LABEL	4	VALUE VAR	DIV	VAR SUB X	TEMP N
	5		PROC DECLAR		ILT NO N
	6	N	STAT SW DECL	VAR	LABEL OF SWITCH
*	7	N	NUMR SW DECL		LABEL OF SWITCH
*	8		PROC CALL		LABEL OF PROCEDURE
	9		ONE OUTPT PR		ILT NO N
*	10	SUBSCRIPT X	INPUT PARAM	VAR	
	11	SUBSCRIPT X	OUTPT PARAM	VAR	
	12		END		
	13		TERM		ILT NO N

* = new statement
 X = subscript letter
 N = number of (ILT entry
 (constant
 (temporary storage
 (entries in switch table
 VAR = variable in Hollerith
 LABEL - label of new statement

STRUCTURE OF ILT PRINTOUT
FIGURE 5

MESSAGE PRINTOUTS

TRACE and STORE Subroutine

- A. "UNASSIGNED UNIT FOR D-1. PRESS START TO CONTINUE."

Indicates to operator tape D-1 not available.

- B. "TAPE D-1 WRITE ERROR. CHANGE DRIVE AND PROCEED."

Indicates to operator a parity before any data was written on D-1.

- C. "TAPE D-1 WRITE ERROR. TRACE ABANDONED."

Informs programmer that no trace was performed because of a parity after data was written on tape.

Disassemble Compool and Data Comparison

- A. "UNASSIGNED UNIT FOR C-1. PRESS START TO CONTINUE."

Indicates to operator tape C-1 not available.

- B. "IDENT RECORD FOR BABY COMPOOL INCORRECT. NO DATA REDUCED. CONTROL RETURNED TO TCP."

Informs programmer that lack of Baby Compool forced abandonment of data reduction.

- C. "TAPE PARITY UNABLE TO READ BABY COMPOOL XXXXXX RECORD. NO DATA REDUCED. CONTROL RETURNED TO TCP."

Informs programmer that data reduction was abandoned due to a parity on read-in of Baby Compool ident or data record.

- D. "IDENT RECORD FOR EXPECTED DATA INCORRECT. DATA BYPASSED."

Informs programmer that data reduction occurred without using any expected results.

- E. "IDENT RECORD FOR INITIAL DATA INCORRECT. DATA BYPASSED."

Informs programmer that data reduction occurred without using any initial conditions.

- F. "TAPE PARITY WHEN ATTEMPTING TO READ EXPECTED DATA RECORD. DATA NOT USED IN DATA REDUCTION."

Informs programmer that data reduction occurred without using any expected results.

- G. "TAPE PARITY WHEN ATTEMPTING TO READ INITIAL DATA RECORD. DATA NOT USED IN DATA REDUCTION."

Informs programmer that data reduction occurred without using initial conditions.

- H. "TILT TABLE FULL. AUTO TRACE STOPS HERE."

Informs programmer that automatic trace performed only on those differences appearing prior to this comment.

- I. "THE NUMBER OF DIFFERENCES FOUND ARE XXXXXX."

Informs the programmer (in automatic trace only) of the number of differing expected results.

- J. "COMREG XXXXXX CON XXXXXX SUB XXXXXX VAT XXXXXX SWT XXXXXX TLT XXXXXX SLT XXXXXX TABREG XXXXXX."

The variables contain the number of registers comprising the tables denoted at the left of each variable.

Automatic and Full Trace

- A. "UNASSIGNED UNIT FOR D-1. PRESS START TO CONTINUE."

Informs operator tape D-1 not available.

- B. "SNAP TAPE TOO LARGE, INCOMPLETE TRACE."

Informs programmer that a full trace will be performed, which omits those snaps which are in excess of core capacity.

- C. "TAPE D-1 READ ERROR. INCOMPLETE TRACE."

Informs programmer that a full trace will be performed, omitting those snaps not read in when the parity occurred.

D. "NO SNAP ON D-1 TAPE. NO TRACE PERFORMED."

Self-explanatory.

E. "FULL TRACE FOLLOWS."

Informs programmer JDSYZ operating in full trace mode.

F. "AUTO TRACE FOLLOWS."

Informs programmer JDSYZ operating in automatic trace mode.

ILT Print

A. "EOF MISSING ON C1."

Indicates to programmer lack of EOF between data record and following ident record caused no ILT printout.

B. "PARITY OR EOT ERROR WHEN READING C1."

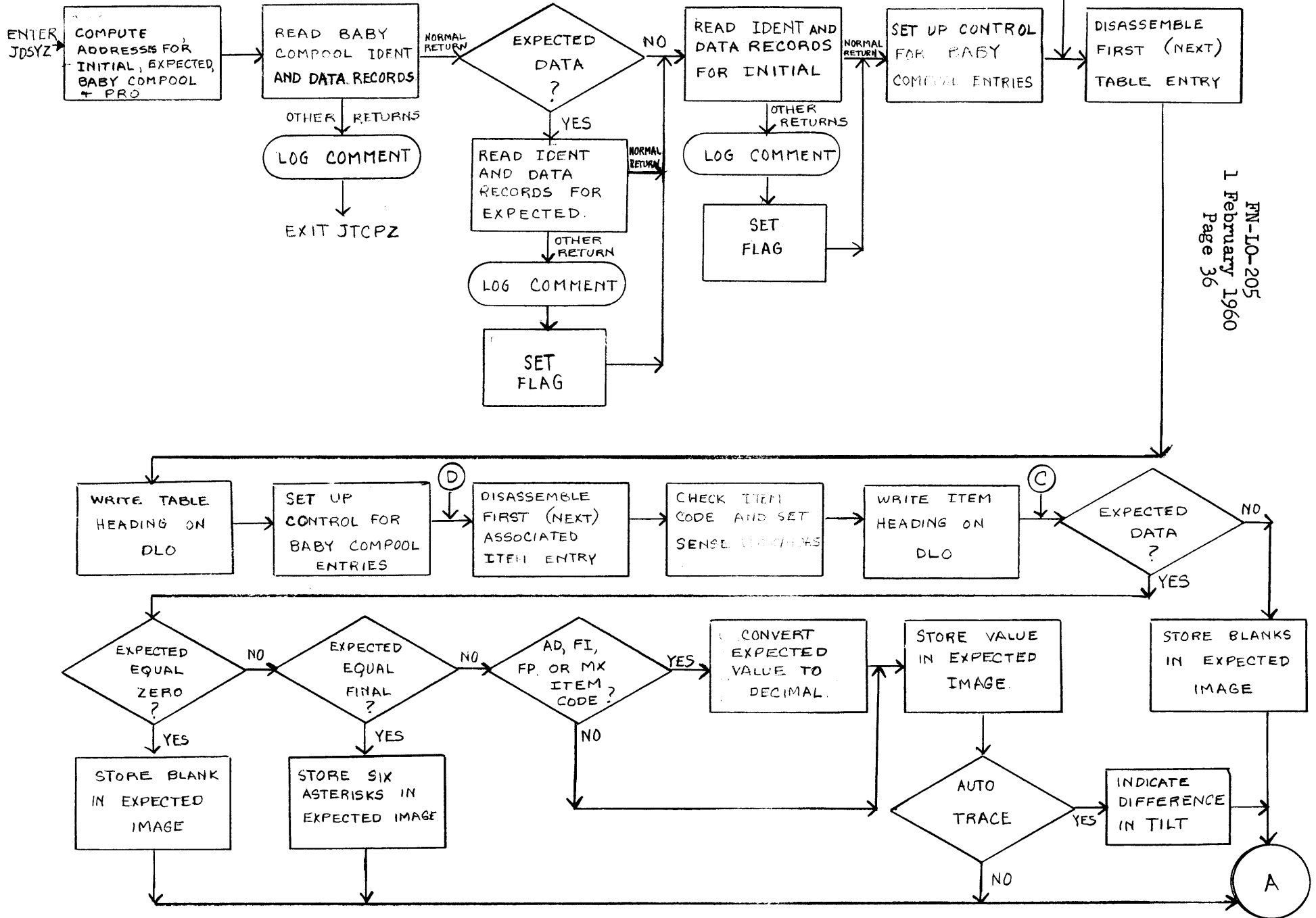
Indicates to programmer that tape error caused ILT print to be abandoned.

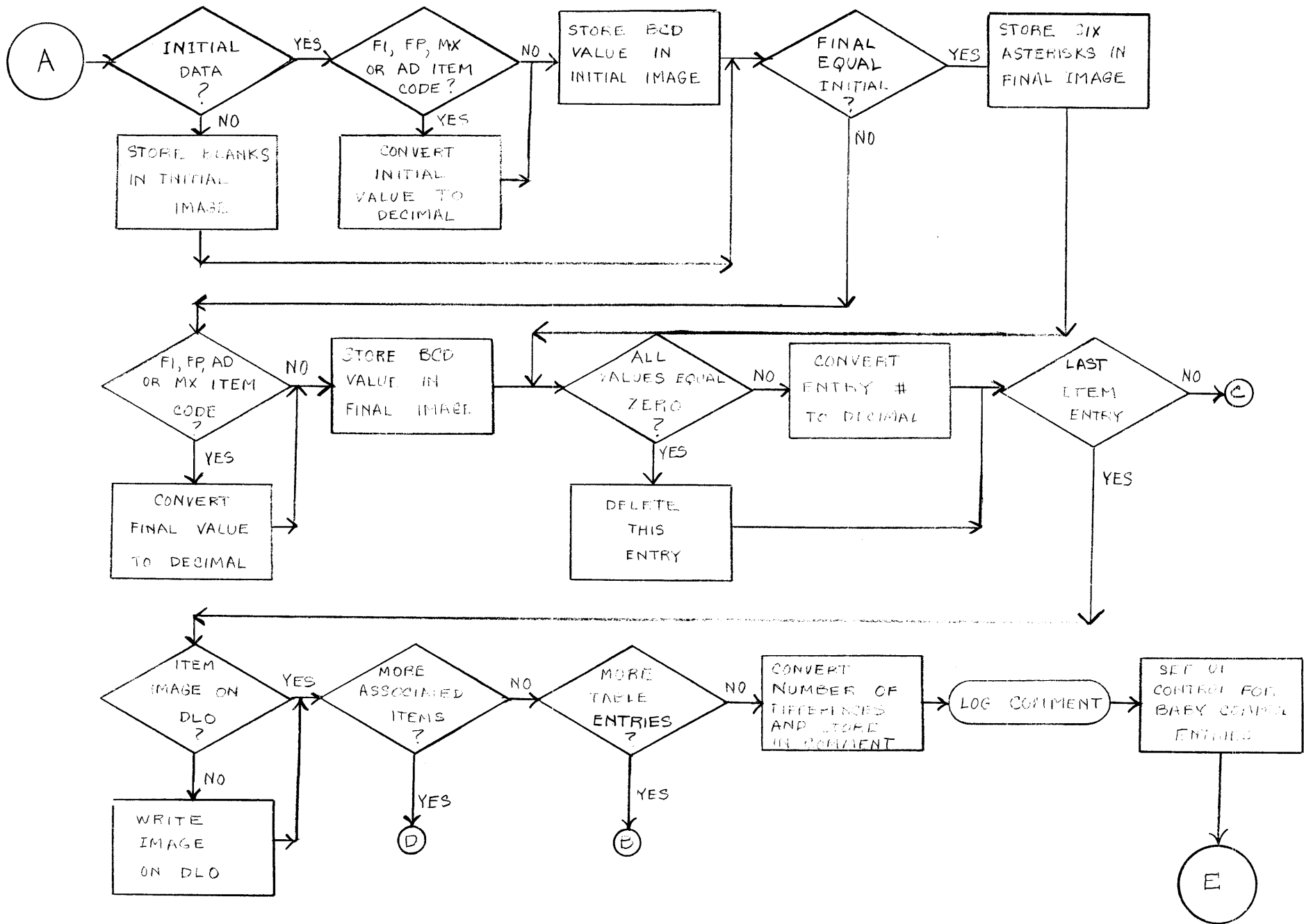
C. "XXXXXX IDENT ERROR." (X's indicate table name.)

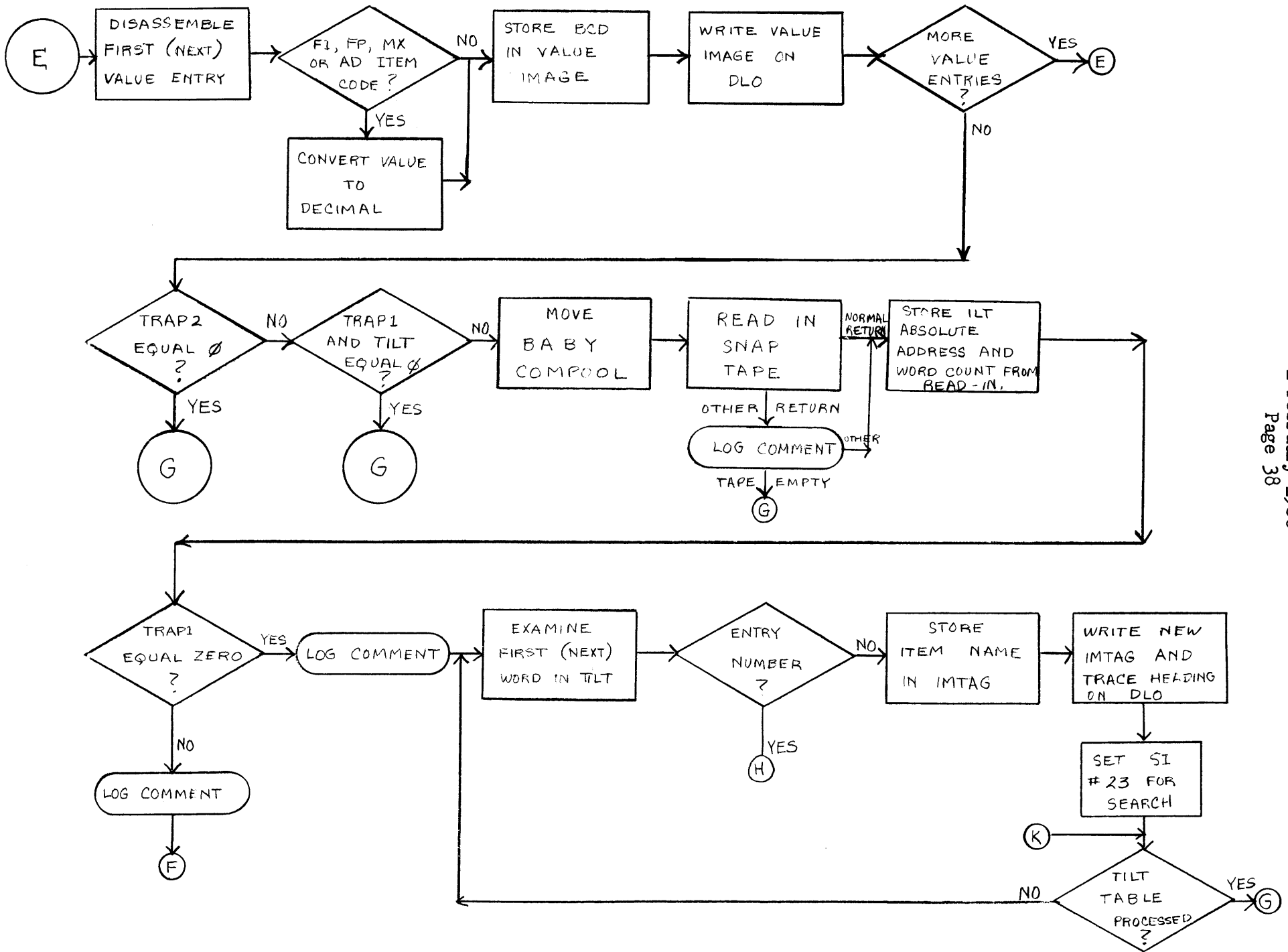
Indicates wrong ident caused abandonment of ILT print.

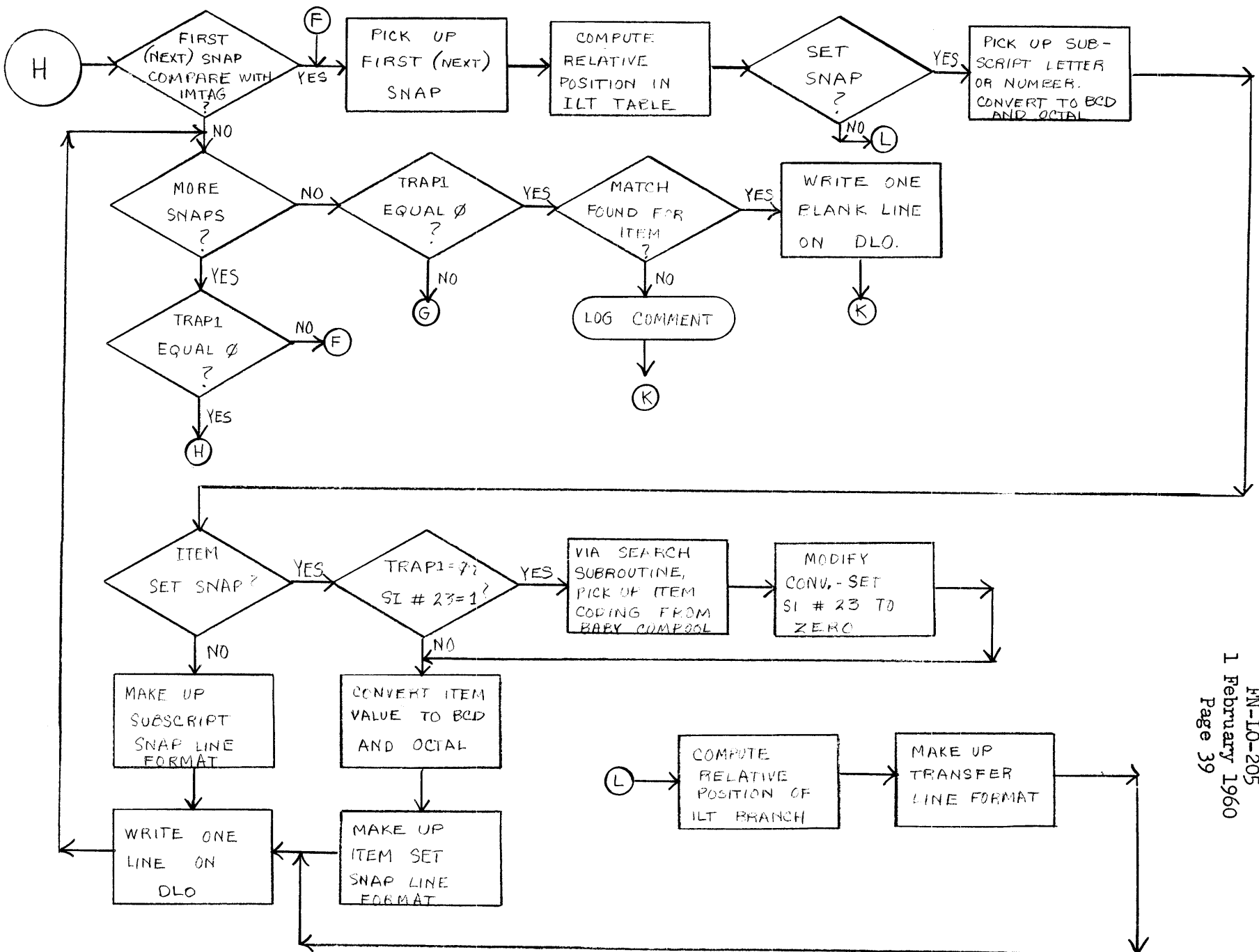
D. "JDSYZ HAS DECODED ILT."

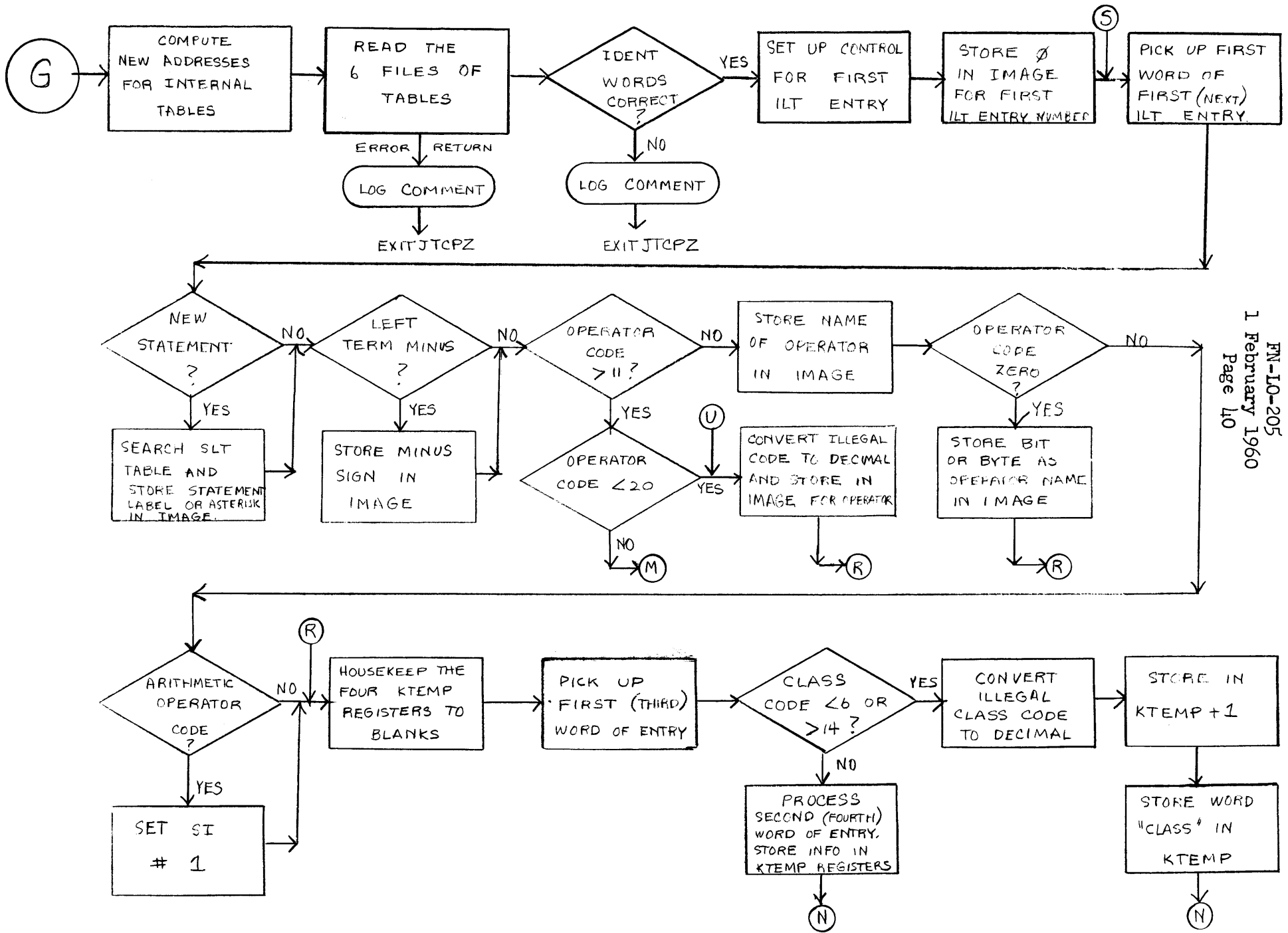
Indicates successful ILT printout.

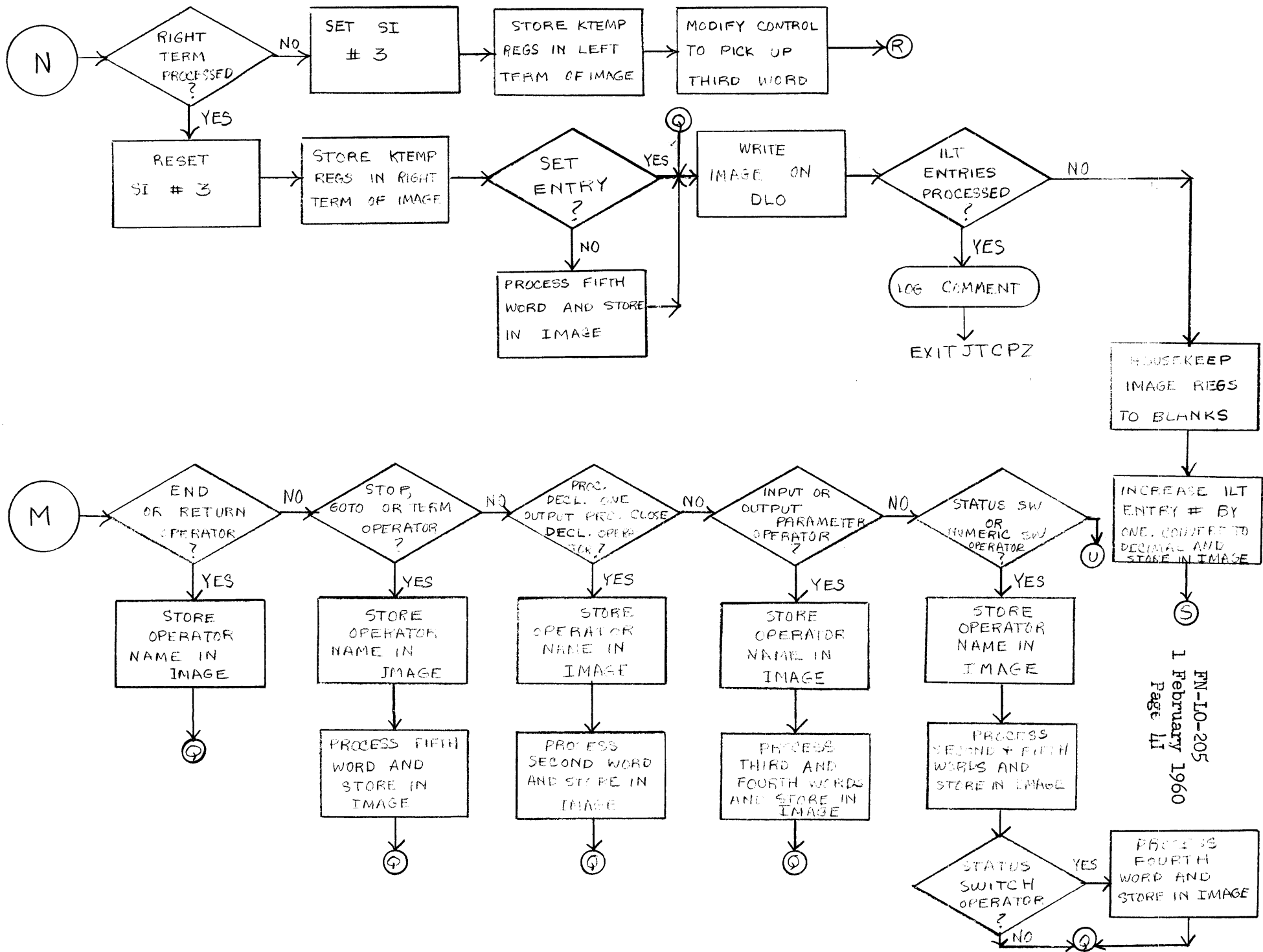












FN-10-205
1 February 1960
Page 11

A listing of the program symbolic deck will be issued as the first supplement to this document (FN-10-205, S-1).

Distribution:

SDC (Lodi)

Division Staff (1 ea.)
Programming Branch Staff (1 ea.)
Program Production Group (1 ea.)
Program Design Group - M. Mineart (20)
Program Requirements Group - F. Diaz (5)
CUSS Project - J. E. Schwartz (10)

SDC (Santa Monica)

IEC

J. D. Madden
R. Bosak
J. Matousek
B. Morriss
G. Dobbs (10)
E. Gordon
C. M. Lawson
D. E. Henley
G. Jacobs

Standard Distribution (35)

:cah