

Application Program

IBM 1401 Decision Logic Translator (1401-SE-05X) Program Reference Manual

Design logic is captured using a FORTRAN-oriented decision table language. The logical statements of this language are the input to the Decision Logic Translator system. After decoding the statements of a table, the system sorts them according to commonalities in rows and columns in order to produce an efficient output program. The sorted rules are then translated into FORTRAN statements. This process is continued table by table until all tables of any single run are translated into FORTRAN statements.

This manual illustrates the use of two IBM forms: "Decision Logic Coding Form" (X20-8024) and "Decision Table" (X28-1630).

Copies of this and other IBM publications can be obtained through IBM branch offices. Address comments concerning the contents of this publication to IBM, Technical Publications Department, 112 East Post Road, White Plains, N. Y. 10601

CONTENTS

General Description of Program	1
Purpose and Objectives	1
Control and Audit Techniques	1
Tape Error Routines	1
Timing	2
Restrictions and Range	2
Program Systems Chart	3
Input/Output Description	4
Input	4
Output	13
Intermediate Data Storage	13
Input/Output Summary	14
Sample Problem	16
Suggested Coding Practices	47
Program Setup	48
Console Operating Instructions	48
Error Codes and Messages List	49
Error and Restart Procedures	53
Storage Map	54

GENERAL DESCRIPTION OF PROGRAM

The Decision Logic Translator reads, lists and validity-checks each condition row. Extended entry rows are reduced to limited entry. The number of "don't care" cells in each row is counted; the rows are sorted so that the rows with the fewest "don't care" cells come first. However, extended entry rows are sorted to the top unless there is only one nonempty cell in the row. Treating each Y as a 1, N as a 2, and . or E as a 3, the columns are sorted on their value. These sorts bring together similar groups of conditions. The condition area is scanned to detect identical groups of conditions within two or more columns. Insofar as possible, only one set of IF statements is generated for each pattern of conditions even though the pattern appears in more than one column. This process minimizes the size of the object program and maximizes its efficiency.

To process the action area the Decision Logic Translator reads, lists and validity-checks each action row. Extended entry rows are reduced to limited entry, and the action area as a whole is validity-checked. Each formula and list is read, edited and stored on tape. The relationship between the sorted and unsorted columns of the condition area is stored within the computer. Each action rule can then be correlated with the proper condition rule without sorting the actions. One or more FORTRAN statements are generated for each nonempty cell in the action area. To conserve space in the FORTRAN object program, complicated formulas and all I/O operations are coded out of line if they appear more than once. The RETURN action is also coded out of line.

When all tables have been processed, the user must supply FORTRAN control cards, correct nonfatal errors, supply FORMAT statements where necessary, and compile the object program.

PURPOSE AND OBJECTIVES

The program was designed to ease the burden placed upon those who are responsible for implementing an Automated Design Engineering system. The Decision Logic Translator substantially reduces the time and cost required to implement such systems by minimizing the amount of programming required to convert from decision tables to FORTRAN program.

CONTROL AND AUDIT TECHNIQUES

The Decision Logic Translator has complete error routines to handle tape read/write errors. These routines are hand-coded to conserve storage. No audit trails are maintained in the strict sense of the word, but counts are kept of the number of records written on tape and the number of tape errors encountered.

TAPE ERROR ROUTINES

For a tape error condition, the tape is backspaced over the error record. The numbers 771 (for tape 1) or 772 (for tape 2) are placed in the A and B registers and the machine halts. Pushing START causes the program to execute the tape command again and

proceed if an error condition does not recur. The end-of-file or end-of-reel condition follows the same procedure except that the A and B registers contain 991 or 992 for tape 1 or tape 2, respectively.

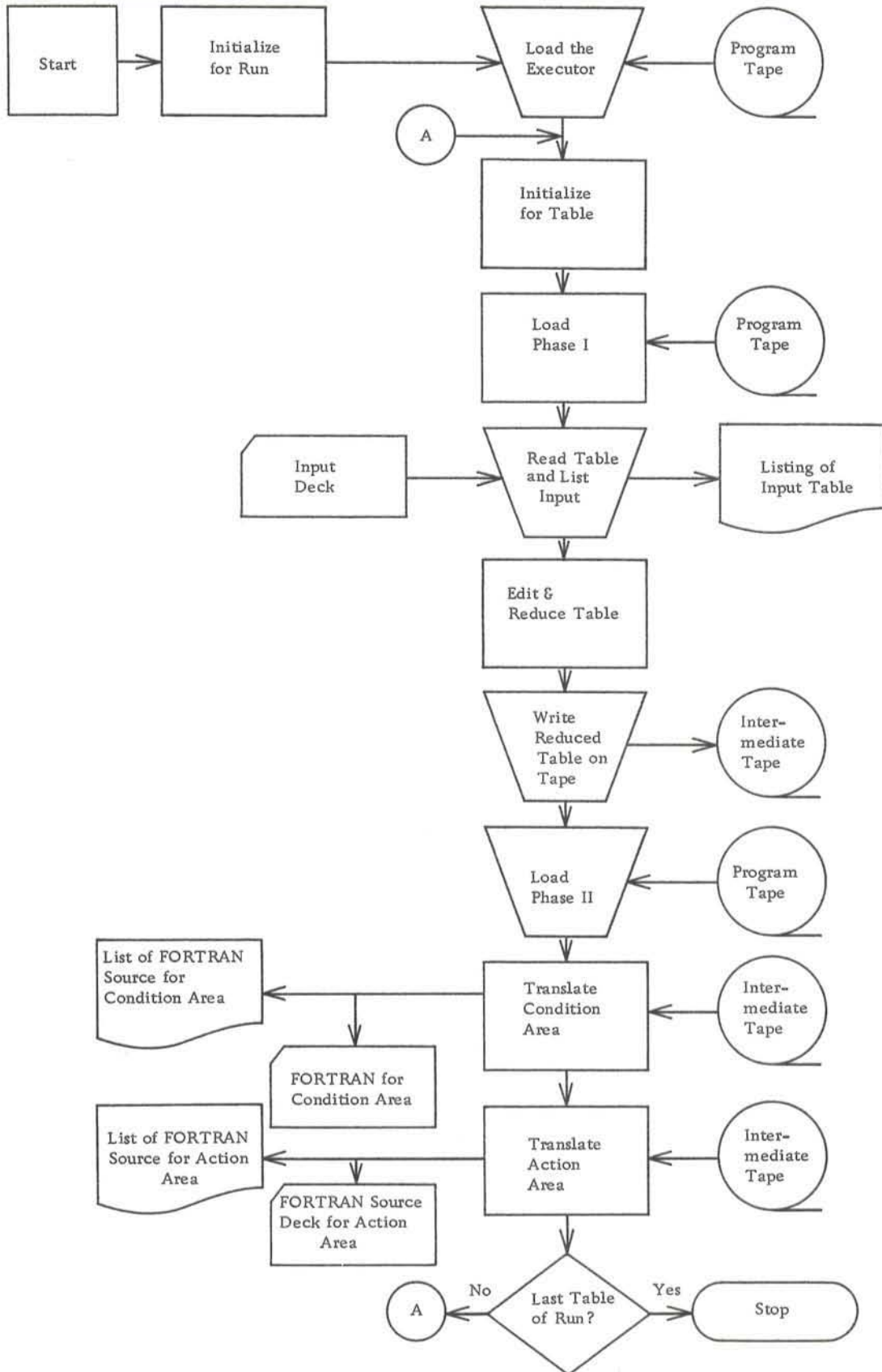
TIMING

After a program load time of 40 seconds, each table takes between five seconds and four minutes to translate. The average time per decision table is one minute. Most errors are detected within the first quarter of the processing time for a table.

RESTRICTIONS AND RANGE

- The number of columns (rule) must be less than or equal to 40.
- After reduction of extended entry rows to limited entry, the number of condition rows and the number of action rows must be less than or equal to 90.
- After reduction to limited entry, the net size of the condition area and the action area must be less than 1,000 cells.
- No more than 50 references to any table may be made by the DO TABLE instruction.
- No more than 100 FORTRAN statement numbers may be required by the object code. If more than 100 numbers are needed, alphabetic symbols have to be used.
- The number of closed tables requested in one run must be less than or equal to 20.

PROGRAM SYSTEMS CHART



INPUT/OUTPUT DESCRIPTION

INPUT

Coding Sheet

Figure 1 shows the coding sheet to be used.

Figure 2 contains examples of source language instructions inserted in a coding sheet. This illustration is not a program but a listing of the various ways in which each command may be coded.

Acceptable Source Language Instructions

Condition area

GE -- Greater than or equal to.

GR -- Greater than.

EQ -- Equal.

NE -- Not equal.

LR -- Less than.

LE -- Less than or equal to.

Action area

SET -- Set operand 1 equal to operand 2.

INCREMENT -- Increment operand 1 by operand 2.

MOVE -- Replace operand 2 by operand 1.

DO FORMULA -- Compute the formula specified by the formula number in operand 2.

DO TABLE -- Perform the logic and computations described in the table specified by operand 2. Then perform the next requested instruction within this table. All closed tables (those tables performed by a DO TABLE command) must always have a RETURN statement to return control to the calling table.

READ -- Equivalent to a FORTRAN read instruction.

PUNCH -- Equivalent to a FORTRAN punch instruction.

IBM

**DECISION LOGIC
CODING FORM**

Printed in U.S.A.
Form X20-8024

SYSTEM _____

ANALYST _____

PAGE _____ OF _____

DATE _____

TABLE NO.	ROW NO.	CARD	STUB		ENTRY																								
					OP1	OPERAND 1				OP2	OPERAND 2				OP3	OPERAND 3				OP4	OPERAND 4				OP5	OPERAND 5			
					01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
001	01		IF ALPHA		LR	3.			EQ	3.			GE	3.															
002	02		IF BETA	EQ					GAMMA				EPSLON																
002	03		IF ZETA	LE -1.0	Y				.				N																
003	01		IF ALPHA	LR 0	Y	Y	E																
003	02		IF ALPHA	EQ 1	.	.	Y	Y	E																
003	03		IF ALPHA	GR 1	Y	Y	.	.	E																
003	04		IF BETA	EQ GAMMA	Y	N	Y	N	Y	N	.	.	E																
004			SET ALPHA	= 3.0					X			.		X															
			SET ALPHA	=					3.0			4.0		5.0															
			INCREMENT ALPHA	BY 1.0					X			.		X															
			INCREMENT ALPHA	BY					1.0			2.0		1.0															
			MOVE 3.5	TO ALPHA					X			.		X															
			MOVE 3.5	TO					ALPHA			BETA		GAMMA															
			DO FORMULA	13					X			.		X															
			DO FORMULA						12			13		14															
			DO TABLE	137					X			.		X															
			DO TABLE						136			137		.															

COMMENTS _____

6

Figure 2. Examples of source language instructions

IBM

**DECISION LOGIC
CODING FORM**

Printed in U.S.A.
Form X20-8024

SYSTEM _____

ANALYST _____

PAGE _____ OF _____

DATE _____

TABLE NO.	ROW NO.	CARD	STUB	ENTRY																							
				OPERAND 1				OPERAND 2				OPERAND 3				OPERAND 4				OPERAND 5				OPERAND 6			
				OP1 01	02	03	04	OP2 05	06	07	08	OP3 09	10	11	12	OP4 13	14	15	16	OP5 17	18	19	20	OP6 21	22	23	24
			READ	LIST 14	X				.			X															
			READ	LIST	14				13			14															
			PUNCH	LIST 3	X				.			X															
			PUNCH	LIST	2				3			2															
			PRINT	LIST 7	X				.			X															
			PRINT	LIST	7				8			9															
			READ INPUT TAPE	5 LIST 2	X				.			X															
			READ INPUT TAPE	5 LIST	2				3			4															
			WRITE OUTPUT TAPE	4 LIST 4	X				.			X															
			WRITE OUTPUT TAPE	4 LIST	4				1			4															
			READ TAPE	2 LIST 3	X				.			X															
			READ TAPE	2 LIST	3				4			5															
			WRITE TAPE	2 LIST 11	X				.			X															
			WRITE TAPE	2 LIST	10				11			12															

COMMENTS _____

Figure 2. (Continued)

IBM

**DECISION LOGIC
CODING FORM**

Printed in U.S.A.
Form X20-8024

SYSTEM _____

ANALYST _____

PAGE _____ OF _____

DATE _____

TABLE NO.	ROW NO.	CARD	STUB	ENTRY																							
				OPERAND 1				OPERAND 2				OPERAND 3				OPERAND 4				OPERAND 5				OPERAND 6			
				OP1 01	02	03	04	OP2 05	06	07	08	OP3 09	10	11	12	13	OP4 14	15	16	17	OP5 18	19	20	OP6 21	22	23	24
			STOP	1	2	3	X			.			X														
			STOP				1	2	3				1	2	4												
			PAUSE	2	4	6	X			.			X														
			PAUSE				2	4	6				4	6	8												
			ERROR	0	0	4	X			.			X														
			ERROR				4						5														
			GO TO TABLE	4			X			.			X														
			GO TO TABLE				4						5														
			RETURN				X			.			X														

COMMENTS _____

8

Figure 2. (Continued)

READ INPUT TAPE -- Equivalent to the FORTRAN statement.

WRITE OUTPUT TAPE -- Equivalent to the FORTRAN statement.

READ TAPE -- Equivalent to the FORTRAN statement.

WRITE TAPE -- Equivalent to the FORTRAN statement.

PAUSE -- Equivalent to the FORTRAN statement.

STOP -- Equivalent to the FORTRAN statement.

ERROR -- This action is taken because of an error in data. Go to the table specified by operand 2.

GO TO TABLE -- Go to table specified by operand 2.

RETURN -- This table was performed via a DO TABLE instruction. Transfer control back to the instruction after the DO TABLE instruction in the calling table.

Operand 1 is the first variable after a command, operand 2 is the second variable(s) after a command, operator (OP) is the action to be performed between operands 1 and 2 in a condition row. For example,

SET ALPHA = BETA

Operand 1 is ALPHA
Operand 2 is BETA

IF ALPHA LE BETA

Operand 1 is ALPHA
Operator (OP) is LE
Operand 2 (OPERAND) is BETA

Keywords

All keywords (condition operators and action codes) are compulsory. They must be spelled the standard way except that embedded blanks may be inserted or deleted. (GOTOTABLE is as acceptable as GO TO TABLE.)

Limited entry condition options

Y -- Yes.

N -- No.

. or E -- "don't care" or "else".

"Don't care" means it does not matter whether this test is true or false (that is, a "don't care" is the same as having both yes and no for this test). "Else" means that all conditions other than those which were tested for in preceding rules will apply to this rule.

Limited entry action options

. -- Don't do this action for this rule.

X -- Do this action for this rule.

Use of the Coding Sheet

Table number--cc. 1-3

- Must be numeric and less than 999. Tables need not be supplied in table number sequence.
- No two tables may have the same number in any given run.

Row number--cc. 4-5

- Must be numeric.
- Rows must be in ascending sequence of row number.
- Row numbers need not be consecutive.

Card code--cc. 6

- This field is used to number the continuation cards of a row that extends over more than one card. The maximum number of cards per row is ten.
- This field is also used to indicate the type of row:
 - I -- Indicates the run identification card.
 - C -- Indicates a comments card.
 - A -- Indicates the first action row.
 - F -- Indicates a formula row.
 - L -- Indicates a list row.
- Formulas and lists may be in any order but must follow the last action row.
- A comments card may appear anywhere except between two cards of the same row.
- A run identification card has I in cc. 6, the run I. D. character in cc. 7, and comments in cc. 8-80.

Stubs and entries

- The table must be complete. Empty cells are not allowed.
- Coding need not be left-justified within the allotted space.
- Stubs appear in cc. 7-32.
- Entries appear in cc. 33-80.

Condition rows

- The left half of the stub must contain IF followed by a variable name.
- If the right half of the stub contains blanks, the row is an extended entry row, type 1. If it contains an operator, it is an extended entry row, type 2. If it contains an operator plus an operand, it is a limited entry row.
- If a table contains a mixture of limited and extended entry rows or only extended entry rows, each row must be spaced in long format (one rule in each eight card columns starting in column 33). If the table contains only limited entry rows, it may be in long or short form (one rule in each two columns starting in column 33).

Action rows

- For arithmetic actions (SET, MOVE, INCREMENT), the left half of the stub must contain an operand.
- All keywords necessary to complete the action operator must be present. If any nonblank fields exist in the right-hand half of the stub after keywords are removed, the row is assumed to be limited entry. The right-hand half of the stub (minus keywords) becomes the second operand.

Formulas and lists

- Formulas must be written as legitimate FORTRAN arithmetic statements. Lists must be written as legitimate FORTRAN lists. Figure 3 shows some samples.
- The words formula and list must start in column 7 and must not contain any embedded blanks. They must be followed by a one- or two-digit number.
- If two or more formulas (or lists) have the same formula (or list) number, only the last one is processed. However, one formula and one list may both use the same number and be correctly processed.

Variables Names

Variable names must be valid FORTRAN names, that is, of six or fewer characters, the first character being alphabetic and indicating the mode (fixed-point or floating-point).

Names of the form III_{nnn} and JJJ_{nnn}, where _{nnn} is a three-digit number, are forbidden. The program will not detect variable names which do not meet the FORTRAN specifications.

Constants

All constants must follow FORTRAN rules. The E format for floating-point constants is allowed. If the left-hand operand is a floating-point name, the program will insert a decimal point in the right-hand operand if necessary.

Restrictions on Table Size and Constants

- The number of columns must be ≤ 40 .
- After reduction of extended entry rows to limited entry, the number of condition rows and the number of action rows must be ≤ 99 .
- After reduction to limited entry, the net size of the condition area and the action area must be $< 1,000$ cells.
- No more than 50 references to any table may be made by the DO TABLE instruction.
- No more than 100 FORTRAN statement numbers may be required by the object code. If more than 100 numbers are needed, alphabetic symbols must be used.

Logic Requirements

If the RETURN operator occurs in the action area, the table is closed. In a closed table, a GO TO TABLE action is valid only if operand 2 is the current table number.

OUTPUT

1403 Printer

- An 80-80 listing of the input table.
- A listing of the FORTRAN II source program.
- A list of diagnostics, where pertinent.

1402 Punch

- The generated FORTRAN II source program deck.

INTERMEDIATE DATA STORAGE

Tape 1

- A record of each called closed table and the number of times it has been called.

Tape 2

- The reduced input table.
- The formula and list input.

INPUT/OUTPUT SUMMARY

Input

Input	Program Name for Input	Source of Input	Input Format
Source table	(Card read in area)	Keypunch	80-column cards
Reduced condition area	STUB (stub list)	Tape 2--phase I	18 x N character record N = no. of condition rows
	AMTRX (condition array)	Tape 2--phase I	N x M character record M = no. of columns
Reduced action area	STUB (action stub)	Tape 2--phase I	18 x L character record L = no. of action rows
	AMTRX (action array)	Tape 2--phase I	L x M character record M = no. of columns
Formulas and lists	FLAREA	Tape 2--phase II	67 x K character records K = no. of FORTRAN card images in formula or list
DO TABLE references	DOREC	Tape 1--phase II	8-character records

Output

Output	Program Name for Output	Disposition of Output	Output Format
Reduced condition area	STUB	Tape 2--phase II	18 x N character record N = no. of condition rows
	MATRXA	Tape 2--phase II	N x M character record M = no. of columns
Reduced action area	STUB	Tape 2--phase II	18 x L character record L = no. of action rows
	MATRXA	Tape 2--phase II	L x M character record M = no. of columns
Source listing	(List of input)	Printer	80-80 list

Output (Continued)

Output	Program Name for Output	Disposition of Output	Output Format
FORTRAN II source listing		Printer	80-character list
FORTRAN II source deck		Punch	80-column punch
Formulas and lists	FLAREA	Tape 2--phase II	67 x K character records K = no. of FORTRAN cards required for formulas and list
DO TABLE references	DOREL	Tape 1--phase II	8-character records

SAMPLE PROBLEM

The problem employed for illustration here involves a set of 15 tables used to design the low-voltage winding of a transformer. The first (input) and last (output) tables in Figure 5 (the translator coding) were added merely to illustrate the use of the input/output statements in the coding language. These two tables read all necessary input data and print computed values.

The original decision tables appear in Figure 4, their coding for input to the translator appears in Figure 5, and Figure 6 contains the printed output from the run.

The I.D. for the run, the first card of the input deck, is A, which was punched in cc. 7. This character is punched in cc. 73 of every output card from this run, and thus helps distinguish output decks from different runs.

The first thing that appears during the run is a listing of the input, one table at a time. All input is listed with the exception of formulas and lists. These are printed only in the generated FORTRAN program. After the first table has been read and listed, it is checked for errors, and if none can be found the two matrices, condition and action, are reduced and stored on tape. The translator then generates the FORTRAN program for the sorted condition area, which will consist entirely of IF statements.

The action area coding then takes place rule by rule. A comments card is printed and punched before the coding of each rule begins to indicate which rule's coding is immediately to follow. All out-of-line coding (of formulas and input/output statements) is indicated as being out of line by a comments card immediately preceding the out-of-line statement.

The translator continues to generate FORTRAN source programs one table at a time until every table in a run has been translated. When the run is completed, the source deck needs only to have the FORMAT information added to the FORMAT card before it can be compiled.

The user should be careful to use good scratch tapes and clean tape drives for the runs because of the heavy use of the tapes by the translator system.

The translator coding (Figure 5) of the decision tables (Figure 4) was straight line for line coding.

This sample problem is intended for use not only as a demonstration but, more important, as an educational tool. It would benefit the user to code from the decision tables (Figure 4) onto the translator coding form, checking each table as coded with its coding in Figure 5, before using the translator for his own tables. After completely coding all tables of the sample problem, the user should have enough familiarity with the coding language to begin writing his tables directly onto the coding form.

IBMSYSTEM _____
Page 3 of 13**DECISION TABLE**Analyst _____
Date _____Form X2B-1630
Printed in U.S.A.

TABLE 20		1	2
ISECON =			
DO FORMULA		1	2
DO FORMULA 3		X	X
DO FORMULA 4		X	X
DO FORMULA 5		X	X
DO FORMULA 6		X	X
GO TO TABLE 21		X	X
FORMULA 1	$SECVC = SECVL$		
FORMULA 2	$SECVC = \frac{SECVL}{\sqrt{PHASE}}$		
FORMULA 3	$XNS = \frac{SECVC}{EPERN}$		
FORMULA 4	$WLVMAX = HCOIL \left(\frac{SLAY}{XNS + SLAY} \right)$		
FORMULA 5	$XIS = \frac{KVA (1000)}{SECVC (PHASE)}$		
FORMULA 6	$WLVMIN = .95 (WLVMAX)$		

Figure 4. (Continued)

IBMSYSTEM _____
Page 5 of 13**DECISION TABLE**Analyst _____
Date _____Form X28-1630
Printed in U. S. A.

TABLE 22	≤4	=6	=8	=10	ELSE
SNSTRW					
TEMP ≤ WLVMAX	Y	Y	Y	Y	ELSE
TEMP > WLVMIN	Y	Y	Y	Y	ELSE
SL ≤ .660	Y	Y	Y	Y	ELSE
SL ≥ .070	Y	Y	Y	Y	ELSE
ADD 1 TO SNSTRW	X
DO FORMULA 1	X	X	X	X	.
DO FORMULA 2	X	X	X	X	.
GO TO TABLE	24	24	24	24	23
FORMULA 1 $AS = \frac{XIS}{SSEC}$					
FORMULA 2 $TSEC = \frac{AS}{SL(SNSTRW)}$					

Figure 4. (Continued)

IBMSYSTEM _____
Page 11 of 13**DECISION TABLE**Analyst _____
Date _____Form X28-1630
Printed in U. S. A.

TABLE 28		Y	N
SK < .078		Y	N
DO FORMULA 1		X	.
CORR = .00089		.	X
DO FORMULA 2		X	X
DO FORMULA 3		X	X
DO FORMULA 4		X	X
DO FORMULA 5		X	X
DO FORMULA 6		X	X
DO FORMULA 7		X	X
DO FORMULA 8		X	X
DO FORMULA 9		X	X
DO FORMULA 10		X	X
DO FORMULA 11		X	X
GO TO TABLE 29			
FORMULA 11	$SLOSSB = .3(XIS)$		
FORMULA 1	$CORR = .21(SK)^2$		
FORMULA 2	$AACT = (SL(SK) - CORR)(SNSTRW)(SNSTRT)$		
FORMULA 3	$SACT = XIS/AACT$		
FORMULA 4	$WTS = XLS(AACT)(.322)(PHASE)(XNS)$		
FORMULA 5	$SLOSSD = 2.54(WTS)(SACT)^2(10)^{-6}$		
FORMULA 6	$SLOSSS = SLOSSD(CYCLE/60.0)$		
FORMULA 7	$SLOSSC = SLOSSD(1 + SLOSSS/100)$		
FORMULA 8	$SRAS = 4.0(PHASE)(SL-6)(SL)(SNSTRW)(XNS/SLAY)$		
FORMULA 9	$SWATIN = SLOSSC/SRAS$		
FORMULA 10	$STEMPR = 1.3512(SWATIN)$		

Figure 4. (Continued)

IBM

**DECISION LOGIC
CODING FORM**

Printed in U.S.A.
Form X20-8024

SYSTEM _____

ANALYST _____

PAGE 2 OF 8

DATE _____

TABLE NO.	ROW NO.	CARD	STUB	ENTRY																									
				OP1	OPERAND 1				OP2	OPERAND 2				OP3	OPERAND 3				OP4	OPERAND 4				OP5	OPERAND 5				OP6
				O1	O2	O3	O4	O5	O6	O7	O8	O9	O10	O11	O12	O13	O14	O15	O16	O17	O18	O19	O20	O21	O22	O23	O24		
19	1		IF SECVL LE 600	Y	N																								
19	2		IF SECVL LE 2400	.	Y																								
19	3	A	SET SLAY = 2	X	.																								
19	4		SET SLAY = 4	.	X																								
19	5		GO TO TABLE 20	X	X																								
20	1		IF ISECON EQ	1				2																					
20	2		DO FORMULA	1				2																					
20	3		DO FORMULA	3				3																					
20	4		DO FORMULA	4				4																					
20	5		DO FORMULA	5				5																					
20	6		DO FORMULA	6				6																					
20	7		GO TO TABLE 21	X				X																					
20			F FORMULA 1 SECVC=SECVL																										
20			F FORMULA 2 SECVC=SECVL/SQRTF(PHASE)																										
20			F FORMULA 3 XNS=SECVC/EPERN																										
20			F FORMULA 4 WLVMAX=HCOIL*SLAY/(XNS+SLAY)																										
20			F FORMULA 5 XIS=XKVA*1000.0/(SECVC*PHASE)																										
20			F FORMULA 6 WLVMIN=.95*WLVMAX																										

COMMENTS _____

Figure 5. (Continued)

IBM

**DECISION LOGIC
CODING FORM**

Printed in U.S.A.
Form X20-8024

SYSTEM _____

ANALYST _____

PAGE 3 OF 8

DATE _____

TABLE NO.	ROW NO.	CARD	STUB	ENTRY																									
				OP1	OPERAND 1				OP2	OPERAND 2				OP3	OPERAND 3				OP4	OPERAND 4				OP5	OPERAND 5				OP6
				01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
21	1	A	SET ITERAZ = 1	X																									
21	2		SET SNSTRW = 1	X																									
21	3		DO FORMULA 1	X																									
21	4		DO FORMULA 2	X																									
21	5		GO TO TABLE 22	X																									
21			FFORMULA1 SL=WLVMAX/SNSTRW*.003																										
21			FFORMULA2 TEMP=SNSTRW*(SL+.003)																										
22	1		IF SNSTRW LE 4.0 EQ 6.0 EQ 8.0 EQ 10.0 E																										
22	2		IF TEMP LE WLVMAX Y Y Y Y E	Y				Y				Y				Y											E		
22	3		IF TEMP GR WLVMIN Y Y Y Y E	Y				Y				Y				Y											E		
22	4		IF SL LE .660 Y Y Y Y E	Y				Y				Y				Y											E		
22	5		IF SL SE .070 Y Y Y Y E	Y				Y				Y				Y											E		
22	6	A	INCREMENT SNSTRW BY 1											X		
22	7		DO FORMULA 1	X				X				X				X											.		
22	8		DO FORMULA 2	X				X				X				X											.		
22	9		GO TO TABLE 24 24 24 24 23	24				24				24				24											23		
22			FFORMULA1 AS=XIS/SSEC																										
22			FFORMULA2 TSEC=AS/(SL*SNSTRW)																										

COMMENTS _____

Figure 5. (Continued)

000001A SAMPLE PROBLEM			PAGE	1
TABLE 010 INPUT				
01001 READ	LIST1	X		
01002 READ	LIST2	X		
01003 READ	LIST3	X		
01004 GO TO TABLE	11	X		

			PAGE	2
C	TABLE 010 CONDITION AREA		A0100000	
01000 CONTINUE			A0100001	
C	RULE 1		A0100002	
01003 READ 01004,	XKVA,SECVL,I SECON		A0100003	
01004 FORMAT			A0100004	
01005 READ 01005,	W,EPLRN,CORENO		A0100005	
01005 FORMAT			A0100006	
01006 READ 01006,	PHASE,LYCLE		A0100007	
01006 FORMAT			A0100008	
GO TO 01100			A0100009	

TABLE 011 INPUT			PAGE	3	
01101 IF XKVA	GR	500	750	1000	1500
01102 IF XKVA	LE	750	1000	1500	2000
01103 SET C	#	.5	.5	.625	.625
01104 SET D	#	.5	.5	.625	.625
01105 DO FORMULA		1	2	2	2
01106 SET E	# .625	X	X	X	X
01107 SET A	# .125	X	X	X	X
01108 SET B	# .1875	X	X	X	X
01109 SET SSEC	#	2650	2550	2350	2200
01110 GO TO TABLE	19	X	X	X	X

Figure 6.

			PAGE	4
C	TABLE 011 CONDITION AREA			A0110010
01100	CONTINUE			A0110011
01103	IFXXKVA	-500. □01104,01104,01105		A0110012
01105	IFXXKVA	-750. □01111,01111,01104		A0110013
01104	IFXXKVA	-750. □01106,01106,01107		A0110014
01107	IFXXKVA	-1000. □01112,01112,01106		A0110015
01106	IFXXKVA	-1000. □01108,01108,01109		A0110016
01109	IFXXKVA	-1500. □01113,01113,01108		A0110017
01108	IFXXKVA	-1500. □01101,01101,01110		A0110018
01110	IFXXKVA	-2000. □01114,01114,01101		A0110019
C			RULE 1	A0110020
01111	C	#.5		A0110021
	D	#.5		A0110022
	HCCIL#W	-6.5		A0110023
	E	#.625		A0110024
	A	#.125		A0110025
	B	#.1875		A0110026
	SSEC	#2650.		A0110027
	GO TO	01900		A0110028
C			RULE 2	A0110029
01112	C	#.5		A0110030
	D	#.5		A0110031
	HCCIL#W	-7.0		A0110032
	E	#.625		A0110033
	A	#.125		A0110034
	B	#.1875		A0110035
	SSEC	#2550.		A0110036
	GO TO	01900		A0110037
C			RULE 3	A0110038
01113	C	#.625		A0110039
	D	#.625		A0110040
	HCCIL#W	-7.0		A0110041
	E	#.625		A0110042
	A	#.125		A0110043
	B	#.1875		A0110044
	SSEC	#2350.		A0110045
	GO TO	01900		A0110046
C			RULE 4	A0110047
01114	C	#.625		A0110048
	D	#.625		A0110049
	HCCIL#W	-7.0		A0110050
	E	#.625		A0110051
	A	#.125		A0110052
	B	#.1875		A0110053
	SSEC	#2200.		A0110054
	GO TO	01900		A0110055
C			OUT-OF-LINE	A0110056
01101	STOP	011		A0110057

TABLE 019 INPUT			PAGE	5
01901	IF SECVL	LE 600 Y N		
01902	IF SECVL	LE 2400 . Y		
01903	ASET SLAY	# 2 X .		
01904	SET SLAY	# 4 . X		
01905	GO TO TABLE	20 X X		

Figure 6. (Continued)

			PAGE	6
C	TABLE 019 CONDITION AREA			A0190058
01900	CONTINUE			A0190059
01903	IF%SECVL -600.	□01905,01905,01904		A0190060
01904	IF%SECVL -2400.	□01906,01906,01901		A0190061
C			RULE 1	A0190062
01905	SLAY #2.			A0190063
	GO TO 02000			A0190064
C			RULE 2	A0190065
01906	SLAY #4.			A0190066
	GO TO 02000			A0190067
C			OUT-OF-LINE	A0190068
01901	STOP 019			A0190069

TABLE 020 INPUT				PAGE	7
02001	IF ISECON	EQ	1	2	
02002	ADD FORMULA		1	2	
02003	DC FORMULA		3	3	
02004	DO FORMULA		4	4	
02005	DD FORMULA		5	5	
02006	DC FORMULA		6	6	
02007	GO TO TABLE	21	X	X	

			PAGE	8
C	TABLE 020 CONDITION AREA			A0200070
02000	CONTINUE			A0200071
02003	IF%ISECON-1	□02004,02005,02004		A0200072
02004	IF%ISECON-2	□02001,02006,C2001		A0200073
C			RULE 1	A0200074
02005	SECVL#SECVL			A0200075
	XNS#SECVL/EPERN			A0200076
	WLVMAX#HCOIL&SLAY/*XNS&SLAY□			A0200077
	XIS#XKVA*1000.0/%SECVL&PHASE□			A0200078
	WLVMIN#0.95&WLVMAX			A0200079
	GO TO 02100			A0200080
C			RULE 2	A0200081
02006	SECVL#SECVL/SQRTF%PHASE□			A0200082
	XNS#SECVL/EPERN			A0200083
	WLVMAX#HCOIL&SLAY/*XNS&SLAY□			A0200084
	XIS#XKVA*1000.0/%SECVL&PHASE□			A0200085
	WLVMIN#0.95&WLVMAX			A0200086
	GO TO 02100			A0200087
C			OUT-OF-LINE	A0200088
02001	STOP 020			A0200089

TABLE 021 INPUT			PAGE	9
02101	ASET ITERA2	# 1	X	
02102	SET SNSTRW	# 1	X	
02103	DO FORMULA	1	X	
02104	DC FORMULA	2	X	
02105	GO TO TABLE	22	X	

Figure 6. (Continued)

		PAGE	10
C	TABLE 021 CONDITION AREA		A0210090
02100	CONTINUE		A0210091
C	RULE 1		A0210092
02103	ITERA2#1		A0210093
	SNSTRW#1.		A0210094
	SL#WLVMAX/SNSTRW-.003		A0210095
	TEMP#SNSTRW*%SL6.003		A0210096
	GO TO 02200		A0210097

TABLE 022 INPUT		PAGE	11			
02201	IF SNSTRW	LE4.0	EQ6.0	EQ8.0	EQ10.0	E
02202	IF TEMP	LE WLVMAX Y	Y	Y	Y	E
02203	IF TEMP	GR WLVMIN Y	Y	Y	Y	E
02204	IF SL	LE .660 Y	Y	Y	Y	E
02205	IF SL	GE .070 Y	Y	Y	Y	E
02206A	INCREMENT SNSTRW BY 1	X
02207	DO FORMULA	1	X	X	X	.
02208	DO FORMULA	2	X	X	X	.
02209	GO TO TABLE	24	24	24	24	23

		PAGE	12
C	TABLE 022 CONDITION AREA		A0220098
02200	CONTINUE		A0220099
02203	IF%TEMP -WLVMAX	02205,02205,02204	A0220100
02205	IF%TEMP -WLVMIN	02204,02204,02206	A0220101
02206	IF%SL -.660	02207,02207,02204	A0220102
02207	IF%SL -.070	02204,02208,02208	A0220103
02208	IF%SNSTRW-4.0	02212,02212,02209	A0220104
02209	IF%SNSTRW-6.0	02210,02213,02210	A0220105
02210	IF%SNSTRW-8.0	02211,02214,02211	A0220106
02211	IF%SNSTRW-10.0	02204,02215,02204	A0220107
C	RULE 1		A0220108
02212	AS#XIS/SSEC		A0220109
	TSEC#AS/%SL*SNSTRW		A0220110
	GO TO 02400		A0220111
C	RULE 2		A0220112
02213	AS#XIS/SSEC		A0220113
	TSEC#AS/%SL*SNSTRW		A0220114
	GO TO 02400		A0220115
C	RULE 3		A0220116
02214	AS#XIS/SSEC		A0220117
	TSEC#AS/%SL*SNSTRW		A0220118
	GO TO 02400		A0220119
C	RULE 4		A0220120
02215	AS#XIS/SSEC		A0220121
	TSEC#AS/%SL*SNSTRW		A0220122
	GO TO 02400		A0220123
C	RULE 5		A0220124
02204	SNSTRW#SNSTRW%1.		A0220125
	GO TO 02300		A0220126

Figure 6. (Continued)

TABLE 023 INPUT				PAGE	13
02301	IF SNSTRW	LE 10	Y N N		
02302	IF ITERA2	LR 3	. Y N		
02303	ADC FORMULA	1	. X .		
02304	INCREMENT ITERA2 BY 1		. X .		
02305	SET SNSTRW	# 1	. X .		
02306	DC FORMULA	2	X X .		
02307	GO TO TABLE	22	X X .		
02308	GO TO TABLE	124	. . X		

				PAGE	14
C	TABLE 023 CONDITION AREA				A0230127
02300	CONTINUE				A0230128
02303	IF%SNSTRW-10.	□02306,02306,02304			A0230129
02304	IF%ITERA2-3	□02307,02305,02305			A0230130
C			RULE 1		A0230131
02306	SL#WLVMAX/SNSTRW-.003				A0230132
	GO TO 02200				A0230133
C			RULE 2		A0230134
02307	WLVMIN#%.95-FLOATF%ITERA2□*.025□*WLVMAX				A0230135
	ITERA2#ITERA2	□			A0230136
	SNSTRW#1.				A0230137
	SL#WLVMAX/SNSTRW-.003				A0230138
	GO TO 02200				A0230139
C			RULE 3		A0230140
02305	GO TO 12400				A0230141

TABLE 024 INPUT				PAGE	15
02401	ASET SNSTR	# 1	X		
02402	DO FORMULA	1	X		
02403	DO FORMULA	2	X		
02404	DO FORMULA	3	X		
02405	GO TO TABLE	25	X		

				PAGE	16
C	TABLE 024 CONDITION AREA				A0240142
02400	CONTINUE				A0240143
C			RULE 1		A0240144
02403	SNSTR#1.				A0240145
	SK#TSEC/SNSTR				A0240146
	TEMP1#.68*SL				A0240147
	TEMP2#.13*SL				A0240148
	GO TO 02500				A0240149

Figure 6. (Continued)

TABLE 025 INPUT			PAGE	17
02501 IF SNSTRY	LE 4	Y E		
02502 IF SK	LE .120	Y E		
02503 IF SK	GE .040	Y E		
02504 IF TEMP1	GE SK	Y E		
02505 IF TEMP2	LE SK	Y E		
02506AINCREMENT SNSTRY	BY 1	. X		
02507 GO TO TABLE	26	. X		
02508 GO TO TABLE	27	X .		

			PAGE	18
C	TABLE 025 CONDITION AREA			A0250150
02500 CONTINUE				A0250151
02503 IF%SNSTRY-4.	□02505,02505,02504			A0250152
02505 IF%SK -.120	□02506,02506,02504			A0250153
02506 IF%SK -.040	□02504,02507,02507			A0250154
02507 IF%TEMP1 -SK	□02504,02508,02508			A0250155
02508 IF%TEMP2 -SK	□02509,02509,02504			A0250156
C		RULE 1		A0250157
02509 GO TO 02700				A0250158
C		RULE 2		A0250159
02504 SNSTRY#SNSTRY.	□			A0250160
GO TO 02600				A0250161

TABLE 026 INPUT			PAGE	19
02601 IF SNSTRY	LE 4	Y N		
02602AINCREMENT SNSTRW	BY 1	. X		
02603 GO TO TABLE	23	. X		
02604 DC FORMULA	1	X .		
02605 GO TO TABLE	25	X .		

			PAGE	20
C	TABLE 026 CONDITION AREA			A0260162
02600 CONTINUE				A0260163
02603 IF%SNSTRY-4.	□02605,02605,02604			A0260164
C		RULE 1		A0260165
02605 SK#TSEC/SNSTRY				A0260166
GO TO 02500				A0260167
C		RULE 2		A0260168
02604 SNSTRW#SNSTRW.	□			A0260169
GO TO 02300				A0260170

Figure 6. (Continued)

TABLE 027 INPUT			PAGE 21
02701ADC FORMULA	1	X	
02702 DD FORMULA	2	X	
02703 GO TO TABLE	28	X	

			PAGE 22
C	TABLE 027 CONDITION AREA		A0270171
02700 CONTINUE			A0270172
C	RULE 1		A0270173
02703 XMDS#2.0*%SLAY*%SK&.003#*SNSTR&D# /2.0&A&B&C#&CORENO			A0270174
XLS#3.1416*XMDS			A0270175
GO TO 02800			A0270176

TABLE 028 INPUT			PAGE 23
02801 IF SK	LR .078	Y N	
02802ADD FORMULA	1	X .	
02803 SET CORR	# .00089	. X	
02804 DD FORMULA	2	X X	
02805 DC FORMULA	3	X X	
02806 DD FORMULA	4	X X	
02807 DC FORMULA	5	X X	
02808 DD FORMULA	6	X X	
02809 DD FORMULA	7	X X	
02810 CC FORMULA	8	X X	
02811 DD FORMULA	9	X X	
02812 DD FORMULA	10	X X	
02813 DD FORMULA	11	X X	
02814 GO TO TABLE	29	X X	

Figure 6. (Continued)

			PAGE	24
C	TABLE 028 CONDITION AREA		A0280177	
02800	CONTINUE		A0280178	
02803	IF%SK	-.078 □02805,02804,02804	A0280179	
C		RULE 1	A0280180	
02805	CORR# .21*%SK**2		A0280181	
	ACCT#%SL*SK-CORR□*SNSTRW*SNSTR		A0280182	
	SACT#XIS/AACT		A0280183	
	WTS#LS*ACCT.322PHASE*XNS		A0280184	
	SLOSSD#2.54*WTS*SACT**2*.000001		A0280185	
	SLOSSS#SLOSSS*CYCLE/60.0		A0280186	
	SLOSSC#SLOCCD*%1.0&SLOSSS/100.0□		A0280187	
	III028# 1		A0280188	
	GO TO 02806		A0280189	
02807	SWATIN#SLOSSC/SRAS		A0280190	
	STEMPR#1.3512*SWATIN		A0280191	
	SLOSSB#0.3*XIS		A0280192	
	GO TO 02900		A0280193	
C		RULE 2	A0280194	
02804	CORR #.00089		A0280195	
	ACCT#%SL*SK-CORR□*SNSTRW*SNSTR		A0280196	
	SACT#XIS/AACT		A0280197	
	WTS#LS*ACCT.322PHASE*XNS		A0280198	
	SLOSSD#2.54*WTS*SACT**2*.000001		A0280199	
	SLOSSS#SLOSSS*CYCLE/60.0		A0280200	
	SLOSSC#SLOCCD*%1.0&SLOSSS/100.0□		A0280201	
	III028# 2		A0280202	
	GO TO 02806		A0280203	
02808	SWATIN#SLOSSC/SRAS		A0280204	
	STEMPR#1.3512*SWATIN		A0280205	
	SLOSSB#0.3*XIS		A0280206	
	GO TO 02900		A0280207	
C		OUT-OF-LINE	A0280208	
02806	SRAS#4.0*PHASE*%SL-6.0□*SL*SNSTRW*XNS/SLAY		A0280209	
	GO TO%02807,02808□,III028		A0280210	

TABLE 029 INPUT				PAGE	25
02901	IF STEMPR	LE 150	Y N		
02902	ADD FORMULA		2 1		
02903	GO TO TABLE		40 30		

			PAGE	26
C	TABLE 029 CONDITION AREA		A0290211	
02900	CONTINUE		A0290212	
02903	IF%STEMPR-150.	□02905,02905,02904	A0290213	
C		RULE 1	A0290214	
02905	ODS#XMDS&D&2.0*%SK&.003□*SNSTR□*SLAY/2.0		A0290215	
	GO TO 04000		A0290216	
C		RULE 2	A0290217	
02904	TEMP#0.6*%STEMPR-150.0□		A0290218	
	GO TO 03000		A0290219	

Figure 6. (Continued)

TABLE 030 INPUT			PAGE 27
03001	IF TEMP	LE 1.0	Y N
03002	ADD FORMULA	1	X .
03003	DO FORMULA	2	. X
03004	DO FORMULA	3	X X
03005	DO FORMULA	4	X X
03006	GO TO TABLE	21	X X

			PAGE 28
C	TABLE 030 CONDITION AREA		A0300220
03000	CONTINUE		A0300221
03003	IF%TEMP -1.0	03005,03005,03004	A0300222
C		RULE 1	A0300223
03005	SSEC#SSEC-.01*SSEC		A0300224
	XIS#XKVA*1000.0/%SECVC*PHASE		A0300225
	WLVMIN#0.95*WLVMAX		A0300226
	GO TO 02100		A0300227
C		RULE 2	A0300228
03004	SSEC#SSEC-.006*%STEMPR-150.0#SSEC		A0300229
	XIS#XKVA*1000.0/%SECVC*PHASE		A0300230
	WLVMIN#0.95*WLVMAX		A0300231
	GO TO 02100		A0300232

TABLE 040 INPUT			PAGE 29
04001	IAPRINT	LIST1	X
04002	PRINT	LIST2	X
04003	PRINT	LIST3	X
04004	STOP	999	X

			PAGE 30
C	TABLE 040 CONDITION AREA		A0400233
04000	CONTINUE		A0400234
C		RULE 1	A0400235
04003	PRINT 04004,	SL,SK,SNSTRW,SNSTRY,WYS,SLOSSC,STEMPR	A0400236
04004	FORMAT		A0400237
	PRINT 04005,	HCOIL,SECVC,XNS,XLS,SRAS,SWATIN	A0400238
04005	FORMAT		A0400239
	PRINT 04006,	SSEC,SLAY,XIS,AS,XMPS,ODS	A0400240
04006	FORMAT		A0400241
	STOP 999		A0400242

Figure 6. (Continued)

SUGGESTED CODING PRACTICES

Several ways by which fast and efficient use can be made of the Decision Logic Translator are described below.

- Formulas and lists need not be sequence-numbered (have row numbers) but must be table-numbered.
- As stated in the Input/Output Description above, table numbers must be lower than 999. However, 999 can be used if it is not the number for the last table of the input to a translator run.
- In order to get the most efficient FORTRAN coding of the action portion of a table, tables with a large number of sequential actions which are common to all rules of the table may be split into two tables. The first of the two split tables would contain the conditions and the uncommon actions; the second would be an unconditional table containing the set of common actions. Thus, splitting enables the translator to code the common actions once only, whereas without splitting they would be coded as many times as there are rules.
- The statement numbers for FORMAT statements and the FORMAT cards are supplied by the translator, but the user must insert the correct FORMAT information into each FORMAT card. All FORMAT cards immediately follow their respective input/output cards. An easy way to determine which list number a given FORMAT number came from is as follows:
 1. If the input/output coding is inline and the statement in question is the third input/output statement executed in rule 2, then the list number is the number of the list of the third I/O command in the actions for rule 2.
 2. If the input/output coding is out of line, the number of statement numbers in the computed GO TO following the FORMAT indicates the number of Xs in the entry portion for the I/O command in question. It is then fairly easy to discover which list in the original coding had that many references to it and thus to be able to identify the correct correlation between an I/O coding list and its FORMAT statement.
 3. The variables in the list of the FORTRAN I/O statement may also be used to correctly identify the FORMAT statement with the original I/O coding and list number in the input table.
- Closed tables may be called by closed tables as well as by open tables. The user must be careful not to have a GO TO TABLE command in a closed table. The only exit commands which can be used in a closed table are STOP, PAUSE, ERROR and RETURN. As was stated earlier in the manual, every closed table must have a RETURN command as its last row. This command serves to define a table as being a closed table, as well as to allow the translator to correctly code the statement to return to the correct calling table.

PROGRAM SETUP

1401								
INPUT				OUTPUT				
Unit	I. D. or Description		Source		Unit	I. D. or Description		Disposition
1402	Decision table cards		Keypunch		1402	FORTRAN source prog.		Compilation
TU 1	Closed table references		Output phase I		1403	Input list and FORTRAN source list		Eng.
TU 2	Reduced table		Output phase I		TU 1	Closed table references		Input phase II
					TU 2	Reduced table		Input phase II table
Sense Sw.	A	B	C	D	E	F	G	
On	X							
Off		X	X	X	X	X	X	
<u>Setup</u>								
<ol style="list-style-type: none"> 1. Reset computer. 2. Place translator program deck in 1402 reader followed by decision table cards. 3. Ready tapes on tape units 1 and 2. 4. Press load. 								

CONSOLE OPERATING INSTRUCTIONS

1. Mount tapes on units 1 and 2. Ready printer and punch. Sense switch A on, B-G off.
2. Load deck in card read hopper — program deck followed by identification card (if supplied) followed by source tables. The object deck must be supplied each time. The program is never loaded from tape.
3. Reset computer and program load.

ERROR CODES AND MESSAGES LIST*

<u>Error Code</u>	<u>Description</u>	<u>Action Taken</u>	<u>Translator Source List Page/Line No.</u>
001	Row is out of sequence.	Go to next table.	03/04
002	Continuation number is out of sequence.	Go to next table.	03/20
003	Table has no rows.	Go to next table	02/43
004	Table has no action area (NOTE: formula or list cards could be in front of the action area).	Go to next table.	02/50
005	IF keyword missing, or A missing in first action row.	Go to next table.	03/52
006	Operand 1 of condition row is not a variable name.	Go to next table.	03/54
007	Operand 1 variable name is too long.	Go to next table.	03/56
008	Condition operator in stub is invalid.	Go to next table.	03/77
009	Invalid entry code in action row.	Go to next table.	04/19
010	Invalid entry code in condition row.	Go to next table.	04/15
011	Extended and limited entry format in the same table.	Go to next table.	04/51
012	Invalid operator code in extended entry condition row.	Go to next table.	05/29
013	Number of rows exceeds the maximum allowed.	Go to next table.	08/66
014	Number of rows exceeds 40.	Go to next table.	08/18
015	One column in conditional table or no columns in unconditional table.	Go to next table.	08/16

*Translator continues processing after printing the error number.

<u>Error Code</u>	<u>Description</u>	<u>Action Taken</u>	<u>Translator Source List Page/Line No.</u>
016	Unconditional table containing more than one column.	Go to next table.	08/12
017	Number of columns is unequal to number of columns in first row of table.	Go to next table.	08/22
018	Number of rows exceeds the smaller of 99 or 999 (number of columns).	Go to next table.	02/84
019	The columns of a limited entry row are mispunched giving two punches in a single column.	Go to next table.	04/05
020	Matrix requires more than 999 positions.	Go to next table.	05/33
021	Matrix requires more than 999 positions.	Go to next table.	05/34
022	Matrix requires more than 999 positions.	Go to next table.	04/44
023	Floating-point constant appearing in the same statement as a fixed-point variable.	Ignore condition.	05/95
024	Formula or list number greater than two digits or table number greater than three digits.	Go to next table.	06/12
025	Trying to adjust value of a constant with an arithmetic statement (Ex: MOVE M to 5).	Go to next table.	06/97
026	TO missing in a MOVE stub.	Ignore condition.	06/81
027	Operand 2 of MOVE instruction is longer than six digits.	Go to next table.	06/99
028	Operand 1 of INCREMENT or SET instruction is longer than six digits.	Go to next table.	07/01

<u>Error Code</u>	<u>Description</u>	<u>Action Taken</u>	<u>Translator Source List Page/Line No.</u>
029	= missing in a SET instruction.	Ignore condition.	07/04
030	BY missing in an INCREMENT instruction.	Ignore condition.	07/19
031	Action is illegitimate, or operand 2 is punched in wrong position.	Go to next table.	07/31
032	Tape number is greater than two digits or LIST is not present in tape instruction.	Go to next table.	07/56
033	LIST is not the first four characters in a nontape I/O instruction.	Go to next table.	07/71
034	RETURN row is not the last row in the table.	Go to next table.	03/39
035	Closed table has a GO TO TABLE or ERROR director action outside of the table.	Go to next table.	10/01
036	Two or more director actions for one column.	Go to next table.	09/76
037	Variable of the form IIIInn or JJJInn in operand 1 of condition stub.	Go to next table.	03/84
038	Variable of the form IIIInn or JJJInn in operand 2 of a condition stub or arithmetic statement.	Go to next table.	06/17
039	Trying to adjust the value of a constant with an extended entry MOVE instruction (see error 25).	Go to next table.	06/21
040	Mixed instruction--contains both a fixed- and floating-point expression.	Ignore condition.	06/36
041	Variable of type IIIInn or JJJInn in operand 1 of action row.	Go to next table.	07/86

<u>Error Code</u>	<u>Description</u>	<u>Action Taken</u>	<u>Translator Source List Page/Line No.</u>
042	Column does not have director action.	Go to next table.	10/07
043	Extended entry row contains a variable name with more than six digits.	Go to next table.	05/506
501	Two condition columns are identical.	Go to next table.	07/170
503	An action column can never be reached by the object program.	Go to next table.	18/280
504	Word FORMULA or LIST is missing.	Try to interpret remainder of card.	52/150
505	Formula (list) number missing on formula (list) card.	Go to next table.	27/040
506	More than seven continuation cards in text of formula (list).	Ignore any additional continuation cards.	52/020
507	Formula (list) requested but not supplied.	Treat missing formula (list) as blank but valid input.	54/150
508	More than 20 closed tables requested.	Go to next table.	35/090
509	More than 50 references to a closed table.	Ignore all references except the first 50.	53/150
510	Closed table referenced after being processed.	Ignore condition.	52/240
511	Closed table never referenced.	Ignore condition.	53/180
512	Formulas and lists not last in table.	Go to next table.	26/033

ERROR AND RESTART PROCEDURES

Halts

<u>A and B Registers</u>	<u>Meaning</u>
77x	Tape error on tape unit x. Push START to try again. *
99x	End of file or reel on tape unit x. *
999	End of job.

*If the error condition persists, the run can be continued by mounting another tape on the drive that caused the error and rewinding both tapes. A new input deck needs to be created by placing the run identification card (if present) in front of all unprocessed tables, including the table in which the error occurred. The program followed by the new input deck is then loaded. Since the program makes heavy use of the tapes, it is recommended that the tapes be in good condition.

STORAGE MAP

<u>Locations</u>	<u>Phase 1</u>	<u>Phase 2</u>
333-600	Constants used by both phases.	Constants used by both phases.
601-791	Print, Read and Error routines.	Print, Read and Error routines.
850-926	Loader (EXECU).	Loader (EXECU).
1000-1999	Area where a row is assembled.	AMTRX--array area.
2000-3999	Not used.	CMTRX--source cell number.
4000-5999	Not used.	DMTRX--failure path number.
6000-7999	Not used.	STUB.
8000-15997		Phase 2 coding.
8000-12999	Phase 1 coding.	
13000-13998	MATRXA--array area.	
14000-15997	STUB.	
15998	Groupmark used to write program on tape.	Groupmark used to write program on tape.