U2769.2

OCEAN DATA SYSTEMS, INC.
6000 EXECUTIVE BLVD., ROCKVILLE, MARYLAND 20852 • 301/881-3031

Mar. 2, 1974

Prof. D. Knuth
Stanford University
Stanford, Calif.

Dear Prof. Knuth,

A letter came today from Fred Gruenberger, and sure enough, I had overlooked the specific paper he had in mind. Actually, it's not a paper, but a Rand Publication that's part of the documentation for a large computer program - I was project leader for this back in 1961-63. Fred particularly likes one or two diagrams which give examples of "structured programming". Actually, with a few hints, mostly letting the reader a) what we're up to, and b) what the programming conventions are (and that they indeed hold), the program is practically self-documenting. (I can still read it 10 years later).

Since I think there's other good material in the report too (I like the flow charts especially, since their main purpose is to resolve the go to problem) I'm sending along more than Fred had in mind. If you want the whole thing, call Rand's Report Department (or if they give you a hard time, my ex sec'y at Rand, Mrs. Janet Lindholm) and ask them to send

Rand Memorandum RM-3721-ASDC, July 1963
Army Cost Model Programmer's Reference Manual

I trust the previous material (still operative!) reached you ok.

Sincerely
Chuck Baker

P.S. Let me know what you make of all this (& previous) stuff!

MEMORANDUM

RM-3721-ASDC

JULY 1963

ARMY COST MODEL
PROGRAMMERS' REFERENCE MANUAL

C. L. Baker

This research is sponsored by the Department of Defense under Contract SD-83, monitored by the Assistant Secretary of Defense (Comptroller). Views or conclusions contained in this Memorandum should not be interpreted as representing the official opinion or policy of the Department of Defense.

The RAND Corporation

1700 MAIN ST. • SANTA MONICA • CALIFORNIA

FOREWORD

From the moment the very first instruction for the Army Cost Model was written on a coding sheet, every attempt has been made to work towards a symbolic assembly listing that would be as completely self-documenting as possible. To this end, a fairly rigid set of conventions has been adopted, so that there would be a consistent "style" of programming throughout the Model.

The most fundamental conventions are concerned with the way in which the individual programs comprising the Model are broken up into a logical hierarchy of routines, and the way in which these routines communicate with one another. No less important is the way in which data storage is organized internally: here the goal has been to parameterize the data definition wherever possible. Conventions for references by the routines to the data storage areas involve the specification of control words in standard formats. Finally, and perhaps most significantly, a large number of commentary cards has been included, detailing each program in as precise terms as possible.

Had the original goal of a completely self-documenting program been reached, there would have been no need for this Memorandum. This Memorandum, then, augments the assembly listing, indexes more salient features of the program for immediate reference, and provides a level of expository discussion not practical within the listing itself. Its aim is to impart to the programmer sufficient detail to enable him to become thoroughly familiar with the program; those working on similar models should also find it useful.

II. PROGRAM DOCUMENTATION

The machine operating instructions for the Army Cost Model in the previous section, along with the other publications describing the data requirements[5] and computational flow,[6] present enough information to permit the operational use of the Model. There is always the possibility, however, that a previously untried combination of data will result in the detection of a "bug" in the program. Hopefully, a more likely event is that continued use of the Model will result in the requirement for an upgraded data base or Model logical structure, necessitating program modifications ranging from minor changes to a complete reprogramming project. While the final documentation of the Model must remain the symbolic assembly listing itself,[7] much information about the code cannot conveniently be presented in the restricted format of such a listing. For this reason, supplementary documentation of the Army Cost Model Code is presented here.

INTRODUCTION TO THE 7090 SYMBOLIC ASSEMBLY LISTING

Organization of the Army Cost Model Code

Primarily due to storage limitations imposed by the 7090 (but also to provide restart capability in case of machine error), the code for the Model is broken up into nine major programs. These are, in order of their execution:

1. INPUT CALCS: Reads all input data and converts it to internal form; sorts the Basic Force Unit data into the order in which it is referenced by Major Force Unit data; collates Major and Basic Force unit data; computes all materiel and personnel requirements.

2. REQ'T SUMS: Sums the requirements for personnel and materiel; writes the Requirement Sums Tape.
3. REQ'TS PRINT: Prints the Requirement Sums Tape.
4. MAT'L ANNEX: Reads the Requirement Sums Tape; calculates and prints the Materiel Annex Output Report; computes the allocated costs for materiel and personnel; writes the Allocated Costs Tape.
5. PROG. TOTALS: Prints the totals, by O.S.D. Program, of all intermediate Major Force Unit Calculations.
6. COST ALLOC: Reads the Allocated Costs Tape; allocates materiel and personnel costs to Major Force Units.
7. OUTPUT CALCS: Computes the final costs for each Major Force Unit; prints the final output report including costs and individual Materiel Annexes.
8. OUTPUT TOTAL: Computes and prints the totals of the final output report, by O.S.D. Program.
9. AGGREGATIONS: Selectively computes and prints the totals of the final output report, in any desired aggregations of Major Force Units.

Programs 1, 2, 4, 6, and 7 comprise an operational model; the remaining programs merely print additional output reports from information computed by one or more of these basic sections.

The individual programs are coded in the SCAT language for the 7090, and operate in the environment of SOS, the SHARE Operating System.[1,2] An attempt was made to keep to a minimum this relationship with a specific operating system, however, with the result that almost all of the program is in standard 7090 machine language. The exceptions are:

a. The macro facilities of SCAT were used to incorporate calling sequences to a tape read/write package; these could equally as well have been coded explicitly. A DETAIL MACRO instruction has been supplied at the beginning of each program so that all generated instructions appear on the assembly listing.

b. The Master Control and tape read/write routines of each program refer to SYSERR and the CORE macro, and the program terminates with a transfer to SYSTEM; these may easily be removed.

c. Some on-line messages are printed via SYSCAP; all of these may be removed.

d. Those routines which print reports use the Output Editor facilities of SOS, including the XFORM, XHEAD, XEJECT, XSPACE, and XPRINT macros; a small subroutine package to simulate these could be added. This would be the major change required to operate under another system.

Organization of the Programs

Each of the programs of the Model, as implied above, is organized into three distinct parts, as follows:

Routines. The first of these parts consists of the individual routines of the program. In specifying the routines which comprise a program, a particular emphasis was put on the creation of a logical structure which would keep each routine as short, as general, and as self-contained as possible. (As a result, many routines appear in several programs; only the names are changed.) Figure 2 presents the assembly listing of a typical routine, and indicates the conventions used in coding individual routines.

The first routine of each program is the Master Control routine, and the first instruction is the entry point to that program. The

master routine is, at all times, the highest level of control for a program, and a good idea of the operation of the Model may be extracted from the commentary appearing in the nine master control routines.

For each of the five major programs of the Model, the detailed organization of the individual routines is given by the flow charts which appear in Appendix E. A complete list of all routines in the Model is presented in Appendix B.

Control Words. The second major part of each program consists of the control words for that program. These control words are used for communication between routines, and for specifying parametrically data and working storage areas in the program. Figure 3 presents the assembly listing for typical control words, and indicates the conventions used.

Data Storage Definition. The final part of each program consists of the pseudo-operations used to define the data and working storage areas for the program. A great deal of the processing done by the Model consists of constructing and scanning tables in working storage; these definitions will be the first part of the Model requiring modification as the nature of the input data changes. Figure 4 presents the assembly listing of the data storage definitions for one program, and indicates the conventions used. (Comparison of Fig. 4 with Fig. 3 will reveal the lack of parameterization, which will require, in general, the corresponding control word to be changed when the storage definition is altered.)

SUPPLEMENTARY PROGRAM DOCUMENTATION

The documentation which follows is primarily concerned with presenting additional information about the first of the nine major programs of the model. There are a number of reasons for this. First, INPUT CALCS is the

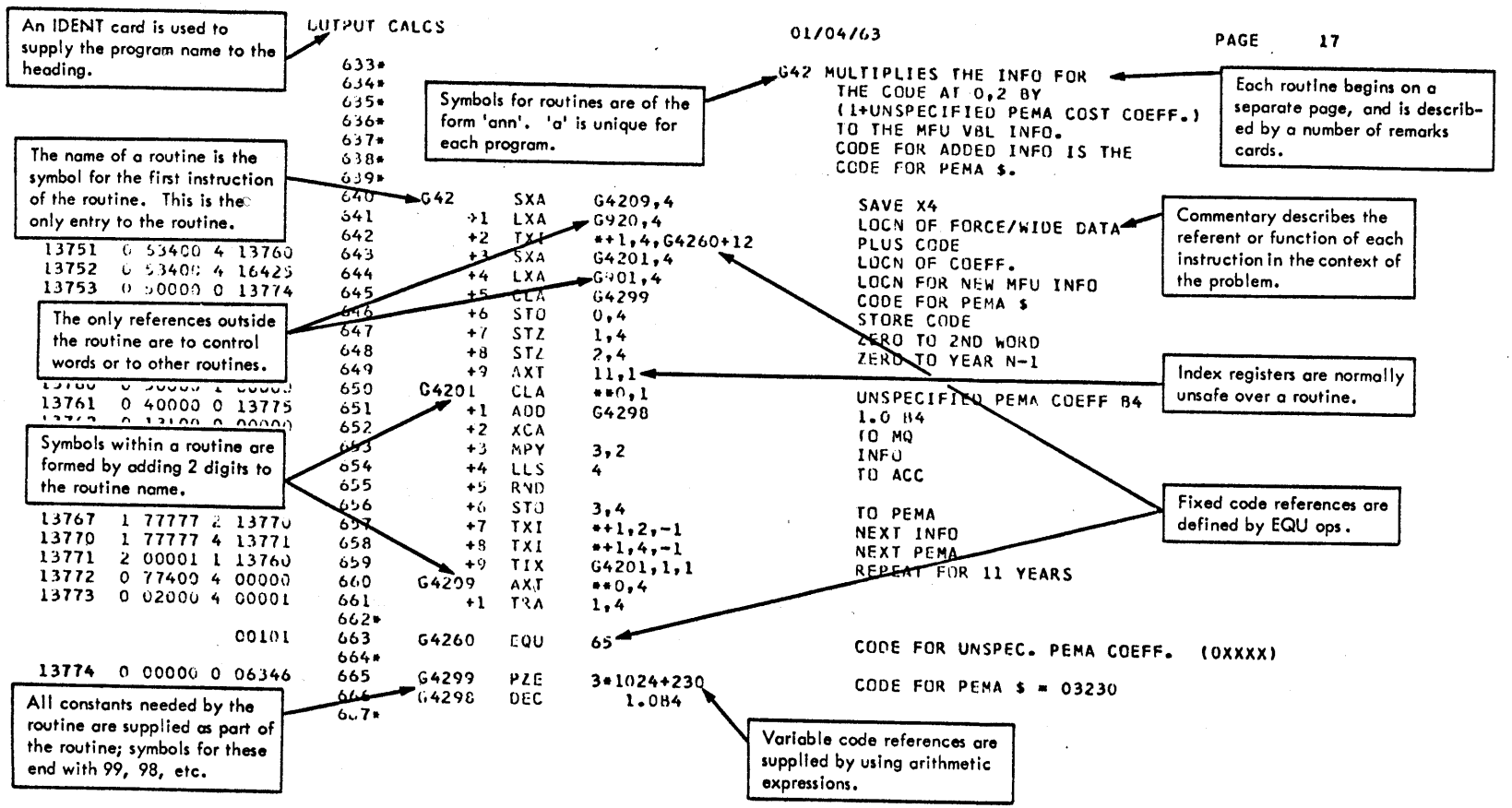


FIGURE 2

CONSTRUCTION OF A ROUTINE IN SCAT LANGUAGE

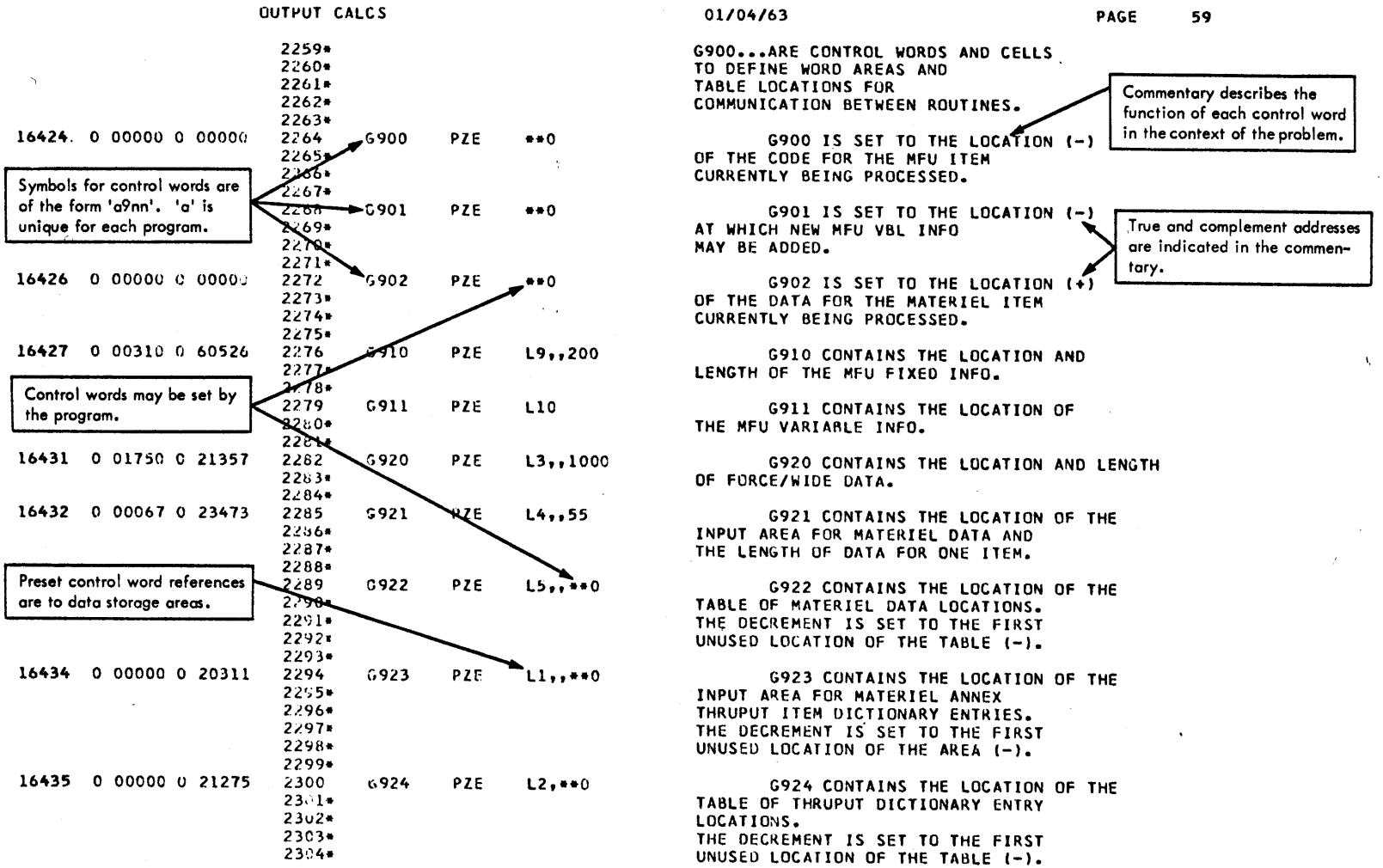


FIGURE 3

CONTROL WORD DEFINITIONS IN SCAT LANGUAGE

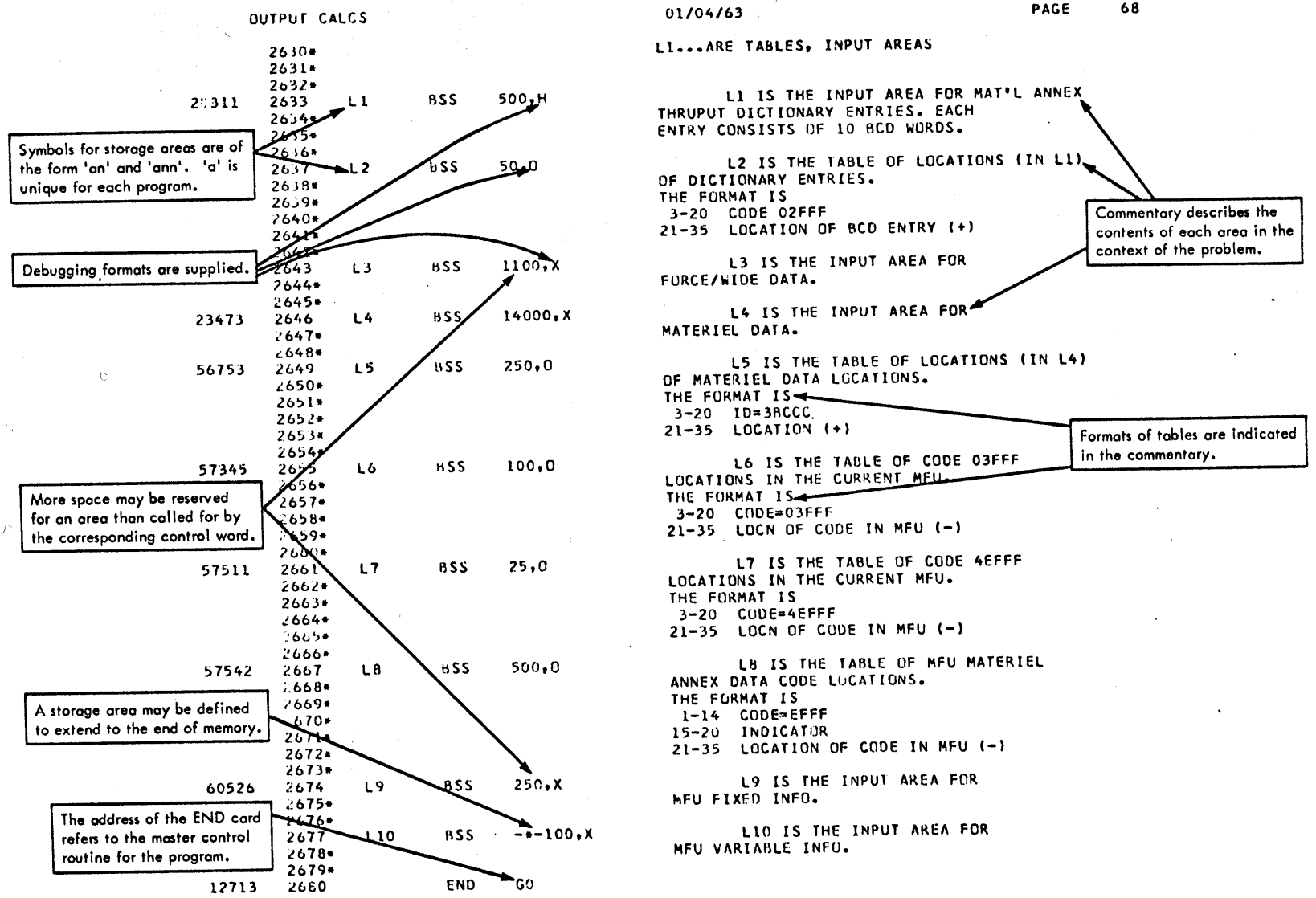


FIGURE 4

STORAGE DEFINITIONS IN SCAT LANGUAGE

longest and most complicated section, including as it does all of the input and conversion routines. To gain operating speed, the required Basic Force Unit data sorting process is interleaved with the input processing. It is in this section that the greatest variety of data appears. Here also is the greatest proportion of non-arithmetic processing. Perhaps most importantly, however, this was the first program to be written, and it was during this programming that the bulk of the finally-adopted conventions evolved. Unfortunately, but predictably, the earliest code was never reprogrammed to adhere to these conventions, and therefore the greatest deviation from the standards occurs in here. Finally, by the time a programmer has worked his way through the program, he should be sufficiently familiar with the Model so that the Symbolic Assembly Listing will indeed serve to document the remaining eight programs.

Tape Read/Write Macro Package

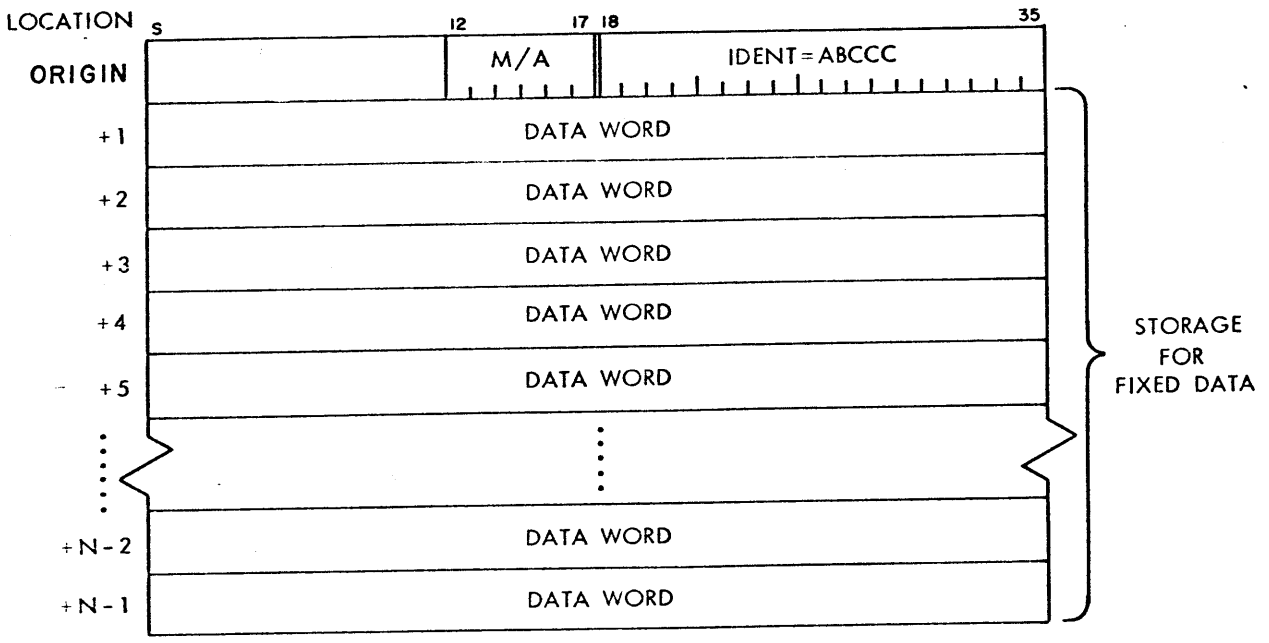
All tape reading and writing functions* of the Model are under the control of a common Tape Read/Write Macro Package. This package provides a complete set of tape select operations, tape checking operations, and channel operations. It is complete and uses no System routines except SYSCAP and SYSERR.

Routines. The calling sequence for each routine in the package is generated by a single macro instruction. The routines may be divided into three groups as follows:

*With the exception of the output report tape, generated under the control of the SOS Output Editor.

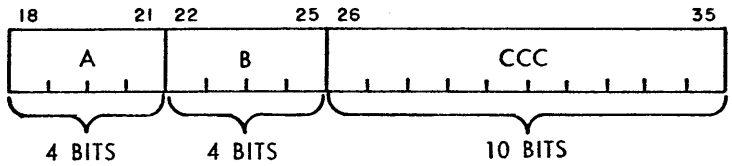
Internal Representation. A unit of data is considered to be all data which has identical values of ID and M/A. As each new unit of data is encountered by the data input routines, two separate tables are created in memory in a standard format; data from each card of that unit is then converted and stored in one or the other of these two tables. The value of the Code appearing in the data card is used to control the internal representation of the data.

Data with Codes 00001 through 00999 is stored in a Fixed Data Table as follows:

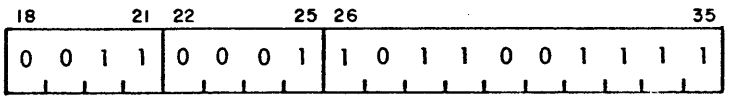


When a new ID-M/A combination is detected during data reading, a table is created (by Routine I26) in this format, and words 1 through N-1 are set to 0.0 or 1.0 (by Routine I27). This, in effect, creates a complete set of Fixed Data, with predetermined values, for all Codes of the form 00FFF.

The standard internal format for ID (ABCCC) in the table heading (and throughout the program) is an 18-bit field composed as follows:



For example, the ID 31719 would appear as:



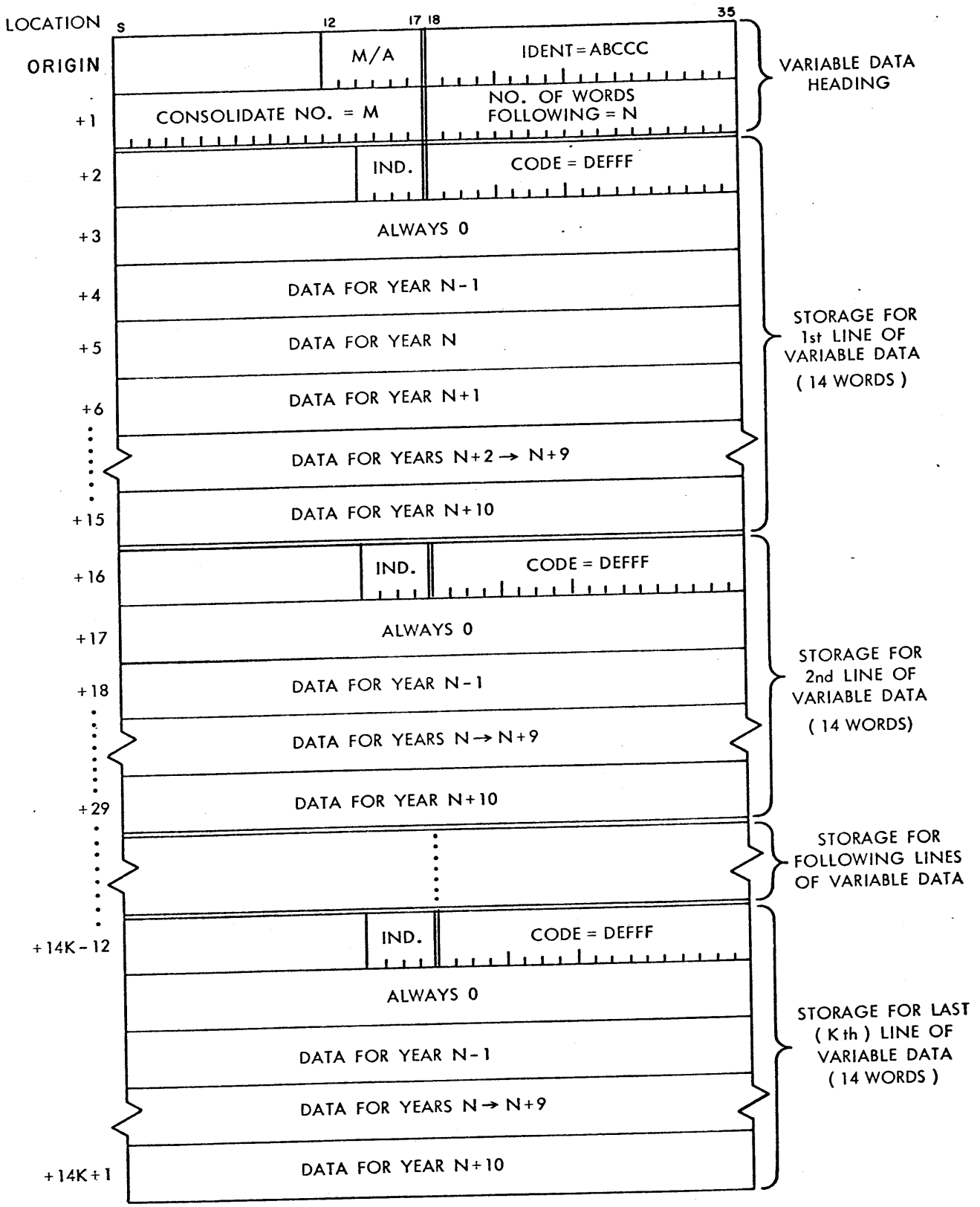
or 143317 in octal.

The Mission/Area (M/A) is carried as a 6-bit field which is a translation of the character M. The translation (and position in the M/A tree) is given by the table at I971 in INPUT CALCS.

The length N of a Fixed Data Table is dependent on the ID for the data, and is a parameter of the computer program; e.g., for ID = 1BCCC, the length is given by the word at 1XXXX in the program as 200_{10}. The origin of the table is determined as data is read, or is predetermined by the program parameters. A complete listing of Fixed Length Data Storage Definitions is given in Appendix A.

As data for Fixed Codes (00001 through 00999) is read and converted, it is stored in the table, replacing the predetermined values. Data for a given Fixed Code is stored in consecutive locations beginning at the relative position corresponding to that Code; the number of words required is determined by the format statement corresponding to the Format (N) on that card. The Code itself is not explicitly entered in the table. For example, data with Code = 00011 and Format = 2, would be stored in 12 consecutive words, beginning with word 11. Data which would extend beyond the limits of the table is ignored.

Data with Codes $\geq$ 01000 is stored in a Variable Data Table as follows:



When a new ID-M/A combination is detected during data reading, a table is created in this format (by Routine I26, at the same time it creates the corresponding Fixed Data Table). The length N of a Variable Data Table is dependent on the number of data lines (one line = data for one Code) for that ID-M/A combination, and is initially set to 0, corresponding to no data in the table. The origin of the table is determined in the same manner as that of the corresponding Fixed Table. A complete listing of Variable Length Data Storage Definitions is also given in Appendix A.

As data for Variable Codes (Codes $>$ 00999) is read and converted, a line of Variable Data is created in the table in the format shown, and N is increased by 14, for each Code. These Codes are explicitly entered in the table (in the same format as the ID), and hence each line of data is uniquely identified.

The consolidate number M and the IND (indicator) bits are not used in the input process and are set to zero.

Data Storage Restrictions. The above conventions for storing data internally imply several restrictions and interrelationships which must be observed. First, for each ID-M/A combination appearing in the input data file, both a Fixed and a Variable Table are formed, even though the current structure of the data does not allow, in some cases, both fixed and variable data to appear in the input file (see Table I). This has been taken into account in the present program. Second, certain precautions must be observed in assigning data Codes and Formats, as follows:

a) No Fixed Code may be larger than the allowed Fixed Table size for that ID-M/A.

b) The Fixed Codes are not carried explicitly in the Fixed Table; therefore, the program itself must "know" what these Codes are: These have been parameterized wherever possible, but appear in many parts of the Model. A complete listing of all Fixed Code references is given in Appendix A.

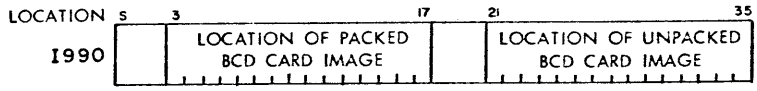
c) A definite relationship exists between Fixed Codes and the Formats for them: The Format for a Fixed Code must always specify a predetermined number of words, and this, in turn, may make other Fixed Codes for that ID-M/A unavailable. For instance, if Fixed Code 00001 is assigned Format F (10 BCD words), this implies the use of relative positions, and hence Fixed Codes 00001 through 00010 are used for storage, and the next available Fixed Code is 00011. Preprinted Codes and Formats on all input sheets are used to comply with this restriction.

d) Since Variable Data always implies data for years N-1 through N+10, the corresponding Formats must always be of this type.

In addition, the convention is made that Fixed Code 00001 is <u>always</u> a BCD TITLE or NAME entry in Format F. These entries are used as dictionary entries throughout the program in too many places to enumerate. Also, Fixed Code 00011 for all M/As of ID = 00001 is automatically filled with the Run Number (from cols. 75-80) by Routine I27.

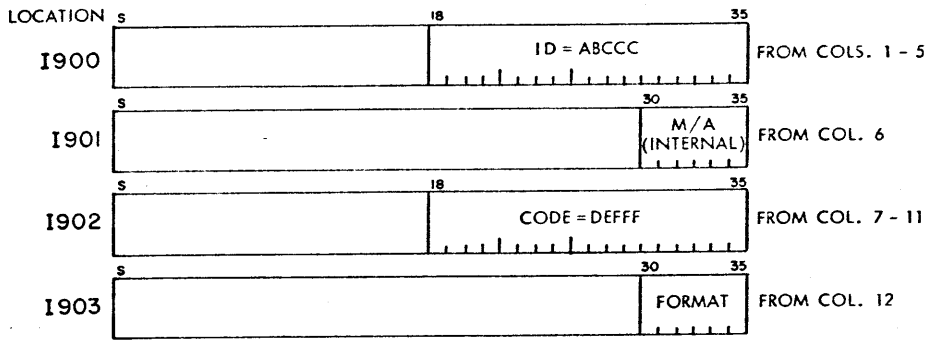
<u>Control Words for Data Input Routines</u>. The Basic Input Routines (I30, I20, I25, I26, I27, I80,...,I99) are under the control of a number of CONTROL WORDS, which specify Data Locations and Lengths, as follows.

Card Image Control Word



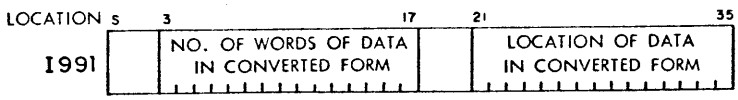
Routine I30 reads a card image from tape into these locations. Routines I20 and I80 through I99 convert information from these locations to internal form. The areas are 14 words and 84 words in length.

Converted Control Information Words



Routine I20 converts cols. 1-12 of the card image into locations I900 through I903, and these words are then used to control all further input processing.

Converted Data Control Word



The Format Routines I80 through I99 convert information from cols. 13-72 of the card image, according to the Format character in I903, and leave the location and length of the resulting information in I991.

Fixed Input Area Control Words

LOCATION	S	3 — 17		21 — 35	
1960		LENGTH OF FIXED AREA		LOCN OF FIXED AREA	FOR ID = 0BCCC
+1		LENGTH OF FIXED AREA		LOCN OF FIXED AREA	FOR ID = 1BCCC
+2		LENGTH OF FIXED AREA		LOCN OF FIXED AREA	FOR ID = 2BCCC
+3		LENGTH OF FIXED AREA		LOCN OF FIXED AREA	FOR ID = 3BCCC
+4		LENGTH OF FIXED AREA		LOCN OF FIXED AREA	FOR ID = 4BCCC

Variable Input Area Control Words

LOCATION	S	21 — 35	
1961		LOCN OF VARIABLE AREA	FOR ID = 0BCCC
+1		LOCN OF VARIABLE AREA	FOR ID = 1BCCC
+2		LOCN OF VARIABLE AREA	FOR ID = 2BCCC
+3		LOCN OF VARIABLE AREA	FOR ID = 3BCCC
+4		LOCN OF VARIABLE AREA	FOR ID = 4BCCC

Converted information (as specified by I991) is stored by Routine I25 into either the Fixed or Variable input area according to the Code of the data (in I902) and the first digit of the ID of the data (in I900). For example, Fixed Data (Code $\leq$ 00999) for ID = 2BCCC would be stored according to the control word at I960+2; Variable Data (Code $\geq$ 01000) for ID = 3BCCC would be stored according to the control word at I961+3.

In addition, Routines I26 and I27 use these control words to initialize the Fixed and Variable input areas whenever a new ID-M/A combination is sensed during input processing.

Note: When data for more than one ID-M/A combination (with the same value of the initial ID digit A) is to be stored in memory at the same time, these control words may be modified by the controlling input routine. For example, Routine I40 modifies control word I960.

Appendix A

CROSS REFERENCES TO THE 7090 ASSEMBLY LISTING

Fixed Length Data Storage Definitions

Variable Length Data Storage Definitions

Fixed Code References

Cost Category Code References

Mission/Area Tree

Mission/Area Tables

Mission/Area Treeing

Fiscal Year Heading Values

OSD Program Tables

FIXED LENGTH DATA STORAGE DEFINITIONS

IDENT	PROGRAM	LENGTH DEFINITION			STORAGE DEFINITION			NOTES
2BCCC	INPUT CALCS	I960+2	PZE	2XXXX,,200	2XXXX	BSS	200	EACH BFU IN SAME LOCN, INFO NOT SAVED.
30CCC	INPUT CALCS	I960+3	PZE	3XXXX,,55	3XXXX	BSS	60	TITLE CODE 00001 ONLY, ALL IN SAME LOCN.
3BCCC	INPUT CALCS	I960+3	PZE	3XXXX,,55	I45+5 → I45+6 I4502+13 → I4502+17			VBL FOLLOWS FIXED INFO FOR ONE ITEM, ALL ITEMS IN ORDER ARE IN 2YYYY+3000.
	MAT'L ANNEX	E931	PZE	Q2,,55	Q2	BSS	100	FIRST M/A ONLY, ALL IN SAME LOCN.
	OUTPUT CALCS	G921	PZE	L4,,55	G40+6 → G40+7 G4001+12 → G4001+15			FIRST M/A ONLY, ALL ITEMS IN ORDER ARE IN L4.
40000	INPUT CALCS	I960+4	PZE	4XXXX,,16	4XXXX	BSS	20	TITLE CODE 00001 ONLY, ALL IN SAME LOCN.
4BCCC	INPUT CALCS	I960+4	PZE	4XXXX,,16	I47+5 → I47+11 I4704+1 → I4704+8			ALL TYPES FOLLOW IN ORDER, NO VBL INFO, ALL FOLLOW 40000 VBL INFO IN 2YYYY+3000.
	MAT'L ANNEX	E932	PZE	Q3,,16	Q3	BSS	100	FIRST M/A ONLY, ALL IN SAME LOCN.

VARIABLE LENGTH DATA STORAGE DEFINITIONS

IDENT	PROGRAM	STORAGE DEFINITION			NOTES
00001	INPUT CALCS	0YYYY	BSS	10	SHOULD NOT APPEAR AS INPUT; DEFINED FOR I26 ONLY.
1BCCC	INPUT CALCS	2YYYY	BSS	-*-100	ALL MFU'S IN SAME LOCN. ALL OF 2YYYY AVAILABLE DURING I11,
		(3YYYY	EQU	2YYYY+7500)	2YYYY THROUGH 2YYYY +7499 AVAILABLE DURING I14,
		(S8	EQU	2YYYY+3000)	2YYYY THROUGH 2YYYY+2999 AVAILABLE DURING I15.
	REQ'T SUMS	R2	BSS	2500	ALL MFU'S IN SAME LOCN.
	PROG. TOTALS	P999	BSS	-*-100	MFU'S CONSOLIDATED IN P999 BY PROGRAM AND ALL PROGRAMS.
	COST ALLOC	M2	BSS	4000	ALL MFU'S IN SAME LOCN.
	OUTPUT CALCS	L10	BSS	-*-100	ALL MFU'S IN SAME LOCN.
	OUTPUT TOTAL	M4	BSS	-*-100	ALL MFU'S CONSOLIDATED IN M4 BY PROGRAM.
		H913	PZE	M4/2+16354	ALL MFU'S CONSOLIDATED IN M4/2+16354 FOR TOTAL ALL PROGRAMS.
	AGGREGATIONS	N5	BSS	-*-100	MFU'S CONSOLIDATED IN N5 BY DESIRED AGGREGATIONS.
2BCCC	INPUT CALCS	2YYYY	BSS	-*-100	2YYYY FILLED WITH BFU'S IN ORDER;
		(S3END	EQU	-2000)	BFU'S WRITTEN ON TAPE WHEN THEY EXTEND BEYOND S3END.
		I43+9 → I43+12 I4304 → I4304+13			BFU'S ARE COLLATED WITH MFU'S DURING I14 IN 2YYYY THROUGH 2YYYY+7499 (SEE 1BCCC ABOVE).

Appendix B

INDEX OF ROUTINES IN THE ARMY COST MODEL

ROUTINES IN INPUT CALCS

I0 Master Control for INPUT CALCS.

I11 Reads All Major Force Unit Data, Writes on Tape T2.

I12 Reads All Basic Force Unit Data, Sorts, Writes Sorted Blocks on Tapes T50+K.

I13 Merges Sorted Blocks of Basic Force Unit Data on Tapes T50+K.

I14 Collates Major and Basic Force Unit Data, Applies Phasing Schedules.

I15 Calculates Materiel Requirements, Applies Personnel Ratios, Calculates Personnel Requirements.

I16 Writes Force/Wide Data, Materiel Data, and Personnel Data on Tape T8.

I20 Converts ID, M/A, Code, and Format to Internal Form.

I21 Converts 5 Columns of Card Image to Internal ID/Code Form.

I22 Converts N Column Field of Card to Binary.

I25 Moves Converted Data to Fixed or Variable Input Area.

I26 Clears Fixed and Variable Input Areas, Stores ID and M/A.

I27 Sets Adjustment Factors in Fixed Areas to 1.0.

I28 Applies Adjustment Factors to MFU or BFU Variable Data.

I29 Multiplies BFU Variable Data by Number of BFU's, Applies MFU Adjustment Factors.

I30 Reads 84 Column Card Image From Tape T1, Converts to Packed and Unpacked BCD.

I31 Writes a Major Force Unit Record on Tape T2.

I32 Sorts and Writes One Block of BFU Variable Data Records on Tape T50+K.

I33 Reads a Major Force Unit Record from Tape T6.

I34 Reads a Basic Force Unit Variable Data Record from Tape T50+K.

I35 Merges Blocks of BFU Variable Data Records, Writes on Tape T50+K.

I36 Ends File and Rewinds All Tapes Written by I32.

I37 Rewinds All BFU Merge Tapes.

Routines in INPUT CALCS (Continued)

I40 Reads All Force/Wide and Mission/Area Data from BCD Input Tape.

I41 Reads All Data for One MFU from BCD Input Tape, Applies MFU Adjustments.

I42 Reads All Data for One BFU (All M/A's) from BCD Input Tape, Applies BFU Adjustments.

I43 Reads One MFU Record and its Referenced BFU Records, Collates and Adjusts.

I44 Reads All Phasing Schedule Data from BCD Input Tape.

I45 Reads All Materiel Data from BCD Input Tape.

I46 Reads All Personnel Support Ratio Data from BCD Input Tape.

I47 Reads All Personnel Data from BCD Input Tape.

I50 Searches MFU Variable Data for BFU References, Adds to Table.

I51 Searches Table of BFU References for References to Current BFU, Adds to Table.

I52 Sorts a Table of One Word Entries.

I53 Searches a Table of One Word Entries.

I60 Locates and Applies Phasing Schedule Data to Each Generic Item of a MFU.

I61 Multiplies Generic Equipment Item Data by One Phasing Schedule Entry.

I71 Writes Force/Wide and Mission/Area Data Record on Tape T8.

I72 Writes Materiel and Personnel Data Records on Tape T8.

I80 Converts Format 12(XXXXX) B35 Blank Field = Value of Prev. Field.

I81 Converts Format 12(X.XXXX) B4 Blank Field = Value of Prev. Field.

I82 Converts Format 6H

I83 Converts Format 12H

I84 Converts Format 24H

I85 Converts Format 36H

Appendix C

ARMY COST MODEL INPUT SHEETS

I	Major Force Unit Data
II	Basic Force Unit Data
III A	Materiel Phasing Schedule
III B	Materiel Data
III C	Materiel Cost Data
III D	Military Personnel Data
III E	Unspecified Unit Personnel Allocation Schedule
IV A	World Wide Data
IV B	Mission/Area Data
IV C	Deliveries to T.O.A. Schedules
V A	Program Add/Change Data
V B	Program Delete Data

MATERIEL DATA

MATERIEL NUMBER 3 ☐☐☐☐ (ALL CARDS) MISSION/AREA ☐ (ALL CARDS) RUN NUMBER ☐☐☐☐☐☐ (ALL CARDS) PAGE 1 OF ____

FISCAL YEAR: N= ______

	CODE	FORMAT	
MATERIEL NAME	00001	F	

	DATA	CODE	FORMAT	N - 1	N	N + 1	N + 2	N + 3	N + 4	N + 5	N + 6	N + 7	N + 8	N + 9	N + 10
YEARLY DATA	UNSPECIFIED UNITS EQUIPMENT ALLOWANCE FACTOR	00023	5												
	REPLACEMENT/CONSUMPTION RATE (/ YEAR)	00039	5												

	DATA	CODE	FORMAT	
NO-YEAR DATA	COMBAT CONSUMPTION RATE (/ MONTH)	00037	V	
	MAINTENANCE FLOAT COEFFICIENT	00038	S	

AMMUNITION DATA

AMMUNITION NAME	DATA	CODE		N - 1	N	N + 1	N + 2	N + 3	N + 4	N + 5	N + 6	N + 7	N + 8	N + 9	N + 10
	COMBAT CONSUMPTION RATE (/ DAY)	33 0	6												
	TRAINING CONSUMPTION RATE (/ YEAR)	33 1	2												
	COMBAT CONSUMPTION RATE (/ DAY)	33 0	6												
	TRAINING CONSUMPTION RATE (/ YEAR)	33 1	2												
	COMBAT CONSUMPTION RATE (/ DAY)	33 0	6												
	TRAINING CONSUMPTION RATE (/ YEAR)	33 1	2												
	COMBAT CONSUMPTION RATE (/ DAY)	33 0	6												
	TRAINING CONSUMPTION RATE (/ YEAR)	33 1	2												
	COMBAT CONSUMPTION RATE (/ DAY)	33 0	6												
	TRAINING CONSUMPTION (/ YEAR)	33 1	2												

OTHER RELATED EQUIPMENT DATA	CODE	
	3	
	3	
	3	
	3	
	3	
	3	
	3	
	3	
	3	
	3	

V A

PROGRAM ADD / CHANGE DATA

V A

RUN NUMBER CHANGE
FROM_____TO_____

NEW RUN NUMBER [ ] ALL CARDS

EFFECTIVE DATE _____

PAGE____OF_____

FISCAL YEAR: N = _____

LINE NUMBER	ID	MISSION/AREA	CODE	FORMAT	FISCAL YEAR N-1	N	N+1	N+2	N+3	N+4	N+5	N+6	N+7	N+8	N+9	N+10
1																
2																
3																
4																
5																
6																
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																
17																
18																
19																
20																
21																
22																
23																
24																
25																

COMMENTS ON CHANGES

Appendix E

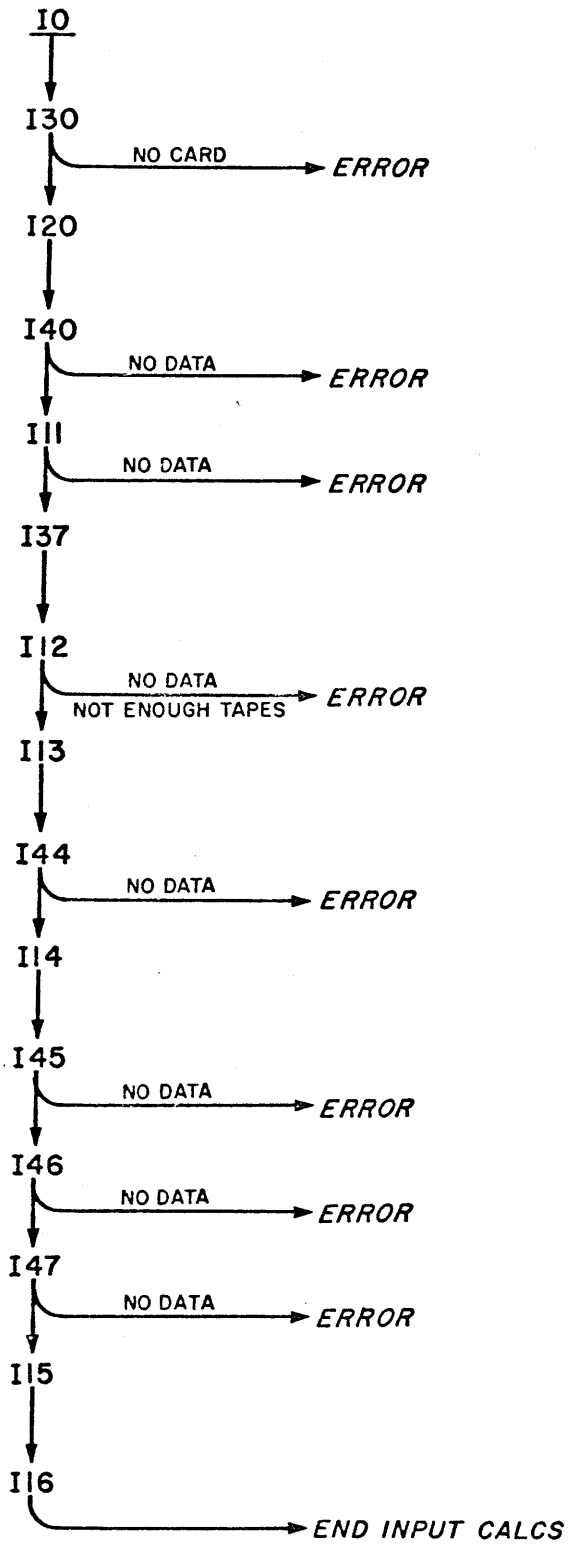
FLOW CHARTS

The flow charts which follow are not intended to present exhaustively the complete detail of the Army Cost Model Program. Instead, they constitute an attempt to present the logical structure of the hierarchy of basic routines from which the program has been constructed. Flow charts are given only for the five major programs which form the heart of the model: INPUT CALCS, REQ'T SUMS, MAT'L ANNEX, COST ALLOC, and OUTPUT CALCS; indeed, these programs themselves make up an operational cost model. The remaining programs, which print out intermediate results from a tape file (REQ'TS PRINT AND PROG. TOTALS) and aggregate the final output (OUTPUT TOTAL and AGGREGATION), are largely rearrangements of routines used in the main programs.

Within each program, flow charts are given only for those routines which call on lower order routines, in an attempt to emphasize the hierarchical structure of the model programming.

IO

MASTER CONTROL FOR INPUT CALCS.



READS 84 COL. CARD IMAGE FROM TAPE T1, CONVERTS TO PACKED AND UNPACKED BCD.

CONVERTS ID, M/A, CODE, AND FORMAT TO INTERNAL FORM.

READS ALL FORCE/WIDE AND MISSION/AREA DATA FROM BCD INPUT TAPE.

READS ALL MAJOR FORCE UNIT DATA, WRITES ON TAPE T2.

REWINDS ALL MERGE TAPES.

READS ALL BASIC FORCE UNIT DATA, SORTS, WRITES SORTED BLOCKS ON TAPES T50+K.

MERGES SORTED BLOCKS OF BASIC FORCE UNIT DATA ON TAPES T50+K.

READS ALL PHASING SCHEDULE DATA FROM BCD INPUT TAPE.

COLLATES MAJOR AND BASIC FORCE UNIT DATA, APPLIES PHASING SCHEDULES.

READS ALL MATERIEL DATA FROM BCD INPUT TAPE.

READS ALL PERSONNEL SUPPORT RATIO DATA FROM BCD INPUT TAPE.

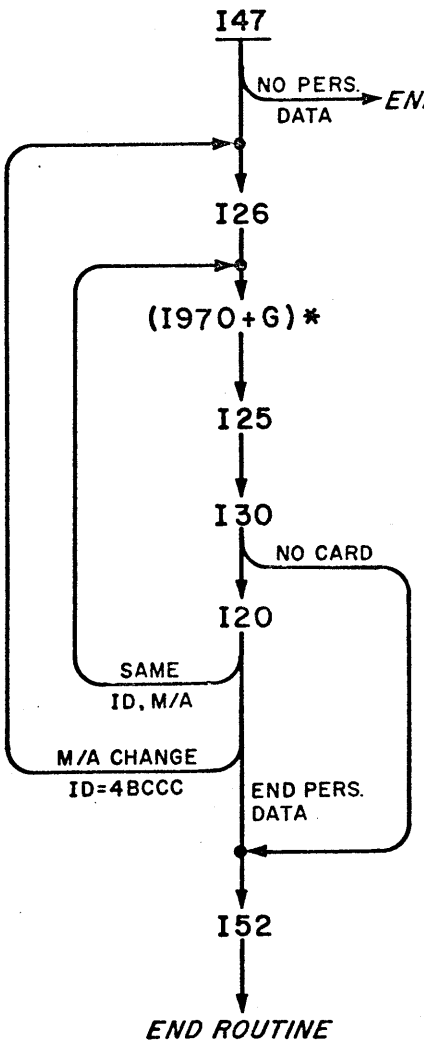
READS ALL PERSONNEL DATA FROM BCD INPUT TAPE.

CALCULATES MAT'L. REQ'TS., APPLIES PERSONNEL RATIOS, CALCULATES PERSONNEL REQ'TS.

WRITES FORCE/WIDE DATA, MATERIEL DATA, AND PERSONNEL DATA ON TAPE T8.

I47

READS ALL PERSONNEL DATA FROM BCD INPUT TAPE.



CLEARS FIXED AND VARIABLE INPUT AREAS, STORES ID AND M/A.

CONVERTS DATA FROM CARD ACCORDING TO FORMAT (G) IN CARD (180...)

MOVES CONVERTED DATA TO FIXED OR VARIABLE INPUT AREA.

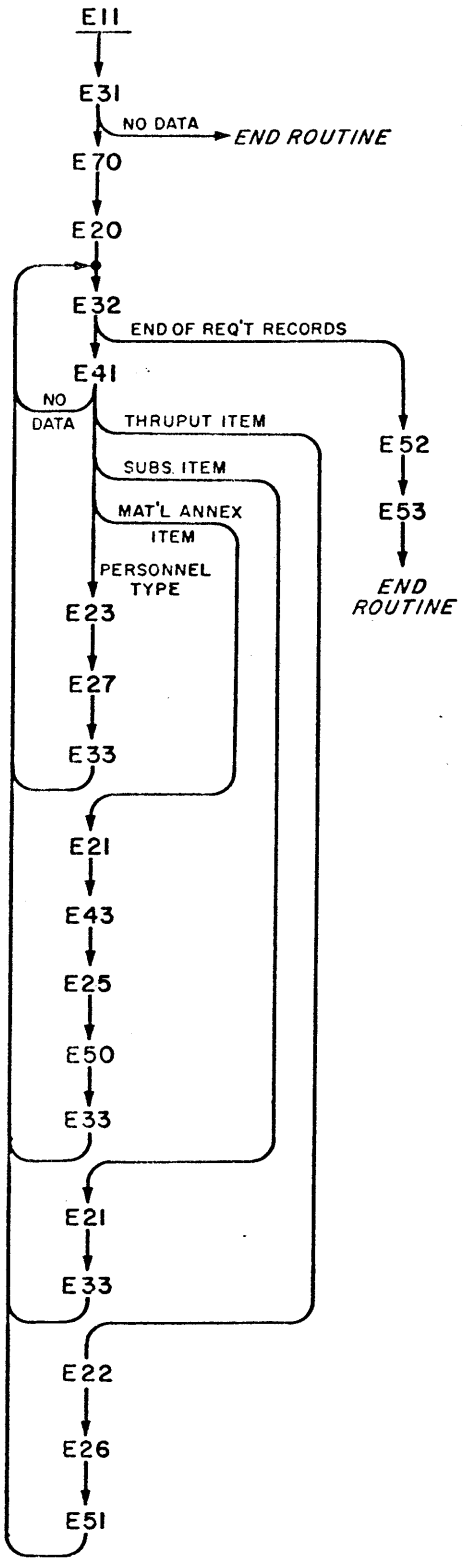
READS 84 COL. CARD IMAGE FROM TAPE T1, CONVERTS TO PACKED AND UNPACKED BCD.

CONVERTS ID, M/A, CODE, AND FORMAT TO INTERNAL FORM.

SORTS THE TABLE OF PERSONNEL DATA LOCATIONS.

E11

COMPUTES MATERIEL ANNEX AND ALLOCATION COSTS, WRITES ALLOCATION TAPE.



READS FORCE/WIDE DATA RECORD FROM TAPE V2.

INTRODUCES AND DEFINES ALL FORMAT STATEMENTS.

CLEARS MATERIEL ANNEX TOTAL AREAS.

READS THE REQUIREMENT SUMS FOR ONE ITEM FROM TAPE V3.

FINDS THE DATA CORRESPONDING TO THE CURRENT MATERIEL ANNEX ITEM.

PRINTS THE MATERIEL ANNEX TOTAL LINES.

PRINTS THE TOTAL PERSONNEL TRAINING COST LINES.

COMPUTES TOTAL TRAINING REQUIREMENTS AND COSTS FOR ONE PERSONNEL TYPE.

ADDS TRAINING COSTS TO TRAINING TOTALS.

WRITES PERSONNEL TYPE ALLOCATION COSTS ON TAPE V4.

COMPUTES TOTAL DELIVERIES AND COSTS FOR ONE ITEM.

CONVERTS DELIVERY QUANTITIES AND COSTS TO T.O.A. FOR ONE MATERIEL ITEM.

ADDS DELIVERY AND T.O.A. COSTS TO MATERIEL ANNEX TOTALS.

PRINTS A MATERIEL ANNEX ENTRY FOR ONE ITEM.

WRITES MATERIEL ITEM ALLOCATION COSTS ON TAPE V4.

COMPUTES TOTAL DELIVERIES AND COSTS FOR ONE ITEM.

WRITES MATERIEL ITEM ALLOCATION COSTS ON TAPE V4.

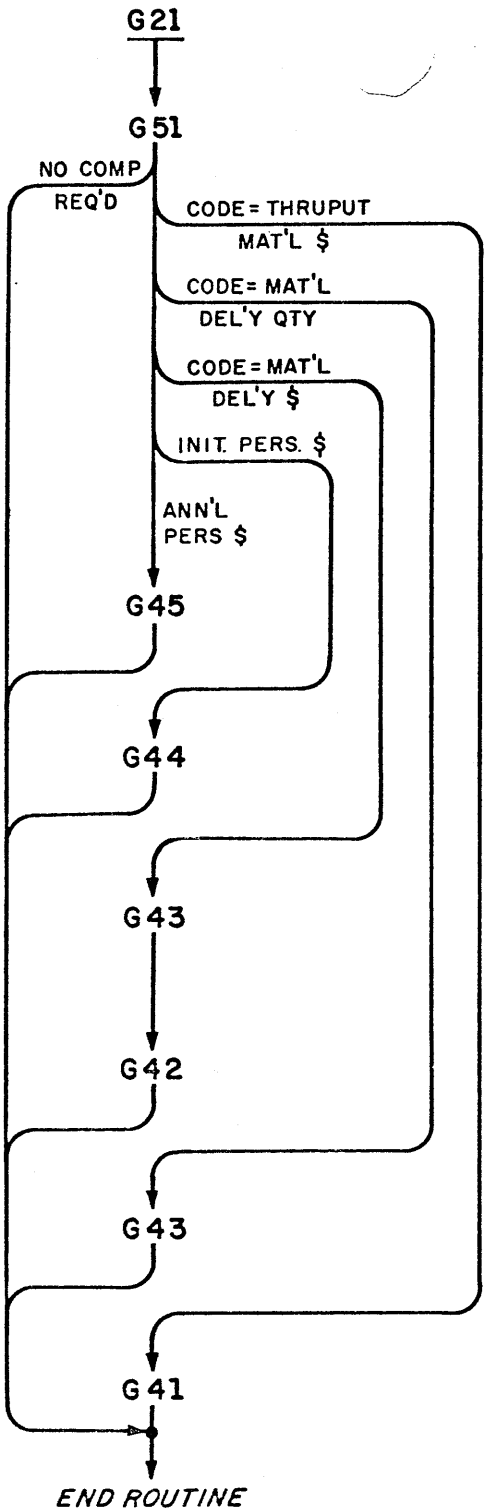
STORES THRUPUT COSTS FOR THRUPUT MATERIEL ANNEX ITEM.

ADDS THRUPUT T.O.A. COSTS TO MATERIEL ANNEX TOTALS.

PRINTS A MATERIEL ANNEX ENTRY FOR ONE THRUPUT ITEM.

G21

COMPUTES FINAL COSTS FOR ONE MAJOR FORCE UNIT CODE.



DETERMINES MFU DATA CODE TYPE, LOCATES CORRESPONDING MATERIEL DATA.

DISTRIBUTES ANNUAL PERSONNEL TRAINING COSTS.

DISTRIBUTES INITIAL PERSONNEL TRAINING COSTS.

CONVERTS MATERIEL DELIVERY $ TO T.O.A.

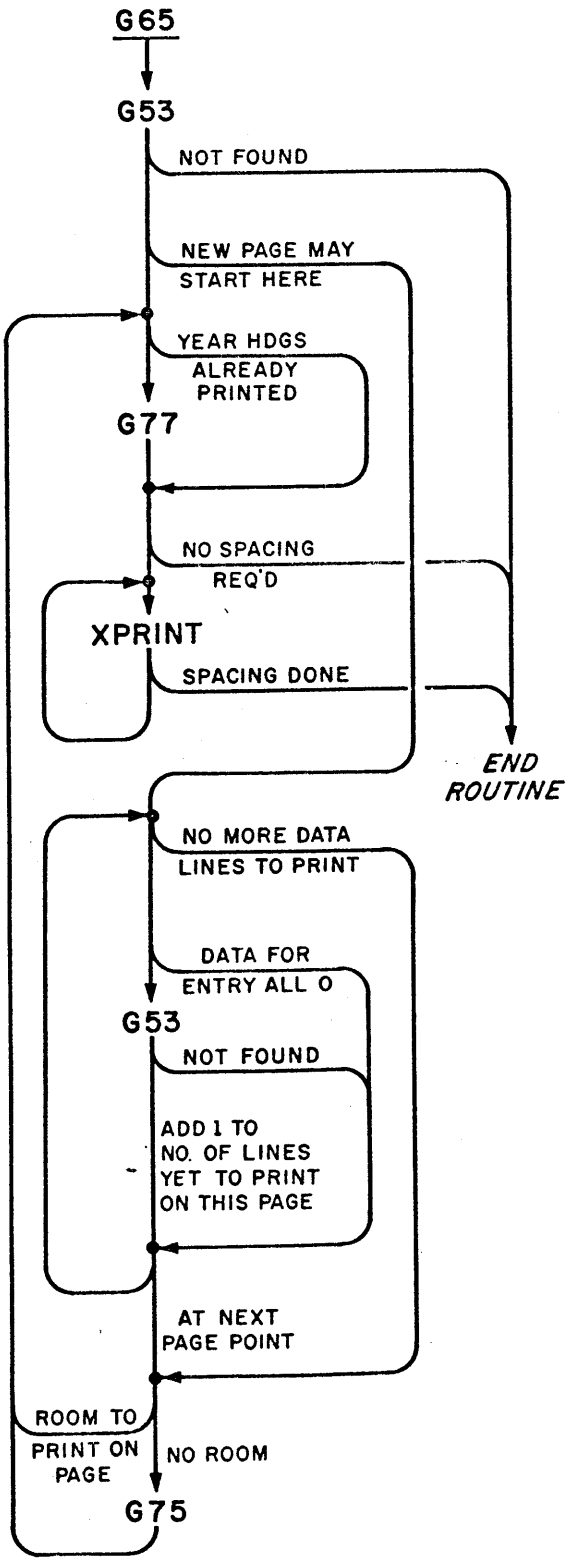
ADDS MATERIEL T.O.A. $ TO PEMA $.

CONVERTS MATERIEL DELIVERY QTY. TO T.O.A.

ADDS MATERIEL THRUPUT T.O.A. $ TO PEMA $.

G65

LOCATES THE BCD STUB ENTRY CORRESPONDING TO A CODE, SPACES AS NECESSARY.



LOCATES STUB ENTRY FOR CODE.

PRINTS THE FISCAL YEAR HEADING FOR THE CHART OF ACCOUNTS.

SPACES ONCE.

LOCATES STUB ENTRY FOR CODE.

PRINTS THE MAJOR FORCE UNIT CHART OF ACCOUNTS HEADING.