Bendix G-15

INTERMAP MANUAL

INTERPRETIVE MATRIX ALGEBRA PROGRAM

DECEMBER 1960



INTRODUCTION

INTERPRETIVE MATRIX ALGEBRA PROGRAM

This program which will be called INTERMAP is an independent programming system, designed to be compatible with INTERCOM 1000 Double Precision. Its purpose is to provide a practical means of performing matrix operations through a coding system which permits the operations on entire arrays of numbers.

INTERMAP was written in response to the interest shown by many G-15D users in the application of matrix methods of analysis to engineering problems. It is, however, also a general purpose mathematical program.

The use of INTERMAP may reduce programming time for many problems which may readily be expressed in terms of simultaneous linear equations. Others, which involve rhythmic operations on arrays of numbers, may be simulated by standard matrix routines. The programming of all such problems may then be handled in two stages:

An INTERCOM 1000 program to generate the co-efficients of the equations or other data, and

an INTERMAP program to solve the equations and or perform other matrix operations.

By adopting such a procedure, the analyst will reduce his own programming efforts to a point where he is mainly concerned with the logical aspects peculiar to his problem.

REFERENCE OF SPECIAL TERMS FOR THE INTERMAP SYSTEM

ACCUMULATOR

Cell, 1.0.2, 2.1.0, 2.3.1, 2.3.4,

First Location (F.L.), 2.1.1, 2.2.3, 2.2.4, 2.2.5, 2.2.6, 2.3.4,

2.4.1, 2.4.4, 2.4.7, 5.2.9

Last Location (L.L.) 2.1.1, 2.2.3, 2.3.4, 2.4.1, 2.4.2, 2.4.4,

2.4.7, 5.2.9,

AMAX, 2.1.0, 2.1.1, 3.1.4, 4.3.1, 4.3.2, 4.3.3, 5.2.9

FILE NUMBER (FN), 1.4.0, 3.2.0, 5.5.2

First Available (FAN), 1.4.0, 4.1.1, 5.5.5, First (FFN), 1.4.0. 4.1.1, 5.5.3, 5.5.5

INCREMENT, MATRIX

Column ($D_j = 2Nmax$), 2.2.1, 2.2.4, Row ($D_i = 2$), 2.2.3, 2.2.4,

INSTRUCTION, INTERMAP, 1.0.1, 1.0.4, 1.0.5, 3.1.5, 3.1.9, 3.2.0, 3.2.2, 3.2.4, 4.2.0, 4.2.1

OPERATION LEVELS

Multiple, 1.0.1, 1.1.2, 1.1.3, Single, 1.0.1, 1.1.2, 1.1.3, 2.1.2, 4.7.1

SEQUENCE, INTERMAP, 1.0.1, 1.1.2, 5.5.5,

SERIAL NUMBER (SR), 1.2.1, 1.3.3, 1.4.0,

TABLE OF CONTENTS

	Page
INTRODUCTION	0.1.1 0.2.1 0.3.1
INTRODUCTORY DESCRIPTION OF INTERMAP Operation Levels and Modes Compatibility With Other Systems Labels and Data Tapes The File Number System	1.0.1 1.0.4 1.1.1 1.2.1 1.3.1 1.4.1
MEMORY ALLOCATION	2.1.1 2.2.1 2.3.1 2.4.1
CODING INSTRUCTIONS	3.1.1 3.2.1 3.3.1 3.4.1
OPERATING INSTRUCTIONS To Start INTERMAP INTERMAP P.P.R Accumulator Selection Routine Typewriter Input Output Routine Punched Card Input Output Routine The Scout Memory Dump Reading Data From Punched Paper Tape Program to Magnetic Tape Loading Routine	4.1.1 4.2.1 4.3.1 4.4.1 4.5.1 4.6.1 4.7.1 4.8.1 4.9.1
THE OPERATION ROUTINES OP = 00 INTERMAP P.P.R. OP = 01 Accumulator Selection Routine OP = 02 Typewriter Input Output Routine OP = 03 Punched Card Input Output Routine OP = 04 Incrementer OP = 20 Store and Transfer Routine OP = 26 Decimal Data Tape Input Routine OP = 21 Matrix Inversion and Equation Solver OP = 24 Matrix Transposition Routine	5.0.1 5.1.1 5.2.1 5.3.1 5.4.1 5.5.1 5.6.1 5.7.1 5.8.1

TABLE OF CONTENTS (Continued)

	Page
OP = 25 Matrix Multiplication Routine	
(Comprehensive)	5.10.1
OP = 22 Matrix Multiplication Routine (Short)	5.11.1
OP = 23 Matrix Addition Routine	5.12.1
OP = 10 Dot Product Multiplication Routine	5.13.1
EQUIPMENT	6.0.1
Different INTERMAP Versions	6.1.1
THE SCOUT AND MEMORY DAMP	7.0.1
Program to Magnetic Tape Loading Routine	7.1.1

INTRODUCTORY DESCRIPTION

DEFINITION

INTERMAP is a double precision floating point interpretive system, compatible with INTERCOM 1000 double precision, which permits the coding of problems in terms of matrix operations only.

OPERATION

On the Multiple Instruction Level up to 108 interpretive INTERMAP Instructions may be stored in memory. Instructions are executed sequentially, however, incrementing instructions, unconditional and conditional transfers are also available, thus permitting more refined programming techniques. An INTERMAP sequence may consist of more than 108 INTERMAP Instructions since additional blocks of pre-punched program tape may be read in under program control.

On the Single Instruction Level INTERMAP Instructions may be typed in and executed one at the time.

EQUIPMENT

Every INTERMAP Instruction may be executed with only the basic G15D or with any combination of four magnetic tape units, three auxiliary photo-readers, and a punched card coupler with assorted punched card equipment. One of the four magnetic tape units may be used for rapidly accessible INTERMAP Program storage. Auxiliary photo-readers, too, may be used for either data or programs. Punched card equipment is used exclusively for the input and output of data. While auxiliary equipment is not a prerequisite for any specific use of INTERMAP, the reading of program tape from auxiliary photo-readers or magnetic tape speeds up the selection of Operation Routines. Similarly, the use of magnetic tape for data storage as an extension of the G15D memory and/or several auxiliary photo-readers will make data handling more efficient.

ACCUMULATORS

An accumulator is a well defined portion of the memory reserved to accept data. Each matrix is stored in one addressable accumulator. There are seven such addressable accumulators available. Their location in the memory and their size and shape is at the discretion of the programmer and may be changed manually or under program control. Thus each one of these seven accumulators may, in turn, contain any set of data placed anywhere in the memory.

For each accumulator, there is an Accumulator Cell, stored in the control portion of the INTERMAP Program. The Accumulator Cell contains the information as to the location of the accumulator, its size and shape and the order of the matrix presently contained in it. The largest single matrix which may be stored in the G15D memory is of 28 x 29 dimensions. The number of smaller matrices which may be stored is limited by the total number of elements which may not exceed 812.

The arrangement of the accumulators is such that several of them may be defined within a portion of the memory assigned to a larger accumulator. Thus the partitioning of a large matrix into parts, and the composition of a larger matrix from several smaller ones can be performed under program control.

COMPATIBILITY TEST

Each INTERMAP operation which results in a matrix being transferred to an accumulator is subjected to a compatibility test. This test insures that the destination accumulator is large enough to hold the matrix in question, and also that no valid data outside the boundaries of the destination accumulator is inadvertently destroyed. If such a test is not satisfied, the operation will be terminated by an Error Halt before any erroneous information can be transferred to a Destination.

VALIDITY TEST

Matrix algebraic operations are controlled by rules which relate the magnitude of the operand, operator and the results. Each INTERMAP Routine performs its own such tests which will be referred to as Validity Tests. One such test, for example, is in the Matrix Multiplication Routine. It tests the number of columns in the pre-multiplier against the number of rows in the post-multiplier. If a validity test is not satisfied, the operation will be terminated by an Error Halt before any erroneous information can be transferred to a Destination.

LIBRARY OF OPERATION ROUTINES

The INTERMAP Program consists of a Control and a library of Operation Routines. Operation Routines are identified by an Operation Code (or op-code), and are stored on a library type program tape in the order of ascending op-codes. The INTERMAP Control automatically selects from this library the Operation Routine which corresponds to the op-code of the current INTERMAP Instruction.

The following Operation Routines are included in the INTERMAP Program:

Category	Op-Code	Operation Routine
Service	.00	INTERMAP P. P. R.
Service	. 01	Accumulator Selection Routine
Service	. 02	Typewriter-Input/Output Routine
Service	. 03	Punched Card-Input/Output Routine
Special	. 04	Incrementer
Algebraic	. 10	"Dot Product" Multiplication Routine
Algebraic	. 20	Store and Transfer Routine
Algebraic	. 21	Matrix Inversion and Equation Solver Routine
Algebraic	. 22	Matrix Multiplication Routine (Short)
Algebraic	. 23	Matrix Addition Routine
Algebraic	. 24	Matrix Transposition Routine
Algebraic	. 25	Matrix Multiplication Routine (Comprehensive)
Algebraic	. 26	Decimal Data Tape Input Routine

The INTERMAP Program provides for a maximum of 64 different op-codes. With the exception of the Service Type Operation Routines (OP = .00, .01, .02, .03) these op-codes have no significance other than to determine the sequence in which the Operation Routines are arranged on the INTERMAP Program tape.

THE INTERMAP INSTRUCTION

The INTERMAP Instruction is a three address code which consists of the following seven fields:

1	2	3	4	5	6	7
OP	С	A	Т _S	SRC	$^{\mathrm{T}}\mathrm{_{D}}$	DST
			Data	Sourçe	Data De	stination
Operation Code	Mode or Characteristic	Operand Accumulator	Auxiliary Equipment	, Operator	Auxiliary Equipment	Result

Table 1.0.1 The Algebraic INTERMAP Instruction

Thus one INTERMAP Instruction may involve data from all three media of data storage i.e., Accumulator (G15D Memory), punched paper tape and magnetic tape. It should be noted that the input and output of data through the typewriter and through punched card equipment is considered an input/output operation rather than logically transferring data from a Source to a Destination.

In INTERMAP Instructions for service type Operation Routines, the various fields of the instruction assume a different meaning. The coding instructions for these op-codes are explained in the section on INTERMAP coding.

The following is a brief description of the available Operation Routines.

INTERMAP P. P. R.

This routine generally performs the same functions as the Standard P. P. R. in that it accepts the decimal type ins of INTERMAP Instructions as well as the input of hexadecimal constants. It produces INTERMAP Sequence tapes or reads in such tapes for execution. It also contains a Sequential Lister.

INTERMAP P. P. R. also controls the Single and Multiple Operation Levels mentioned above, as well as all the error indications.

ACCUMULATOR SELECTION ROUTINE

This routine controls the location, size and shape of each accumulator in the memory by permitting manual or automatic changes to the Accumulator Cells.

TYPEWRITER-INPUT/OUTPUT ROUTINE

This routine permits selective decimal type-in and type-out of data from any accumulator. Input-output instructions may be given for an entire matrix, a row, a column, a diagonal or parts thereof. This includes the automatic storing of zeros or clearing. Also included is a matrix shrinker.

PUNCHED CARD-INPUT/OUTPUT ROUTINE

A data card consists of one element a. (a double precision floating point decimal number) and i and j, the row and column numbers. The output operation is selective and similar to the one described above. No cards are punched for zero elements. The input operation is as follows. The INTERMAP Instruction specifies the part of the accumulator into which data is to be read, for example, a row. The program proceeds to read the entire deck of cards and stores every element whose i and j falls into the selected part, all other data is disregarded.

Each deck of cards, on input or output, starts with a First Card, which is equivalent to a label on tape, and is terminated by a Last Card.

INCREMENTER

The Incrementer permits continuous automatic operation of INTERMAP by facilitating unattended running of INTERMAP Sequences over long periods of time. Since in most such operations the use of magnetic tape may be assumed, emphasis is placed on functions which are most useful with magnetic tape for data storage.

Specifically, the Incrementer increments the Source or Destination of an INTERMAP Instruction and tests it against a limit. It also modifies INTERMAP Instructions and performs unconditional transfers. Thus it permits looping and more refined programming techniques.

STORE AND TRANSFER ROUTINE

This is not an algebraic routine since it does not have an operand. The purpose of this operation routine is to transfer data (matrices) from a Source to a Destination. The Source may be an accumulator, one of four photoreaders, or one of four magnetic tapes. The Destination may again be an accumulator, a paper tape punch or one of three magnetic tape units.

Among other uses this is also a tape input/output routine. Matrix data tapes may even be duplicated by this means (Source: Photoreader; Destination: Paper Tape Punch).

Every transfer between a Source and a Destination may be performed in one of four modes, namely: Clear and Add; Clear and Subtract; Clear; and Transpose.

DECIMAL DATA TAPE INPUT ROUTINE

This routine provides a one-way flow of decimal offline prepared data tape to an accumulator. The decimal format is the same as outlined in the INTERCOM 1000 operating manual.

MATRIX INVERSION AND EQUATION SOLVER ROUTINE

This routine operates on the matrix specified in the INTERMAP Instruction as the operand. In the Inversion Mode, in addition to an inverse, several solutions to the given set of equations may be obtained. In the Equation Solver Mode, no inverse is obtained, which reduces computing time to approximately 50% of the inversion time.

The following are the computing times in minutes:

t = .004N³

For the inversion of one Nth order matrix of NxN non-zero elements.

 $t = .002N^3 + .004N^2$ For one solution of a set of N simultaneous equations having $N^2 + N$ non-zero elements.

Computing times for matrices having a predominance of zero elements is sharply reduced.

MATRIX TRANSPOSITION ROUTINE

The Transposition Routine transposes the matrix designated by the INTERMAP Instruction as the operand. The transposition takes place within the designated accumulator. The matrix need not be a square matrix, however, the transposition will be performed on a square matrix having as its dimensions the largest dimension of the rectangular operand.

This operation may also be performed in a "Transpose and Negate" mode.

MATRIX MULTIPLICATION ROUTINE (COMPREHENSIVE)

The comprehensive Multiplication Routine multiplies matrix [A], the pre-multiplier or operand [A], by the matrix [B], the post-multiplier or operator (SRC), and transfers the product to the Destination (DST).

Both the post-multiplier and the product may be any accumulator, magnetic tape or punched tape. The number of accumulators used in an instruction imposes obvious limitations on the order of the product

matrix. Product matrices for which $N \le 54$ and $M \le 54$ may be produced on tape.

This multiplication routine operates in four modes. The Normal Mode is the one described above. In the Scalar Mode the operand contains a scalar multiplier. In the Pre- and Post-diagonal Modes, the operand or the operator respectively contain a diagonal matrix. The NxN diagonal matrix is stored as an Nxl column vector and may be described as the vertical projection of the N^{th} order diagonal matrix. This arrangement permits considerable savings in storage space as well as computing time over the multiplication by a diagonal matrix in the Normal Mode.

MATRIX MULTIPLICATION ROUTINE (SHORT)

Compared to the comprehensive multiplication routine, this one will operate only in the normal mode. It has no facilities for scalar, preor post-diagonal multiplication, or those operations in which the post-multiplier and the product or the operand and the product are in the same accumulator. For all other operations, which are the majority of all the possible cases, in the Normal Mode the short multiplication routine has an advantage in computing time, search time and handling ease.

MATRIX ADDITION ROUTINE

The Addition Routine adds to the operand in "A" of the INTERMAP Instruction, the operator from the source (SRC) and stores the sum of the two matrices in the destination (DST).

Both the source and the destination may be an accumulator, magnetic or punched tape. It follows that since only the operand has to be stored in the G15D memory, the highest order of matrices which may be added to each other is one that would occupy the entire G15D memory.

The Addition Routine operates in the following four modes: $+[A] \pm [B] = [C]$ where [A] is the operand, [B] is the operator (SRC), and [C] is the destination (DST).

"DOT PRODUCT" MULTIPLICATION ROUTINE

"Dot Product" multiplication is not a proper algebraic matrix routine. It follows precisely the same rules as the Matrix Addition Routine (Op-Code . 23) and is a useful device to "extract" any part of a matrix.

OPERATION LEVELS AND MODES

INTERMAP Instructions are entered and executed under the control of the INTERMAP P. P. R (OP = .00). They may be executed on two instruction levels:

The Single Instruction Level, and the Multiple Instruction Level.

On the Single Instruction Level, the control is returned to INTERMAP P. P. R. following the execution of each INTERMAP Instruction.

On the Multiple Instruction Level, a number of INTERMAP Instructions will be executed sequentially. Such a programmed series of INTERMAP instructions is called an INTERMAP Sequence. Each INTERMAP Sequence must be terminated by a programmed return to I. P. P. R.

Only the service type INTERMAP Operation Routines may be operated in both the manual or the automatic mode.

OP-Code	Service Type Operation Routines
. 00	INTERMAP P. P. R
. 01	Accumulator Selection Routine
. 02	Typewriter Input/Output Routine
. 03	Punched Card Input/Output Routine

Manual or automatic operation is independent of the instruction level.

MANUAL MODE

Following an INTERMAP instruction which initiates the manual mode, the program will halt with the first type-in gated. Any number of manual instructions relative to an Operation Routine may now be typed in and executed. After each Manual Instruction the program returns to the proper type in.

To return from the manual to the automatic mode, type: sc3f, and automatic control will be resumed on the same instruction level which prevailed before initiating the manual mode. If this was the Single Instruction Level, the program returns to I.P.P.R. If it was the Multiple Instruction Level, the program proceeds to execute the next instruction in the INTERMAP Sequence.

AUTOMATIC MODE

If an INTERMAP Instruction is coded to operate in the automatic mode, then following the execution of the instruction operations will continue on the same instruction level which prevailed before execution, i.e., a return to I.P.P.R if on the Single Instruction Level, and an advance to the next instruction in the INTERMAP Sequence if on the Multiple Instruction Level.

Instruction levels may be changed only under the control of INTERMAP P. P. R.

COMPATIBILITY WITH OTHER SYSTEMS

GENERAL

INTERMAP is fully compatible with INTERCOM 1000 Double Precision. Output tapes produced with either system may be read by both systems. Similarly, off line prepared decimal data tape may be read by both systems. Information which is generated by either of the systems and stored in the memory may be picked up and used by the other.

A special INTERCOM subroutine to read and punch INTERMAP Tape Labels will be available at a later date.

All data and program storage on magnetic tape is in accordance with the rules and format of the Magnetic Tape Service Routine, as stated below.

INTERMAP is not compatible with MAP-29. The rules for converting MAP-29 to INTERMAP data tapes are outlined below.

APPLICATIONS PROJECT 61--MAGNETIC TAPE SERVICE ROUTINE

The only noncompatibility between INTERMAP and the M. T. S. R is that the INTERMAP serial number line does not contain a block searching routine (Reference A. P. 61-MTSR, Part III, 6).

Following the storage of data (a matrix) on magnetic tape, a "next serial number" is always written. Even though it does not contain the block searching routine, it may be found by the method described in A. P. Gl-M. T. S. R., Part II, 2, Record, c(i).

Following the storage of the INTERMAP Program itself on magnetic tape, see "Program to Magnetic Tape Loading Routine" page 7. l. l, a fully compatible serial number line which does have the block searching routine is written on tape, and its serial number is typed out as part of the INTERMAP program to magnetic tape loading procedure.

APPLICATIONS PROJECT 43-MAP 29

MAP 29 data tapes may be converted to INTERMAP format if the following differences are corrected:

1. Reference is made to A. P. 43 - MAP 29, Issue II of 1 February 1959, page 26 for a detailed description of a MAP 29 data tape.

The arrangement of an INTERMAP data tape on the other hand is always as shown below:

Line 19: 00,01 02,03 04,05 06,07 u6, u7

Element: a_{1,j} a_{2,j} a_{3,j} a_{4,j} a_{54,j}

All zero-elements following the last non-zero element in a block are absolute zeros.

2. An INTERMAP data tape label has to be produced to precede the converted tape.

Data stored by the two systems is interchangeable if:

- 1. In INTERMAP terms N=Nmax, see AP43-MAP 29, Issue II, page 8, Figure 1.
- 2. MAP 29 is modified in accordance with Supplement No. 1 to AP 43, dated 25 May 1959, so that the address of $a_{1,1}$ is not lower than 03. u4, u5.

LABELS AND DATA TAPES

All INTERMAP data tapes consist of a Label followed by M blocks of data tape. All Operation Routines which permit data tape input perform the Compatibility Test and the Validity Test based on the dimensions of the matrix, N and M, on tape as indicated by the contents of the Label. The number of data blocks which an Operation Routine attempts to read is equal to M, the horizontal dimensions of the matrix.

The three types of Labels are as follows:

THE BINARY LABEL

The Binary Label is a four word block produced by INTERMAP preceding any data tape output onto punched tape or magnetic tape. It is used by all Operation Routines which permit a data Tape Source.

THE DECIMAL LABEL

The Decimal Label is also a four word block which may be prepared on off-line equipment. It may be used on data tapes which have been produced without a label by other G15D programs. The Decimal Label may only be used on the store and transfer Routine, OP=20.

Both the Binary and the Decimal Labels are similar in format and are shown in Table 1.3.1.

Word	Binary Label	Decimal Label
.00	.0000000 or .Z000000	.0000000 or .Z000000
. 01	(01) = 0 .0000000	(01) ≠ 0 . Z000000
. 02	2N . 2 ⁻¹² .0180000	N. 10 ⁻³ .0120000
. 03	M. 2 ⁻¹² .00x0000	M. 10 ⁻³ .0130000

Table 1.3.1 - Binary and Decimal Labels for N = 12 and M = 13.

THE FLAGGED LABEL

A Label in which (00) \neq 0 is a Flagged Label. Except for off line preparation only the Store and Transfer Routine (OP = 20) produces a Flagged label, see Section 5.6 on Forced Transposition and Table 5.6.1.

The effect of the Flagged Label is a Forced Transposition in reverse. M blocks of tapes containing N elements each are read, but instead of storing a matrix of M columns and N rows the Destination will receive a Matrix having M rows and N columns. This forced Transposition is reflected in the new Label if the Destination is tape or in the Accumulator Cell if the Destination is an accumulator.

Below is a list of Operation Routines which permit a data tape source:

<u>OP</u>	Operation Routine	Accepts Decimal Label?	Accepts Flagged Label?
. 10	Dot Product Mult. Routine	No	No
. 20	Store and Transfer Routine	Yes	Yes
. 22	Matrix Multiplication (Short)	No	No
. 23	Matrix Addition	No	No
. 25	Matrix Multiplication (Comp.)	No	No

At times it may be required to prepare a Binary or a Decimal Label on the Flexowriter, and the following format must be used:

DDD00000(tab)DDD0000(tab)D000000(Tab)D000000(Tab) (stop 2)

THE FLEXOWRITER LABEL

The Flexowriter Label may be used only with the Decimal Data Tape Input Routine (OP = 26), and always precedes blocks of double precision decimal data which has been prepared in accordance with instructions outlined on page 20 and 21 of the INTERCOM 1000 Operating Manual. This Label should be prepared in the following manner:

- (tab) 53. MMM000000000(tab) (tab) 53. NNN000000000(CR)
- (tab) 50. D00000000000 (tab) (tab) wwwwwwwwwwwww(tab) (Stop 2)

Note that the excess-fifty exponents are not affected by the magnitude of M and N. If the matrix is to be transposed on input then $D \neq 0$, otherwise D = 0.

For example, the Flexowriter Label for a 12×13 matrix which is to be transposed on input is:

- (tab) 53.013000000000 (tab) (tab) 53.01200000000(CR)
- (tab) 50. 100000000000 (tab) (tab) wwwwwwwwwwwww (tab) (Stop 2)

In the INTERMAP System each block of data tape consists of one matrix column. Data may be either decimal or binary and is fully compatible in both forms with INTERCOM 1000 Double Precision.

BINARY DATA TAPES

The input and output punched paper tapes of INTERCOM 1000 D. P. and INTERMAP are interchangeable with regard to the number format.

INTERMAP requires that the elements of each column be arranged in the following sequence after reading a data block into the G15D input-output buffer:

All INTERMAP Output Tapes contain absolute double precision zeroes instead of floating point zeros (50.00000000000). This makes some INTERMAP data tapes considerably shorter since no tape is produced for zero elements following the last non-zero element in a column.

Data on magnetic tape is not interchangeable because INTERMAP and INTERCOM 1000 are not using the same magnetic tape serial number system.

FLEXOWRITER DATA TAPE

Flexowriter data tape is prepared in accordance with the instructions on page 20 and 21 of the INTERCOM 1000 Operating Manual. For each column A_1 , j is to be typed first followed by A_2 , j ---- to A_n , y. The Flexowriter Label and all but the last data block should have the trailing W - codes. The last data block should contain the Z codes.

THE FILE NUMBER SYSTEM

The INTERMAP File Number System permits the writing of general purpose INTERMAP Sequences in which File Numbers, expressed as a Source or Destination in an INTERMAP Instruction, do not necessarily correspond to the Serial Numbers actually written on a magnetic tape.

The following are the definitions of terms used in this manual:

Serial Number = SR = The number written on tape

File Number = FN = The number represented by a Source or Destination

First Available File Number = FAN

First File Number = FFN = FAN - 1

(1) Generally: SR = FFN + FN

The number typed in for magnetic data tape (Section 4.1, par. 3) or for program use (Section 4.9, par. 2 and 4) is the First Available File Number, or FAN. However, when computing the serial number on data tape, corresponding to any File Number in a Source or Destination, formula (1) above must be used. This is demonstrated in a sample INTERMAP Sequence in Section 5.5.

To compute serial numbers on program tape, when using the MTA-2 Version, refer to Section 7.1.

Whenever data is written on a magnetic tape, a new serial number is written following this data. If data was written following serial number SR, serial number block SR + 1 was written immediately following it. This means that all magnetic tapes used by INTERMAP are always ready to be used by any other program compatible with A. P. 61-M. T. S. R.

MEMORY ALLOCATION

PROGRAM

The program associated with each INTERMAP operation is stored and executed from command lines 00 to 02. During some operations, line 19 may also be used as a command line, if not used as buffer storage or for input-output.

CONTROL

The control portion of INTERMAP is the only permanently stored portion of the program, and occupies line 03 from word .00 to .93.

ACCUMULATOR SPECIFICATIONS AND A $_{ m MAX}$ - TEST

A maximum of seven accumulators may be in use at any one time. The specifications of each accumulator with regard to its location in the memory, its size and shape are stored in a four word group, called Accumulator Cells, starting from (03.u0, u1, u2, u3) for accumulator No. 1 to (04.16, 17, 18, 19) for accumulator No. 7.

The maximum number of accumulators which may be used in an INTER-MAP sequence is called A_{max} . A_{max} may be set manually, or under program control using the Accumulator Selection Routine (op-code .01),

where

$1 \le A_{max} \le 7$

Table 21-1 shows, that as fewer accumulators are used, or as A_{max} becomes smaller more data storage space is available. All INTERMAP Operation Routines test each accumulator number in an INTERMAP Instruction against A_{max} and if such an accumulator number is larger than A_{max} , the operation is terminated by an Error Halt before any information in the memory may be inadvertently destroyed. The Accumulator Selection Routine will reject a First or Last Location (F. L. or L. L.) smaller than the one shown in the column "First Location for Data" for a respective value of A_{max} .

	Specifications	IF Acc. = A _{max}	
Acc. No.	Stored in:	First Location for Data	Acc. Available
1	03.u0u3	03.u4, u5	1
2	03.u4u7	04.00,01	1 to 2
3	04.0003	04.04,05	1 to 3
4	04.0407	04.08,09	l to 4
5	04.0811	04.12,13	1 to 5
6	04.1215	04.16,17	1 to 6
7	04.1619	04.20,21	1 to 7

Table 2.1.1 - Accumulator Cells

INTERMAP INSTRUCTIONS

Intermap instructions are stored in line 18. Instructions are numbered from .00≤ab≤u7 and are stored in location (.u7 - .ab) of line 18. Thus, for an INTERMAP program using less than 108 instructions, a portion of line 18 from 18.00 up may remain unused.

DATA STORAGE

Data storage in the form of double precision floating point numbers, is between locations 04.20, 21 and 17. u6, u7, thus permitting the minimum storage of 746 elements.

If however, less than seven accumulators are used or the INTERMAP program in line 18 consists of fewer than 108 instructions, additional data storage space will be made available in parts of lines .03, .04 and .18.

The maximum number of storage locations is available if:

- 1. only one accumulator (No. 1) is used, and
- 2. no INTERMAP instructions are stored in line 18.

In this case, the available storage space is between 03.u4,u5 and 18.u6,u7, permitting the storage of a maximum of 812 elements which are the equivalent of a 28 x 29 matrix. Instructions may then be executed from the keyboard on the single Instruction Level.

Figure 2.1.2 is a graphic presentation of the memory allocation.

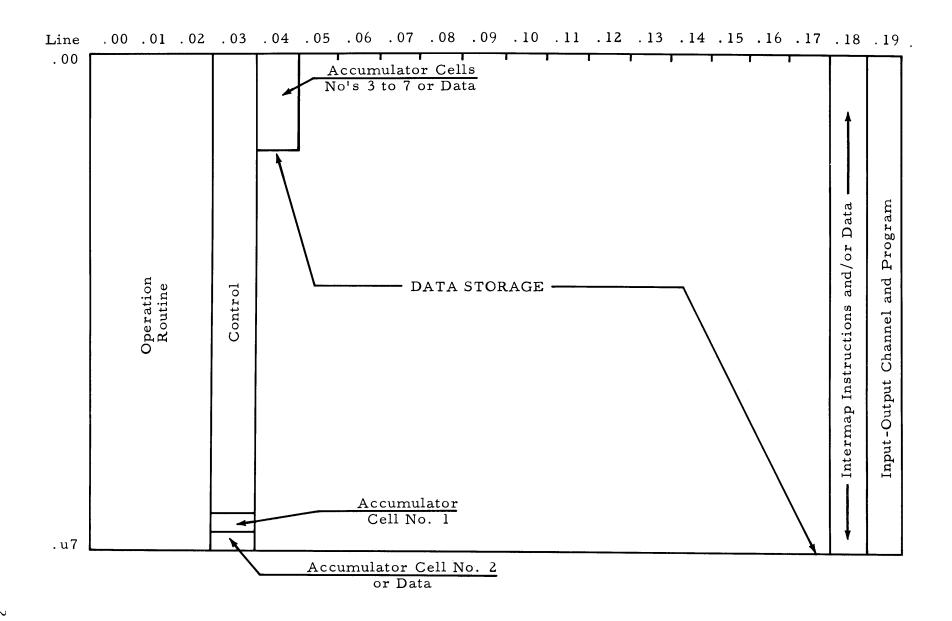


Figure 2.1.2 - G15D Memory Allocation for INTERMAP Use

Each element ai, j occupies a double precision word and is addressed by the address of its least significant (or the even) word.

In the INTERMAP addressing system the address of each element on the drum is defined by one unique number, thus:

The address of an element stored in 10.06,07 is:

$$10 \times 108 + = 1086$$

Matrices are stored by columns, so that element $a_{1,j}$ is followed by $a_{2,j}$, etc. Figure 2.2.1 shows the locations on the drum occupied by a 3 x 4 matrix. Note, that the vertical difference between addresses is two and the horizontal difference is six, or 2N.

In Figure 2.2.2 the same matrix is now stored in a differently shaped accumulator. The vertical difference between addresses is still two, however, the horizontal difference is now 10. Obviously, the 3 x 4 matrix does not fill this accumulator which could hold any matrix for which

$$1 \le N \le 5$$
 , and

$$1 \le M \le 6$$

It follows that:

1. If the largest vertical dimension of an accumulator is N max, then the horizontal or column increment in the direction of ascending column number is

$$D_j = 2 Nmax$$

2. In the direction of ascending row numbers the vertical increment is a constant, namely

$$D_i = 2$$

3. For an accumulator with a constant 2 Nmax the address of any element $a_{i,j}$ will be the same regardless of the order N and M of the matrix.

Let: $/a_{ij}$ / be equivalent to "The address of a_{ij} "

F.L. = First Location of an accumulator

L.L. = Last Location of an accumulator

N and M = dimensions of a matrix

 N_{max} and M_{max} = dimensions of an accumulator

$$N \le N \max$$
 and $M \le M \max$, and $1 \le i \le N$ and $1 \le j \le M$

Then the following can be defined mathematically:

Generally

$$/a_{ij}/ = F.L. + 2Nmax(j-1) + 2(i-1)$$
 (1)

When i=j=1, then

$$/a_{11}/ = F.L.$$
 (2)

When i=N and j=M, then

$$/a_{N,M}/ = F.L. + 2Nmax(M-1) + 2(N-1)$$
 (3)

When N=Nmax and M=Mmax

$$/a_{NM}/ = L.L. = F.L. + 2Nmax(Mmax) - 2$$
 (4)

It should be noted that the address of element all of a matrix is always equal to the First Location of its accumulator. However, only if

$$N = Nmax$$
 and $M = Mmax$

does the location of the last element a_{NM} of a matrix coincide with the Last Location (L. L.) of its accumulator.

The First Location of an accumulator is like a reference mark on a datum line. The location of an accumulator in the memory may be shifted by changing its First Location or $/a_{11}/.$

In order to generate any address $a_{i,\,j}$ it is necessary to keep two counters or registers, one for the rows and one for the columns.

The row counter is Bin and

The column counter is B_j^m

These counters assume their initial values when n = 1 and m = 1.

By definition then:

$$B_{i}^{1} = -2N$$

$$B_{i}^{1} = -2N_{\max}(M)$$

As mentioned above, the row increment $D_{\rm i}$ = 2 and the column increment $D_{\rm j}$ = $2N_{\rm max}.$

j =	1	2	3	4
i = 1	A _{1, 1} (10.06,07)	A _{1, 2} (10.12, 13)	A _{1,3} (10.18,19)	A _{1,4} (10.24,25)
2	A _{2,1} (10.08,09)	A _{2, 2} (10.14, 15)	A _{2, 3} (10.20, 21)	A _{2, 4} (10. 26, 27)
3	A _{3, 1} (10.10,11)	A _{3, 2} (10. 16, 17)	A ₃ , ₃ (10. 22, 23)	A _{3, 4} (10. 28, 29)

Figure 2.2.1 - A 3 x 4 Matrix in a 3 x 4 Accumulator

j :	= 1	2	3	4	5	6
i = 1	A _{1,1} (10.06,07)	A _{1, 2} (10.16,17)	A _{1,3} (10.26,27)	A _{1, 4} (10. 36, 37)	(10.46,47)	(10. 56, 57)
2	A _{2, 1} (10.08,09)	A _{2, 2} (10.18, 19)	A _{2, 3} (10. 28, 29)	A _{2, 4} (10. 38, 39)	(10.48,49)	(10.58,59)
3	A _{3, 1} (10.10,11)	A _{3, 2} (10. 20, 21)	A _{3, 3} (10. 30, 31)	A _{3, 4} (10.40,41)	(10.50,51)	(10.60,61)
4	(10.12, 13)	(10.22, 23)	(10.32,33)	(10.42,43)	(10.52,53)	(10.62,63)
5	(10.14, 15)	(10.24, 25)	(10.34,35)	(10.44, 45)	(10.54,55)	(10.64,65)

Figure 2.2.2 - A 3 x 4 Matrix in a 5 x 6 Accumulator 2Nmax = 10.

The following table will illustrate the settings of the row and column counters for various values of i and j.

i & j	Вi	Вј
1	- 2N	- 2N _{max} M
2	- 2N + 2	- $2N_{max}M + 2N_{max}$
3	- 2N + 4	- $2N_{max}M + 4N_{max}$
4	- 2N + 6	- $2N_{max}M + 6N_{max}$
N or M	-2N + 2(N-1) = -2	$-2N_{\max}M + 2(M-1)N_{\max} = -2N_{\max}$
$\left[\begin{array}{c} N+1\\ \text{or}\\ M+1\end{array}\right]$	æ	æ

Table 2.2.3 - In general for any given value of i and j,

where

$$B_i^i = -2N + 2(i-1)$$
 = $-2[N - (i-1)]$ (5)

$$B_{j}^{j} = -2N_{max}M + 2N_{max}(j-1) = -2N_{max}[M-(j-1)]$$
 (6)

Having the index registers arranged as shown, there is no need to store a limit, because

$$B_i^{N+1} = 0 \text{ and } B_i^{M+1} = 0$$

and a zero-test indicates when the limit has been reached.

To facilitate generating the address of any element a_{ij} , it has been found convenient to make the following definitions:

$$Q = F.L. + 2NmaxM + 2N, so that$$
 (7)

$$/a_{ij}/=Q+B_{i}^{i}+B_{i}^{j}$$
 (8)

Reference to Figure 2.2.2 will show that $Q = \frac{a_{N+1}}{M+1}$.

The following general example demonstrates the validity of Equations (7) and (8) above.

(7)
$$Q = F.L. + 2NmaxM + 2N$$

(6) $+ B_{j}^{j} = -2NmaxM + 2Nmax(j-1)$
(5) $+ B_{i}^{i} = -2N + 2(i-1)$
(1) & (8) $/a_{ij}/ = F.L. + 2Nmax(j-1) + 2(i-1)$

Subroutines which compute Q, B_i^i and B_j^j have been standardized and made part of the INTERMAP controls. Once $/a_{ij}/$ has been obtained, it is used as the input into a "Read and Write Subroutine". This subroutine may either "read" the element a_{ij} into an arithmetic register or "write" the contents of this arithmetic register into $/a_{ij}/$.

ACCUMULATORS

An accumulator consists of a group of double precision storage locations, or elements, in which each element may be addressed individually using such constants to generate the addresses as are stored in the Accumulator Cell of a particular accumulator.

The Accumulator Cells for any one of the seven accumulators consist of four words, which contain:

- 1. Nmax
- 2. First Location and Last Location
- 3. N
- 4. M

N and M do not refer to the size of the accumulator but to the order of the matrix contained in the accumulator. 2Nmax and F.L. and L.L. determine the size and shape of the accumulator.

INTERMAP Accumulators may be classified as:

- 1. Solid Accumulators
- 2. Staggered Accumulators

There is no difference between the cells of the two classifications.

An accumulator may be considered "solid" if during a specific operation it is never expected to contain valid data from more than one matrix. Any other accumulator is staggered.

The following examples will illustrate these two classifications.

Both Figure 2.2.1 and 2.2.2 represent Solid Accumulators. Note in Figure 2.2.2, that a Solid Accumulator does not have to be solidly full. The 5 x 6 accumulator could accommodate a matrix of:

$$1 \le N \le 5$$
, and

1≤M≤6

where for any order of N and M the address of $a_{i,j}$ is the same for a given 2Nmax.

Figure 2.3.1 is an example of "staggered" accumulators. The accumulator shown in (a) containing a 4×4 matrix would seem to be solid. However, it can be seen that (b) which occupies precisely the same space in the memory consists of four 2×2 accumulators. It follows that:

$$\begin{bmatrix} A \end{bmatrix} = \begin{bmatrix} B & D \\ C & E \end{bmatrix}$$

a ₁₁ (12. 20, 21)	^a 12	^a 13	^a 14
	(12.28,29)	(12.36,37)	(12.44,45)
a ₂₁ (12. 22, 23)	a ₂₂ (12.30,31)	a ₂₃ (12.38,39)	^a 24 (12.46,47)
^a 31	a ₃₂ (12. 32, 33)	a33	^a 34
(12. 24, 25)		(12.40,41)	(12.48,49)
^a 41	^a 42	a ₄₃ (12.42, 43)	^a 44
(12.26,27)	(12.34,35)		(12.50,51)

a. One 4 x 4 Accumulators

			_
b ₁₁ (12.20,21)	^b 12 (12.28,29)	^d 11 (12.36,37)	^d 12 (12.44,45)
^b 21 (12. 22, 23)	^b 22 (12.30,31)	d ₂₁ (12.38, 39)	d ₂₂ (12.46,47
c ₁₁ (12. 24, 25)	c ₁₂ (12.32, 33)	e ₁₁ (12.40,41)	e ₁₂ (12. 48, 49)
^c 21 (12. 26, 27)	^c 22 (12.34,35)	e ₁₂	e ₂₂ (12.50,51)

b. Four 2 x 2 Accumulators

Figure 2.3.1 - Example of Staggered Accumulators

If it is assumed for example that

$$[A] = (Acc. 1)$$

$$[B] = (Acc. 2)$$

$$[C] = (Acc. 3)$$

$$[D] = (Acc. 4)$$

$$[E] = (Acc. 5)$$

then each one of these five matrices can be addressed individually.

The Accumulator Cells for such an arrangement would be set as follows:

Acc. No.	Nmax	F.L. & L.L.	N	M
1	4	12.20 12.50	4	4
2	4	12.20 12.30	2	2
3	4	12.24 12.34	2	2
4	4	12.36 12.46	2	2
5	4	12.40 12.50	2	2

Staggered accumulators permit the partitioning of matrices into smaller ones or the composition of larger matrices from smaller parts. Staggered accumulators will only function properly if 2Nmax for all the staggered accumulators is the same.

Accumulators may also be overlapping. This situation may arise in too many different ways to make a definition possible. In general it must be the programmer's responsibility to avoid the overlapping of accumulators in regions where significant data is stored.

Some of the possible accumulator arrangements are illustrated in Figure 2.3.2.

- 1. A combination of stagger and overlap
- 2. All staggered accumulators
- 3. All overlapping accumulators terminating in the same Last Location
- 4. All solid accumulators.

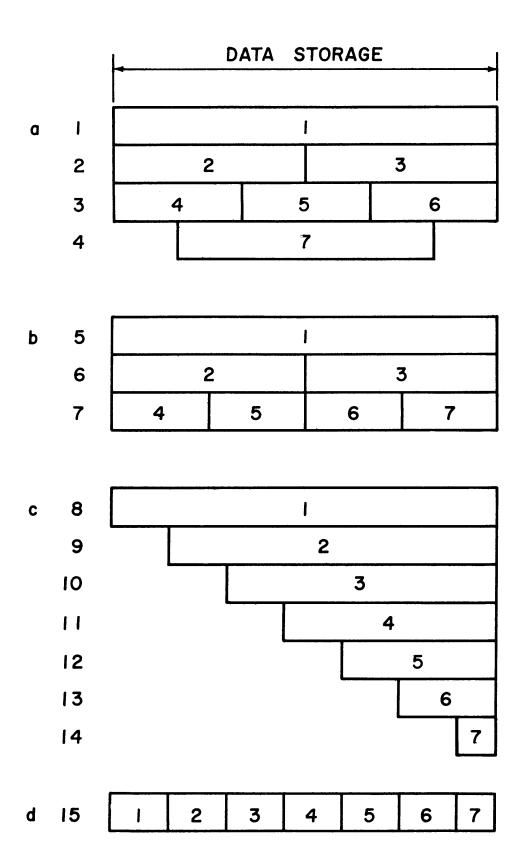


Figure 2.3.2 - Accumulator Arrangements

If the order of the matrices is generally in the 10×10 range, the arrangement shown in (d) may be the easiest to work with. Of course, a similar arrangement may be used for fewer than seven accumulators, thus permitting larger matrices. Rows 2, 3 or 7 in (a) and (b) are representative of such an arrangement. Operating with solid accumulators may be uneconomical space wise, but will otherwise provide for the greatest programming ease.

COMPATABILITY TESTS

Whenever the possibility exists of inadvertently destroying valid data, all INTERMAP Operation Routines contain Compatibility Tests, the purpose of which is to avoid such accidental destruction.

The Amax-Test described in Section 2.1 is one of these Compatibility Tests. Another Compatibility Test is performed for each operation during which an accumulator is made the destination of data (a matrix). The purpose of the test is to ensure that the Destination accumulator can hold the Source data in its entirety without:

losing any of the source data, or changing any data outside the boundaries of the destination accumulator.

Given that a Source matrix [A] is to be transferred to a Destination accumulator, and the dimensions of the matrix at the source are:

 N_S = number of rows of source matrix

 M_S = number of columns of source matrix

The dimensions of the Destination Accumulator are:

F.L._D = First storage location in the destination accumulator and also the address of a₁, ₁.

L.L._D = The last available storage location in the destination accumulator.

L.L._S = The address of $a_{N,M}$ of the source matrix when stored at the destination.

2Nmax(D) = The column or horizontal increment of the destination accumulator.

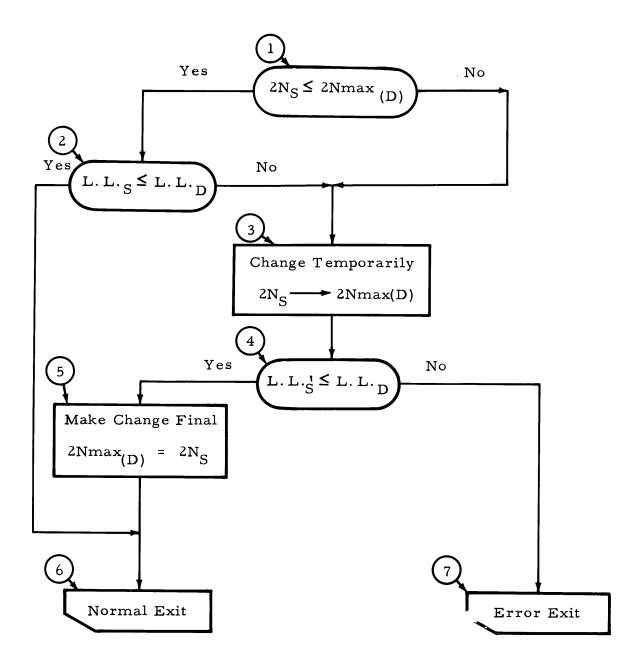
Naturally, in order not to destroy valid data, the following has to be satisfied:

$$L.L._{S} \le L.L._{D}$$

It has been shown above that while the F.L. and the L.L. of an accumulator determine its size, its shape is determined by 2Nmax. Comparing Figure 2.4.1 with Figure 2.2.2 it can be seen that both accumulators have the same First and Last Location, the same size and contain the same order of matrix. However, in their present form the first could not hold a 6 x 3 matrix while the second could not hold a 4 x 5 matrix.

, j =	= 1	2	3	4
i = 1	A ₁₁	A ₁₂	A ₁₃	A ₁₄
	(10.06,07)	(10.24, 25)	(10.42,43)	(10.60,61)
2	A ₂₁	A ₂₂	A ₂₃	A ₂₄
	(10.08,09)	(10.26,27)	(10.44,45)	(10.62,63)
3	A ₃₁	A ₃₂	A ₃₃	A ₃₄
	(10.10,11)	(10.28,29)	(10.46,47)	(10.64,65)
4				
	(10.12, 13)	(10.30,31)	(10.48,49)	
5				
	(10.14, 15)	(10.32,33)	(10.50,51)	
6				
	(10.16,17)	(10.34,35)	(10.52,53)	
7				
	(10.18,19)	(10.36,37)	(10.54, 55)	
8				
	(10.20,21)	(10.38.39)	(10.56,57)	
9	(10 22 22)	(10 40 41)	(10 50 50)	
	(10.22,23)	(10.40,41)	(10.58,59)	

Figure 2.4.1 A 3 x 4 Matrix in a Non-rectangular Accumulator



Where:
$$L.L._S = F.L._D + 2Nmax_{(D)}(M_S-1) + 2N_S - 2$$

 $L.L._S' = F.L._D + 2N_S(M_S-1) + 2N_S - 2 = F.L._D + 2N_SM_S - 2$

Figure 2.4.2 Compatibility Test - Logical Model

It follows that a transfer of a matrix from a Source to a Destination may necessitate a change of 2Nmax or a change of the shape of the accumulator. As mentioned above, a necessary characteristic of staggered accumulators is that their 2Nmax has to be the same. A change of 2Nmax may therefore disrupt the relationship between staggered accumulators if such existed.

The general rule followed by the compatibility tests is as follows:

Assume that the accumulators are staggered and that $2Nmax_{(D)}$ should not be disturbed. First try to fit the source matrix between the First L Location and Last Location of the destination accumulator without changing $2Nmax_{(D)}$. If this fails, the test is repeated for:

$$2N_S \longrightarrow 2Nmax_{(D)}$$

If the test succeeds this time, the above change is made permanent and the operation proceeds. Thus a staggered accumulator is assumed initially and retained only if the following conditions are satisfied:

$$L.L._S \le L.L._D$$
 and
 $L.L._S = F.L._D + 2Nmax_{(D)}(M_S-1) + 2N_S - 2$

If the second test succeeds, $2N_S$ is substituted for $2Nmax_{(D)}$ and in this case:

$$L.L._{S}^{1} \le L.L._{D}$$
 and
 $L.L._{S}^{1} = L.L._{D} + 2N_{S}(M_{S}-1) + 2N_{S} - 2 = FL_{D} + 2N_{S}M_{S} - 2$

Should the second test fail as well, then the destination accumulator is not compatible with the source data and the operation is terminated by an Error Halt.

Figure 2.4.2 is a logical model of a Compatibility Test.

Numbering each of the steps on the logical flow chart in Figure 2.4.2 from 1 to 7, it can be seen that there are five possible routes through this chart, namely:

Case I: 1, 3, 4, 5, 6

Case II: 1, 3, 4, 7

Case III: 1, 2, 6

Case IV: 1, 2, 3, 4, 7

Case V: 1, 2, 3, 4, 5, 6

These five cases are mathematically demonstrated in Figure 2.4.3.

					Case		
			I	II	III	IV	V
Source		2N	20	20	12	12	12
Sou		М	5	12	5	26	16
ion	()	Original	14	14	14	14	14
inat	2Nmax(D)	Temporary	20	20	/	12	12
Destination	2Nn	Final	20	14	14	14	12

GIVEN:

$$F.L._{D} = (04.20) = 452$$

Case	Step	Note
I	1 3 4 5 6	2N _S > 2Nmax(D) Change 2Nmax(D) temporarily L.L.s' = 452 + (20) (5) - 2 = 550 < L.L. _D Make change of 2Nmax(D) final Normal Exit
II	1 3 4 7	2N > 2Nmax(D) Change 2Nmax(D) temporarily L. L. 's = 452 + (20) (12) - 2 = 690 > L. L. Error Exit (No change)
III	1 2 6	2N < 2Nmax(D) $L.LS = 452 + (14)(4) + 12 - 2 = 518 < L.LD$ Normal Exit (no change)
IV	1 2 3 4 7	2N < 2Nmax(D) L.L. _S = 452 + (14) (25) + 12 - 2 = 812 > L.L. _D Change 2Nmax(D) temporarily L.L.' _S = 452 + (12) (26) - 2 = 762 > L.L. _D Error Exit (no change)
V	1 2 3 4 5 6	2N < 2Nmax(D) L. L. _S = 452 + (14) (15) + 12 - 2 = 672 > L. L. _D Change 2Nmax(D) temporarily L. L.' _S = 452 + (12) (16) - 2 = 642 < L. L. _D Make change of 2Nmax(D) final Normal Exit

Figure 2.4.3. Compatibility Test - Mathematical Model

Only after the compatibility test has been passed will the order of the destination accumulator be made equal to the order of the source matrix, or

$$2N_S \longrightarrow 2N_D$$
 and $M_S \longrightarrow M_D$

No change in any data stored in the memory is possible until and unless the compatibility test has been passed.

The Compatibility Test is also performed when the Source is not an accumulator. This includes:

Typewriter

Punched Card

Punched Tape

Magnetic Tape

For the typewriter input N_S and M_S are provided by a type-in. For punched card input N_S and M_S are contained in the First Card and for any tape input it is contained in the Label. For further information on the Label and First Card, refer to the section on INTERMAP Data.

If the Destination is not an accumulator only the "Amax--Test" is performed. However, for tape output the following test is performed.

$$2N_S \le 108$$

When $2N_S > 108$ an Error Halt occurs since it is impossible to produce a tape longer than 108 words long. For an exception to this rule, refer to the Store and Transfer Routine, section 5.6.

CODING INSTRUCTIONS

THE ALGEBRAIC INTERMAP INSTRUCTION

The algebraic INTERMAP Instruction is a three address code which consists of seven fields as shown in Table 3. 1. 1 below:

Field	1	2	3	4	5	6	7
	OP	С	A	T_S	SRC	T _D	DST
				Data	Source	Data D	estination
FUNCTION	Operation Code	Mode or Characteristic	Operand-Accumulator	Auxiliary Equipment	Operator	Auxiliary Equipment	Result

Table 3.1.1 - The Algebraic INTERMAP Instruction

The following is a detailed interpretation of the seven fields of an algebraic INTERMAP Instruction:

OPERATION

00≤OP≤63 = Op-code to call out Operation Routine 0≤C≤3 = One of four modes in which an Operation

Routine may be performed.

Routine may be performed

OPERAND

 $1 \le A \le 7$ = 1st address - the accumulator number of

the operand.

SOURCE

 $0 \le T_S \le 3$ = The number of one of three magnetic tape units or auxiliary photoreaders (if) used as a data source for the operator. If no accessory is used then $T_S = 0$.

0≤SRC≤127 = 2nd address - The data source or operator.

This may be an accumulator number, a file number on magnetic tape, or punched tape on a photoreader.

DESTINATION

 $0 \le T_D \le 3$ = The number of one of three magnetic tape units (if) used for the data destination or the results. If no accessory is used then $T_D = 0$

0≤DST≤127 = 3rd address - the data destination or result. This may be an accumulator number, a file number on magnetic tape, or the paper tape punch.

The following are the rules for writing algebraic INTERMAP Instructions:

OPERATION

Table 3.1.2 lists the algebraic op-codes and their functions in each of the four modes. For a detailed description of each mode, refer to the description of respective Operation Routines in Section 5.

OPERAND, OPERATOR AND RESULT

Table 3.1.3 relates the three addresses of the INTERMAP Instruction to the functions of the algebraic Operation Routines. Note that the flow of information is a logical one, namely: the Source operates on the Operand and the Result of the operation goes to the Destination. In Operation Routines like the Store and Transfer Routine, for instance, the operand is logically absent. Similarly, in the Matrix Inversion Routine, there is no Source or Destination. In general, where there is an Operator (Source), there must also be a Result (Destination) and vice versa.

Op-Code	Operation Routine	C=0	C= 1	C= 2	C=3
10	"Dot Product" Multiplication	A B	A. (-B)	(-A). B	(-A). (-B)
20	Store and Transfer	Clear and Add	Clear and Subtr.	Clear	Transpose
21	Matrix Inversion and Equation Solver	Invert	Solve		
22	Matrix Multipli- cation (short)	$A_1A_2=A_3$ $A_1A_1=A_2$	A ₁ A ₂ =T _{out} A ₁ A ₁ =T _{out}	A ₁ T _{in} =A ₂	A _l T _{in} = T _{out}
23	Matrix Addition	A + B	A - B	-A+B	-A-B
24	Matrix Transposition	Non-permissive: - Normal	Non-permissive: - Transpose & Negate	Permissive: - Normal	Permissive: - Transpose & Negate
25	Matrix Multipli- cation (comp.)	Normal	Scalar	Pre Diagonal	Post Diagonal
26	Decimal Data Tape Input Routine	Clear and Add Paper Tape Source			

Table 3. 1. 2 - Algebraic Operation Routines - Modes.

Op-Code	Operation Routine	Accumulator: - Operand	Source: - Operator	Destination: - Result
10	"Dot Product" Multiplication	(A)	(B)	(C)
20	Store and Transfer	none	(A)	(A)
21	Matrix Inversion and Equation Solver	(A)	none	none
22	Matrix Multiplication (short)	(A)	(B)	(A) (B)=(C)
23	Matrix Addition	(A)	(B)	(A)+(B)=(C)
24	Matrix Transposition	(A)	none	none
25	Matrix Multiplication (Comprehensive)	(A)	(B)	(A)(B)=(C)
26	Decimal Data Tape Input Routine	none	(A)	(A)

Table 3.1.3 - Algebraic Operation Routines -- The Use of the Three-Address Systems.

SOURCE AND DESTINATION

The following are the rules for determining the source and destination in an INTERMAP Instruction:

- l. When $T_S=0$ or $T_D=0$ then the source or the destination equipment respectively is part of the basic G15D computer.
- 2. When SRC=0 or DST=0 then the Source or the Destination are both on punched paper tape. The type of equipment then is a photoreader for the source and a paper punch for the destination. With reference to (a) above, if also $T_S=0$ the photoreader must be the basic G15D photoreader, and if also $T_D=0$ the paper tape punch too must be the one on the basic G15D.
- 3. When SRC \neq 0 and DST \neq 0 both Source and Destination may have different meanings based on the respective values of T_S and T_D .
 - a. If T_S=0 or T_D=0 then by (a) above, the Source or Destination must be on the basic G15D, and in this case they are accumulators in the G15D memory. Where 1≤ACC≤ Amax and Amax≤7, Any SRC>Amax or DST>Amax will be rejected by the Amax-Test.
 - b. If $T_S \neq 0$ or $T_D \neq 0$ then by (a) above the Source or Destination are not on the basic G15D. They are instead relative file numbers on magnetic tapes. It follows logically, that since all magnetic tape is auxiliary equipment and there is no "basic magnetic tape", there could be no magnetic tape source or destination when $T_S=0$ or $T_D=0$. This is not so for punched tape.
- 4. When $T_S > 0$ and $T_D > 0$ their number stands for one of three auxiliary photoreaders or magnetic tape units. There is only one paper tape punch and consequently for any destination punched tape, T_D and T_S must be equal to zero ($T_S = 0$, $T_D = 0$). $T_D \neq 0$ when DST=0 is not a valid instruction.
 - 5. When $T_S = T_D$ and SRC=DST $\neq 0$ Error Halt occurs.

The above rules are tabulated in Table 3.1.4.

THE NON-ALGEBRAIC INTERMAP INSTRUCTION

The non-algebraic INTERMAP Instruction performs automatically all the functions of INTERMAP which may be performed in a manual mode. These instructions are limited to the service type Operation Routines, which are listed below, and which are the only Operation Routines which respond to manual type-ins:

Op-Code	Operation Routine
.00	INTERMAP P. P. R.
.01	Accumulator Selection Routine

Op-Code	Operation Routine			
.02	Typewriter-Input/Output Routine			
. 03	Punched Card Input/Output Routine			

It follows that a non-algebraic INTERMAP Instruction may be written for every possible manual instruction which can be typed in. Coding non-algebraic instructions in effect means translating the manual type-ins, described under Operating Instructions elsewhere in this manual, in accordance with a definite set of rules.

In order to write a non-algebraic INTERMAP Instruction to perform automatically any service type operation, the programmer should first determine the equivalent manual type-ins and then translate these type-ins into an INTERMAP Instruction.

The above Operation Routines have two type-ins for manual instructions. The first type-in designates the accumulator number and the second type-in defines the operation. (Only the INTERMAP P. P. R has no equivalent of the first type-in.) These two type-ins are translated into an INTERMAP Instruction following these rules:

Coding Rules for Non-algebraic INTERMAP Instructions

OPERATION	
OP	No change from the algebraic instruction.
С	One of three modes.
OPERAND	
Α	The operand or accumulator number specified the first type-in.
	In case of INTERMAP P. P. R, OP=.00, which has no first type-in and does not operate on accumulators, make A=0.
T_S and T_D	$T_S + T_D + 9 = H$ where $9 \le H \le Z$. H is the first digit of the second type-in.
SOURCE and DESTINATION	SRC=. 0ab and DST=. 0cd where . ab and . cd are part of the second type-in H. ab. cd.

by

Type of	So	urce or In		Destination or Out		
Data Storage	Equipment	T _S	SRC	Equipment	$^{\mathrm{T}}\mathrm{_{D}}$	DST
Accumulator	G15D Memory	0	l≤A≤Amax	G15D Memory	0	l≤A≤Amax
Paper Tape	Photo Reader	0≤T _S ≤3	0	Paper Tape Punch	0	0
Mag. Tape	MT-A-2	1≤T _S ≤3	1≤SRC≤127	MTA-2 ₁	1≤T _D ≤3	1≤DST≤127

Table 3. 1. 4 - Algebraic Operation Routines - Source & Destination Table.

The above rules are summarized in Table 3.1.5.

$$T_S + T_D + 9 = H$$

H	9	u	v	w	x	у	z
T _S	0	1	2	3	3	3	3
T_{D}	0	0	0	0	1	2	3

C - Characteristic

С	Mode	Function			
0	Manual	Disregards the remainder of the INTERMAP			
1	Wilding	Instruction and proceeds to 1st type-in			
2	Automatic	Equivalent to a positive type-in: (+H abcd)			
3	ratomatic	Equivalent to a negative type-in: (-H abcd)			

SRC and DST

		2nd Type-in	
Op-Code	Service Type Operation Routine	Type-in followed by (tab) s	SRC & DST
00	INTERMAP PPR	H. ab.	SRC = .0ab DST = .000
01	Accumulator Selection Routine	H. NN. MM.	SRC = . 0NN DST = . 0MM
02	Typewriter Input-Output Routine	H. LL. TT.	SRC = .0LL DST = .0TT
03	Punched Card Input-Output Routine	н. п. јј	SRC = .0II DST = .0JJ

Table 3.1.5 - Conversion of Manual Type-ins into Non-Algebraic INTERMAP Instructions.

THE SPECIAL INTERMAP INSTRUCTION:

INTERMAP Instructions which are neither algebraic nor service type instructions are called Special INTERMAP Instructions.

Special instructions have the same decimal format as algebraic instructions, however the meaning of all fields but the first (OP-code) is different.

Special INTERMAP Instructions are used for the following Operation Routine:

.04 Incrementer

For a detailed description of the Incrementer and its special coding rules, refer to section 5.5.

INTERMAP CODING EXAMPLES

Algebraic INTERMAP Instructions

1. 21. 0. 1. 0. 000. 0. 000

Invert the matrix contained in accumulator No. 1.

2. 21. 1. 2. 0. 000. 0000

Solve the equations contained in accumulator No. 2.

3. 20. 1. 0. 1. 000. 2. 005

Store and transfer the source matrix from punched tape on auxiliary photoreader No. 1 to magnetic tape unit No. 2 following (relative) File Number 5. On transfer change sign of each element.

4. 20. 2. 0. 0. 005. 0. 005

Clear to zero every element in accumulator No. 5.

5. 20. 3. 0. 0. 001. 0. 000

Transpose the matrix in accumulator No. 1 onto punched tape. If the accumulator contains a matrix of N rows and M columns, a tape will be produced consisting of N blocks (columns) having M double precision words (rows) each.

6. 25. 0. 1. 1. 002. 2. 001

Perform a multiplication, where the premultiplier is the matrix in accumulator No. 1, the post-multiplier the matrix written on magnetic tape No. 1 following (relative) File Number 2. Store the product matrix on magnetic tape No. 2 following (relative) File Number 1. This operation can be categorized as: $A_1 T_{\rm IN}^{-1} T_{\rm OUT}^{-1}$

7. 25. 1. 2. 0. 003. 0. 003

Multiply the matrix in accumulator No. 3 by the scalar which is the element all in accumulator number 2 and store the product in accumulator No. 3.

8. 25. 2. 2. 0. 003. 0. 000

Pre-multiply the matrix in accumulator No. 3 by the diagonal matrix in accumulator No. 2 and punch the product matrix on tape. The diagonal matrix in accumulator No. 2 consists only of one column and may be thought of as the vertical projection of a diagonal matrix.

9. 23. 0. 1. 0. 002. 0. 003

Add the matrix in accumulator No. 1 to the matrix in accumulator No. 2 and store the sum in accumulator No. 3.

10. 24. 1. 1. 00000000

Transpose and negate the matrix in accumulator number 1.

11. 10. 0. 1. 0002. 0001

Dot product multiplication. If the matrix in accumulator No. 2 is a null matrix, the effect of this instruction is to clear accumulator No. 1.

If the matrix in accumulator No. 1 is a unit matrix (only one's on the diagonal), only the main diagonal will be retained in accumulator No. 1 and all other elements will be zero.

NON-ALGEBRAIC INTERMAP INSTRUCTIONS

1. INTERMAP P. P. R:

-v03(tab)s 00.3.0.2.003.0000

Start computing on the Multiple Instruction Level from instruction number .03. This may be used as an unconditional transfer.

-w05(tab)s 00.3.0.3.005.0.000

Read on INTERMAP Sequence tape into line 18 and start/or continue computing on the Multiple Instruction Level from instruction .05.

-y09(tab)s 00. 1. 0. 3. 009. 2. 000

Type-out as an INTERMAP Instruction the contents of . 09, if any, and be ready to accept new instruction for storage in the same location.

NOTE: INTERMAP P. P. R does not operate on accumulators and therefore the Operand in these instructions is equal to zero.

2. Typewriter Input/Output Routine

a. 2(tab)s z0101(tab)s 02. 2. 2. 3. 001. 3. 001

Type-out in decimal the contents of accumulator number 2 by columns for all elements for which

 $0.1 \le ii \le N$

 $01 \le jj \le M$

b. 2(tab)s z0201(tab)s 02. 2. 2. 3. 002. 3. 001

same as above, except: 02 ≤ ii ≤ N

 $01 \le jj \le M$

In this case, all but the first row is typed out.

c. 3(tab)s -y0105(tab)s 02.3.3.3.001.2.005

permit type-in of one column of data such that

01≤ii≤N

jj = 05

which in this case is all of column 5.

d. 4(tab)s -w0104(tab)s 02.3.4.3.001.0.001

Clear accumulator Number 4 beginning with column 4.

01 ≤ ii ≤ N

04 ≤ jj ≤ M

3. Punched Card Input-Output Routine

l(tab)s -z0101

03. 3. 1. 3. 001. 3. 001

Read one deck of cards (matrix) and store in accumulator No. 1 where:

 $01 \le ii \le N$

 $01 \le jj \le M$

4. Accumulator Selection Routine

l(tab)s

y0910(tab)s Clear Acc. Cell No. 1, then

.01.2.1.3.009.2.010

set N = 9 and M = 10.

-w. 04. 20(tab)s

Clear L. L. then set E. L. = 04. 20 then compute L. L.

. 01. 3. 1. 3. 004. 0. 020

2(tab)s

v(tab)s

Type out Acc. Cell No. 2

. 01. 2. 2. 2. 000. 0. 000

5. Initiate Manual Operating Mode

OP. C. A. T_S . SRC. T_D . DST

Enter in the Manual Mode: -

INTERMAP P. P. R.

00. 0. 0. 0. 000. 0. 000

Accumulator Selection Routine

01. 0. 0. 0. 000. 0. 000

OP. C. A. T_S . SRC. T_D . DST

Enter in the Manual Mode: -

Typewriter Input-Output Routine 02. 0. 0. 0. 000. 0. 000

Punched Card Input-Output Routine 03. 0. 0. 0. 000. 0. 000

SPECIAL INTERMAP INSTRUCTIONS:

Incrementer 04. 3. 1. 0. 022. 0. 022

If N=5 and M=6 in Accumulator Cell No. 1, then after the execution of the special instruction above, the following change will occur in the instruction in . 22:

Before: .22 02.0.1.3.000.3.000

After: . 22 02. 0. 1. 3. 005. 3. 006

The next instruction to be executed is in . 22.

More examples of special instructions are found in INTERMAP Sequence in Section 5. 5.

ERROR HALTS

The only programmed HALTS (S = 16, D = 31) not accompanied by any type-outs are Error Halts. An Error Halt may occur anywhere in an INTER-MAP Sequence, and it is possible that at the time of the Halt the operator does not know which part of the program is being processed. The following procedure should be followed:

- 1. Observe the light configuration on the G15D panel and write down the following:
 - a. Command line which the Halt occurred
 - b. The characteristic of the Halt Command
- 2. Turn Compute Switch OFF then back to GO. The INTERMAP Control will search the program (punched tape or magnetic tape) for the INTERMAP P. P. R.
- 3. After entering I. P. P. R. the INTERMAP Instruction during which the error occurred will be typed out. Following this a type-in will be gated to accept any manual I. P. P. R. Instruction.
- 4. The type-out is equivalent to having executed a manual instruction of: -yab, and may therefore also immediately be followed by typing in a corrected INTERMAP Instruction.

Following the type-out of the incorrect INTERMAP Instruction, the Operation Routine in which the error occurred will be known. The operator may now determine the type of error by comparing the notes made in (1) above with the errors listed in the Error Table for the respective Operation Routine.

No Error Halts will occur in INTERMAP P. P. R. (OP = .00). However, see Section 4. 2, Editing of type-in errors.

When the same magnetic tape unit is both designated as a Source and as a Destination in the same INTERMAP Instruction (see Section 3.1.) the program will return to I. P. P. R to type out the INTERMAP Instruction in error, however, it will not come to a Halt as described in (1) and (2) above.

ERROR TABLE:

Accumulator Selection Routine (Automatic Mode)

(OP = .01)

NO.	LINE	CHARACTERISTIC	ERROR
1	00	1	Operand, $A = 0$
2	00	2	Operand, A > Amax
3	00	3	⊗ First Location, F. L < F. L. max
4	00	4	⊗ Last Location, L. L. < F. L. max
			⊗See table 2.1.1.

NO.	LINE	CHARACTERISTIC	ERROR
1	19	1	Operand, A > Amax
2	-	-	A decimal number greater than 10 ³⁸ . The i and j of the incorrect element are typed out after which the program continues. The mode is changed from auto-
			matic to manual and after conversion of the entire matrix the program will come to a halt at the 1st type-in, where the errors in the Flexowriter tape may be corrected manually by typing in the correct data.
			sc3f will return the program to the automatic mode.

ERROR TABLE: Punched Card - Input - Output Routine (OP = .03)
(Automatic Mode)

NO.	LINE	CHARACTERISTIC	ERROR
1	01	1	Operand, A > Amax or A = 0
2	01	2	The operand is too small to hold a matrix of the size specified by the First Card.
			(Test No. 1 has been passed).
3	06	3	$T_S + T_D \le 3$

NO.	LINE	CHARACTERISTIC	ERROR
1	02	0	Incrementation is not permitted because the Source or the Destination of the instruction to be incremented is equal to Zero.
2	01	1	When $C = 3$, $A > Amax$

NO.	LINE	CHARACTERISTIC	ERROR
1	00	1	The Source is an accumulator number which is greater than Amax.
2	00	2	The Destination is an accumulator number which is greater than Amax.
3	00	3	The Destination is an accumulator which is too small to hold the source matrix. The compatibility test failed.

NO.	LINE	CHARACTERISTIC	ERROR
1	00	0	The source is an accumulator number which is greater than Amax or greater than 7 if Amax = 7.
2	02	2	The destination is an accumulator number which is greater than Amax or greater than 7 if Amax = 7.
2	02	3	The number of rows of the operator (SRC) exceeds the number of rows of the operand. This does not exclude Error No. 4.
4	02	4	The number of columns of the operator (SRC) exceeds the number of columns of the operand. This excludes Error No. 3.
5	01	5	Destination accumulator is too small.
6	02	6	Destination tape - number of rows exceeds 54.
7	00	7	Source Tape - Source matrix is transposed.
8	01	2	Overflow; - The product of two elements exceeds 10 ³⁸ .
9	?	?	Operator greater than Amax.

NO.	LINE	CHARACTERISTIC	ERROR
1	02	1	C = 2, Illegal INTERMAP Instruction.
2	02	2	<pre>C = 3, Illegal INTERMAP Instruction.</pre>
3	02	3	Overflow: A product exceeds 10 ³⁸ .
4	01	1	Overflow: A quotient or sum exceeds 10^{38} .
5	01	2	Division by zero.

NO.	LINE	CHARACTERISTIC	ERROR
1	00	1	The label of the source data tape indicates:
			decimal data tape, or a transposed matrix
2	00	2	Validity Test fails. The number of columns in the premultiplier is not equal to the number of rows in the post multiplier, or $M_A \pm N_S$.
3	01	1	Occurs only where the product (DST) is on tape and when the number of rows of the premultiplier and product is greater than 54, i. e., the product matrix has columns in excess of 108 words.
4	01	2	Occurs only where the product (DST) is in an accumulator and when the destination accumulator is too small to hold the product matrix.
5	01	3	Overflow on addition. The absolute value of the sum of several
			products is greater than 10 ³⁸ .
6	02	1	Overflow on multiplication. The absolute value of the product of two elements is greater than 10^{38} .
7	02	2	C = 1 when T _D = 0. Either one may be a coding error. This prevents data being written on the basic magnetic tape which is not available for data.

NO.	LINE	CHARACTERISTIC	ERROR
1	00	0	The source is an accumulator number which is greater than Amax or greater than 7 if Amax = 7.
2	02	2	The destination is an accumulator number which is greater than Amax or greater than 7 if Amax = 7.
3	02	3	The number of rows of the operator (SRC) exceeds the number of rows of the operand. This does not exclude Error No. 4.
4	02	4	The number of columns of the operator (SRC) exceeds the number of columns of the operand. This excludes Error No. 3.
5	01	5	Destination accumulator is too small.
6	02	6	Destination tape - number of rows exceeds 54.
7	00	7	Source Tape - source matrix is transposed.
8	01	2	Overflow - the sum of two elements exceeds 10^{38} .
9	?	?	Operator greater than Amax.

ERROR TABLE: Matrix Transposition Routine (OP=.24)

No.	Line	Characteristic	Error
1	00	0	Operand greater than Amax
2	01	1	L _S > Nmax
3	01	2	Accumulator too small, change of L. L. not permitted (C=0 or 1)
4	01	3	Accumulator too small, change of L. L. permitted (C=2 or 3), but L _S > Nmax
5	01	4	Accumulator too small, change of L. L. permitted (C=2 or 3), $L_S \le Nmax$, but $L_S > M_D$.
6	01	5	$L_S > M_D$.

ERROR TABLE: Matrix Multiplication Routine (Comprehensive) (OP = .25)

No.	Line	Characteristic	Error
1	00		$ \frac{\text{Illegal Operations:}}{\text{Scalar Mode and Pre-diagonal Mode}} A_1 A_2 = A_1 $ $ A_1 A_1 = A_1 $ $ A_1 T_{\text{in}} = A_1 $
2	00	2	Post-diagonal $A_1 T_{in} = T_{out}$ mode:
3	01	2	Normal Mode: $A_1 A_1 = A_1$ Pre-diagonal mode: $A_1 A_1 = A_2$ $A_1 A_1 = A_2$ $A_1 A_1 = A_2$
			Post-diagonal $A_1 A_1 = A_2$ mode: $A_1 A_1 = A_1$ $A_1 A_1 = A_1$ $A_1 A_1 = A_1$
4	00	3	Source: Tape(1) Either a punched or a magnetic data tape may have a flagged label, or(2) a decimal label has been read from punched paper tape.
5	00	4	Validity test failed. The number of columns in the pre-multiplier is not equal to the number of rows in the post-multiplier.
6	00	5	Destination Tape: The number of rows in the product matrix exceeds 54.
7	01	1	Overflow on multiplication. The absolute value of the product of two elements is greater than 1038.

ERROR TABLE: Matrix Multiplication Routine (Comprehensive) (OP = .25) (continued)

No.	Line	Characteristic	Error
8	00	7	Overflow on addition. The absolute value of the sum of several products is greater than 10^{38} .
9	00	6	Compatibility test failed. The Destination is an accumulator too small to hold the product matrix.

ERROR TABLE: Decimal Data Tape Input Routine (OP = .26)

No.	<u>Line</u>	Characteristic	Error
1	00	1	C ≠ 0
			The INTERMAP Instruction is incorrectly written.
2	02	1	The Compatibility Test failed. The Destination is an accumulator which is too small to hold the Source matrix.
3	02	2	The Destination is an accumulator number which is greater than Amax.

CODING CONVENTIONS AND SYMBOLS

The following symbols and conventions are recommended:

- 1. Definitions (A6) = The contents of Acc. 6
 - (CA) = Punched cards
 - (PO) = Photoreader zero or basic G-15 Photoreader
 - (P2) = Auxiliary Photoreader number 2
 - (M2/005) = Magnetic Tape Unit Number 2, file number 5.
 - (TY) = Typewriter
- 2. Store and transfer operations (OP = .20 and .26).
 - $(A5) \xrightarrow{CA} (A6)$ Clear and add Acc. 5 to Acc. 6
 - (A3) $\xrightarrow{\text{CS}}$ (PO) Negate and punchout Acc. 3
 - (P3) TR (M2/006) Read from Aux. Photoreader Number 3 and transpose to Mag Tape Number 2 File Number 6
 - $(A6) \xrightarrow{CL} (A6)$ Clear to zero Acc. 6
- 3. Matrix Inversion and Equation Solver (OP = .21)
 - $(A1) \longrightarrow (A1)^{-1}$ Invert matrix in Acc. 1
 - Solve (A1) Solve equations in Acc. 1
- 4. Transposition (OP = .24)
 - $(A3) \longrightarrow (A3)^T$ Transpose matrix in Acc. 3
 - $(A4) \xrightarrow{P} (A4)^{T}$ Transpose matrix in Acc. 3 in the permissive mode.
- 5. Multiplication (OP = .22 and .25)
 - (A1) (PO)→(A1) Multiply Acc. 1 by matrix on basic photoreader and store product in Acc. 1
 - $(A1)_S$ $(A2) \rightarrow (A3)$ Acc. 1 is a scaler
 - (A1)_{PD} (A2)→(A3) Acc. 1 is a pre-diagonal multiplier
 - (A2) (A1)—(A3) Acc. 1 is a post diagonal multiplier

6. Addition (OP =
$$.23$$
)

$$(A6. A7) \longrightarrow (A6)$$
 As above, but negate.

8. Input Output (
$$OP = .02$$
 and $.03$).

If the input or output is less than an entire matrix, specify II and JJ as defined in Section 4.4.

(A6): F. L. =
$$10.16$$

(A6): L. L. =
$$10. u6$$

(A6):
$$N=5$$
; $M=6$

$$(A6)$$
: Nmax=8

10. Incrementer (OP =
$$.04$$
).

$$(DST)_{12}$$
 + $(A) \le (DST)$? Mode 0 and 1 increment the Destination of instruction $(SRC)_{12}$ + $(A) \le (DST)$? 12 by the operand Number A and test against the limit.

The following forms are provided to facilitate INTERMAP programming. The INTERMAP Coding sheet is suitable for coding the INTERMAP Instruction (See Section 3.1). The column labeled "ab" is for listing the locations in Line 18. Provision is made for the Operation to be written in symbolic representation (Example: $(Ty) \longrightarrow (A3)$ for Type into Acc. 3). The last three columns are used to record the order of the matrices involved.

The Accumulator Map (Fig. 3.4.2) is equivalent to the memory allocation chart. Space is provided in the top half of the sheet to map accumulator arrangements (See Fig. 2.3.2). Below special provision is made to list the Accumulators, their First and Last Locations, their size and order. Careful use of the Data Storage and INTERMAP Sequence lines will prevent overlapping.

The INTERMAP Magnetic Tape Log (Fig. 3.4.3) is essential for keeping the record of each Tape Unit. A separate log sheet is required for each Tape Unit. Under Serial No. columns are provided for five tape runs. In the File Number column the File Number that appears in SRC or DST is written. In the first column to the left (FAN) the First Available File Number is recorded. For example, for File Number 6 the number 106 would be written in the first (FAN) column to the left of the File Number column.

(×
•	
٠	4
١	4

Bendix Computer
Division
LOS ANGELES 45. CALIFORNIA

INTERMAP



PROBLEM_

CODING

DATE	PAGE	of
PREPARED BY:		

AUXILIARY EQUIPMENT:

NOTEC					S	OUR	CE		DE	ESTI	N,	ODERATION		A	S	RC	C DS		
NOTES	a b 0) P	С	А	Ts	S	R C	T	d I	D S	T	OPERATION	N	M	N	М	N		
		\top						T	\top								T		
							+	\dagger	\dagger								+		
								1	+					+	+-		\dagger		
					Н			╁	+		-			+	+	-	\dagger		
		+			Н	+		╁	+		-			+	+	-	+		
					Н	-	_	-	+	_	-			+-	+	-	+		
								1	4						-	ļ	\downarrow		
								\perp							_	ļ.,	\downarrow		
					T			T	\dagger								1		
					П			╁	\dashv					1			_		
		+			Н			\dagger	+		-				╁	+-	1		
					\vdash		+	+	+		 			+	+	+	+		
		-		_	\vdash	\vdash	-	╀	+	+	-			+	+	+	+		
								ł				1	1				1		



INTERMAP ACCUMULATOR MAP

DATE_____ PAGE_____ of ____

															_				
ACC	F.L.	L.L.	N	М	Nm				CO	MN	1E I	NT:	S						
1																			
2																			
3								 											
4							 						_						
5																			
6																			
7																			

A max. _____

DATA STORAGE: FROM ______TO _____

INTERMAP SEQUENCE: FROM _____TO ____

_	Be	ndi	x C	ompute	er Di	vision



INTERMAP MAGNETIC TAPE LOG TAPE UNIT NO.

DATE	PAGE_	0f	

		SERIAL N	10.			
DATE	DATE	DATE	DATE 	DATE	FILE NO.	CONTENTS
FAN:	FAN:	FAN:	FAN:	FAN:		

OPERATING INSTRUCTIONS

TO START INTERMAP

1. The G15D Version

Mount rewound magazine No. 1 on the G15D photoreader.

2. The MTA-2 Version

- a. The magnetic tape version of the INTERMAP Program has previously been written on a magnetic tape which is mounted on a unit whose selector switch is set at 4. For writing the MTA-2 version on tape see Section 7.2.
- b. The magnetic tape unit No. 4 is on, and the read head is located anywhere below the first and above the last program block.
- c. Mount rewound Control magazine on the G15D photoreader.

3. The PR-1 Version

Mount rewound magazine No. 1 on the G15D photoreader; mount magazine No. 2 on the auxiliary photoreader which responds to C = 1 (the master PR-1). If there should be more than two magazines, the third magazine is mounted on the auxiliary photoreader which responds to C = 2.

4. In order to clear the entire G15D memory, type:

<P> then to GO

<P> then to GO

When the light configuration is steady, repeat

<P> then to GO

Wait for a gated type-in and then proceed to step 3.

5. If it is not required to clear the G15D memory, type:

 $\langle P \rangle$ $\langle P \rangle$ then to GO.

Wait for the gated type-in, then proceed to step 6.

- 6. If no magnetic tape is used for auxiliary data storage, type: (tab)s.
- 7. If magnetic tape is used for data storage, the respective magnetic tape units should be ON, their selector switches set between 1 and 3, to correspond to T_S and T_D of INTERMAP Instructions which are activating these units. It is assumed that data stored on these tapes has been written in accordance with A.P. 61-MTSR.

Type: T (tab) FAN (tab)s where $1 \le T \le 3$ is the unit number and $1 \le FAN \le 999$ is the First Available File Number see section 1.4.0.

This type-in may be repeated any number of times. It should be repeated until the FFN is entered correctly for each of the tapes used. Following any type-in for which T and FFN is not within the prescribed limits, the computer comes to a HALT. Turn compute switch OFF then to GO, a Bell rings and type-in will again be gated. Repeat type-in. To proceed from here, type: (tab)s.

8. Following step 6, the program tape is searched for the INTERMAP P. P. R (OP = .00).

When the next type-in is gated, I.P.P.R will have been read in. From here follow the operating instructions for I.P.P.R.

If the magnetic tape version is used, the control magazine may be removed from the G15D photoreader, which is now available for reading data from punched paper tape.

INTERMAP PROGRAM PREPARATION ROUTINE (OP = .00)

GENERAL

There are two type-ins in I. P. P. R.

The first type-in will accept data, a decimal INTERMAP Instruction, or any manual I.P.P.R Instruction. Data will be stored in appropriate locations in line 18, and manual I.P.P.R instructions will be executed.

The second type-in only follows the manual I.P.P.R. instruction of: -zab (tab)s. Any number entered through the typewriter during this type-in is stored as a hexadecimal constant (7 digits and sign) in line 18.

The following is a list of the manual I.P.P.R instructions:

±	$\frac{\text{H.ab}}{\text{u.ab}}$	List INTERMAP Instructions
±	v.ab	Initiate computation
±	w.ab	Read INTERMAP Sequence tape
	x.ab	Punch out INTERMAP Sequence tape
±	y ab	Accept decimal type-in of INTERMAP Instructions

Type-in or type-out of hexadecimal constants

It is apparent from the above that manual I.P.P.R instructions of the u, y, and z type are the ones which cause type-ins and/or type-outs of data. These occur in four columns. The first column contains .ab which is the relative location of the data in line 18. The second column is the type-out of the contents of location ab. The third column is the type-in and the fourth column is a verification type-out.

MANUAL OPERATIONS

± z.ab

Type in INTERMAP instructions:

-yab (tab)s Permit the type-in of decimal INTERMAP Instructions
(Example 1) and store same in a binary form in successive locations starting with .ab, and followed by ab + 1 etc. If location .ab is not equal to zero, its contents will be typed out as a decimal INTERMAP Instruction.

+yab (tab)s The INTERMAP Instruction thus typed in is not stored (Example 2) in line 18, but is immediately executed on the Single Instruction Level. .ab may be any number between .00 and .u7. There will be no type-out (column 2) of the instruction contained in 18.u7-ab. On Breakpoint the program will halt inline .01 (C = 0; S = 28; D = 03) following the verification type-out (column 4) to permit the correction of type-in errors. If the type-in is correct, turn the

Compute Switch to GO and the INTERMAP Instruction typed in will be executed on the Single Instruction Level.

If the verification indicates a typing error, type sclf and the program returns to the gate type-in position.

Listing:

uab (tab)s Type out decimal INTERMAP Instructions starting from (Example 3) . ab and stop at the first INTERMAP Instruction whose binary equivalent is equal to zero.

-uab (tab)s As above, but do not stop at zero.

(Example 4) In either case listing will be terminated when ab = .u8. On B. P. operation the computer will halt after typing out ab. To return to the first type-in, type f.

TYPE-IN OR TYPE-OUT OF HEXADECIMAL CONSTANTS.

zab (tab)s Type out .ab (Col. 1) followed by the hexadecimal (Example 5) contents of ab (Col. 2) and return to the first type-in.

-zab (tab)s As above, but the second type-in will be gated instead.

(Example 6) Any number entered during the second type-in (Col. 3) will be stored in location . ab as a hexadecimal constant. A verification type-out follows (Col. 4) after which the program returns to the first type-in.

Commence Computing:

v. ab (tab)s Execute INTERMAP Instruction . ab on the Single Instruction Level, and return to I. P. P. R.

-vab (tab)s Execute a stored INTERMAP Sequence starting from instruction . ab on the Multiple Instruction Level.

Read INTERMAP Sequence Tape:

w00 (tab)s Read one block of INTERMAP Sequence tape and store in line 18, then return to first type-in.

-wab (tab)s Read one block of INTERMAP Sequence tape and store in line 18. Also, set the command counter CC=ab, then execute the stored INTERMAP Sequence starting from instruction .ab on the Multiple Instruction Level.

PUNCH PAPER TAPE

xab (tab)s Punch out the contents of line 18 from 18. u7 to 18. (u7-ab) (The sign of the typed-in instruction has no significance).

Following the manual I. P. P. R instruction xab (tab) s, the following will take place:

Line 19 is cleared

(18. u7 to 18. u7-ab) copies into line 19.

Line 19 is punched out as a block of 108 words of which words .00 to [u7-(ab+1)] are equal to zero.

When reading in the same punched tape following a wab (tab)s this will take place:

A block of 108 words is read into line 19. (Every word in line 19, starting from 19.01 is tested until the first non-zero word is found in (19. TT). 19. u7 to 19. TT is then copied into line 18. The contents of 18. TT-1 to 18.00 are not changed.

CLEAR LINE 18

x00 (tab)s Line 18 is cleared. This instruction may be positive or negative.

EDITING OF TYPE-IN ERRORS

INTERMAP P. P. R edits the decimal type-in of INTERMAP Instructions to conform to the limits stated in section 3.1.

Any error in typing in decimal instructions or hexadecimal constants which is detected before typing may be corrected by typing 12 digits (tab)s for instructions, and 7 digits (tab)s for constants.

In Example 7 are shown type-in error. The portion of the instruction which is in error is underlined. Note that the program rejects these type-ins, again types . ab and is ready to accept a new correct type-in. In Col. 4 is the verification type-out of the correct instruction as it was stored internally.

The manual instruction H ab is edited as follows: If H < u return to type-in.

No test exists for .ab > u7.

Col. 1	Col. 2	Col. 3	Col. 4
	EXAMPLE 1		
	Clear line 18		
×00	s y00- (Permit t	type in of decimal instructi	
.00 .01 .02 .03		010410000000 ⑤. 022130013001 ⑥. 033230022005 ⑥. y04- ⑥	. 01. 0. 4. 1. 000. 0. 000 . 02. 2. 1. 3. 001. 3. 001 . 03. 3. 2. 3. 002. 2. 005
. 04 . 05 . 06		220100020003 ⑤. 210300000000 ⑥.	. 22. 0. 1. 0. 002. 0. 003 . 21. 0. 3. 0. 000. 0. 000
. 05	. 21. 0. 3. 0. 000. 0. 000.	y05- ⑤ 211300000000 ⑤ .	. 21. 1. 3. 0. 000. 0. 000
	EXAMPLE 2		
0.1			e Instruction Level Operation)
.01		230100020003 ⑤. y03 ⑤	. 23. 0. 1. 0. 002. 0. 003 ⑤ clf
. 03		012330023000 ⑤ .	. 01. 2. 3. 3. 002. 3. 000 ⑤ clf
	EXAMPLE 3		
.00 .01 .02 .03	. 01. 0. 4. 1. 000. 0. 000 . 02. 2. 1. 3. 001. 3. 001 . 03. 3. 2. 3. 002. 2. 005	u00 🜀 (List ins	structions which are non-zero)

Table 4. 2. 1

Col. 1	Col. 2	Col. 3	Col. 4
	EXAMPLE 4		
.00 .01 .02 .03 .04 .05 .06 .07 .08 .09	.01.0.4.1.000.0.000 .02.2.1.3.001.3.001 .03.3.2.3.002.2.005 .22.0.1.0.002.0.003 .21.1.3.0.000.0.000	u00- (S)	(List all instructions)

Table 4. 2. 1

Col. 1	Col. 2	Col. 3	Col. 4	
	EXAMPLE 5			
	Clear line 18			
x00 .01 .02 .03 .04 .02	z01- ⑤ . 2222222 EXAMPLE 6	(Permit type in of binary 1111111- (3). 2222222 (3). 2222222- (3). 2222222- (3). 3333333- (3).	constants) 1111111 .2222222 zzzzzzz .yyyyyyy 3333333	z02- ⑤ z03- ⑥ z04- ⑤ z02- ⑤
.01 .02 .03 .04 .05 .06	1111111 3333333 zzzzzzz - yyyyyyy	z01 (6) z02 (6) z03 (6) z04 (6) z05 (6) z06 (6)	(Type out as binary constant)	

Table 4.2.1

Col. 1	Col. 2	Col. 3	Col. 4
	EXAN	MPLE 7 - ERRONEOUS TYPE-	INS
x00 (S) .00 .00 .01	y00- ⑤	OP > 63 <u>64</u> 0130562055 ⑤ 630130562055 ⑤ .	. 63. 0. 1. 3. 056. 2. 055
.01		$C > 3$ 014130013001 (§) $01\overline{3}130013001$ (§).	. 01. 3. 1. 3. 001. 3. 001
02 . 02 . 03		y02- ⑤ A > 7 012830013001 ⑤ 012730013001 ⑤ .	. 01. 2. 7. 3. 001. 3. 001
. 03 . 03 . 04		$y03-$ § $T_s > 3$ $223\underline{4}0013001$ § $223\overline{3}0013001$ §.	. 02. 2. 3. 3. 001. 3. 001
.04 .04 .04 .04 .04		$ \begin{array}{c} y04- & \textcircled{\$} \\ 45321283001 & \textcircled{\$} \\ 45321273001 & \textcircled{\$} \\ 45321263001 & \textcircled{\$} \\ 453231283001 & \textcircled{\$} \\ 453231273001 & \textcircled{\$}. \end{array} $	C > 3 SRC > 127 . 45. 3. 2. 3. 127. 3. 001

Table 4. 2. 1

ACCUMULATOR SELECTION ROUTINE (OP = .01)

The Accumulator Selection Routine has two type-ins. In both, the characteristic neons on the G15D panel will show: "OFF, ON" or a binary "one" which is the op-code number for this routine.

Type-ins are preceded by one and two bells respectively. A second type-in always follows a first.

The first type-in follows the second only if the second type-in is:

negative or

u0000 or

v0000.

All positive second type-ins are followed by another second type-in.

To return to the first type-in from any part of the program, type scf or sclf.

The significance of these type-ins is explained in more detail in section 5.2

MANUAL OPERATIONS

The first type-in is of the following format: A(tab)s where A is the operand whose Accumulator Cell is to be set by the second type-in.

The second type-in is of the following format:

+ Habcd(tab)s, where:

H. ab. cd

9NNMM Set N and M

u0000 Set Amax equal to A of First type-in

v0000 Type-out accumulator cell

WLLTT Set Last Location to 00.00 and First Location to LL. TT

xLLTT Set Last Location to LL. TT

yNNMM Clear cell then set N and M

zNN00 Set 2Nmax = 2(NN).

All operations are performed on the accumulator cell specified by the First type-in.

Insignificant zeros as shown above need not be typed, for example:

u(tab)s Set Amax equal to A of First Type-In

v(tab)s Type-out accumulator cell

x12(tab)s Set Last Location to 12.00

y(tab)s Clear cell, then set N = 0 and M = 0

z06(tab)s Set 2Nmax = 12

Amax

Generally, the following has to be observed for the first type-in:

 $1 \le A \le A \max$

If, A > Amax

an Error Halt occurs following the second type-in. Turn the Compute Switch OFF then back to GO. In the manual mode the program returns to the first type-in, but in the automatic mode control will return to I. P. P. R and the incorrect INTERMAP Instruction will be type-out.

There is one exception to this rule. If the second type-in is u(tab)s, no Error Halt occurs when A > Amax, thus permitting the changing of Amax to lower or higher numbers.

EDITING OF TYPE IN ERRORS

In the manual mode the type-ins are tested for the following errors:

No.		Consequence
1	A=0	Return to the 1st type-in.
2	A > 7	Return to the 1st type-in.
3	H < 9	Return to the 2nd type-in.
4	A > Amax	HALT in line 00 (2.16.31.)
		Then Compute Switch OFF and back to GO to return program to the 1st type-in.
5	⊗ First Location < F. L. max	HALT in line 00 (3.16.31.)
		Turn Compute Switch OFF and back to GO to return program to the 2nd type-in.
6	⊗ Last Location < F. L. max	HALT in line 00 (4.16.31)
		Turn Compute Switch OFF and back to GO to return program to the 2nd type-in.
	⊗See table 2.1.1	

TYPEWRITER - INPUT/OUTPUT ROUTINE (O = .02)

GENERAL

There are two type-ins for manual instructions in this routine. At both type-ins the characteristic neons on the G15D panel show the following: ON, OFF which is the equivalent of a binary two, the op-code number of this Operation Routine.

Both type-ins are preceded by one or two bells respectively. On entering this Operation Routine in the manual mode, one ring bell will occur and the first type-in will be gated.

The first type-in is of the form \pm A NNMM (tab)s, where \pm A is the accumulator number NN & MM are the number of rows and columns respectively.

If the type-in is of the form ±A0000 the order or the setting of the accumulator cell is not changed. The latter form would most often be used on output, generally at times when the operator does not know what the setting of an accumulator is, but wants to type it out as is.

If the type-in is of the form -A NNMM the accumulator cell setting will be changed \(\neq \) appropriately.

If, however, the type-in is of the form A NNMM the setting of the accumulator cell will not be changed, but subsequent type-ins and type-outs are from a matrix of the order specified.

For example; 20×20 matrix is contained in accumulator No. 1 and it is desired to type out the upper left part of that matrix only, a type-in of 11010 (tab)s will specify that the matrix to be operated on in subsequent type-ins is to be a 10×10 matrix.

The following rules apply to all manual instructions performed by the second type-in:

a. The format of the type-in is: \pm H. II. JJ(tab)s

where
$$9 \le H \le z$$
 and

$$01 \le II \le u7$$
 and $01 \le JJ \le u7$

- b. Generally, negative type-ins are output instructions and positive type-ins are input instructions, though there are exceptions to this rule which are stated below.
- c. H. II. JJ means: Perform a specific operation defined by H for values of i and j in the following range.

If
$$H = \pm W$$
 and $\pm z$
$$\begin{cases} II \le ii \le N \\ JJ \le jj \le M \end{cases}$$

If
$$H = \pm u$$
 or $\pm x$

(a row)

If $H = \pm v$ or $\pm y$

(a column)

$$H = 9$$

(diagonal)

$$II = ii$$

$$JJ \le jj \le M$$

$$JJ = jj$$

$$JJ \le jj \le M$$

$$II \le ii \le M + (II-JJ)$$

MANUAL INSTRUCTIONS

The following is a list of the manual operations:

± 9IIJJ Input or output of elements of a diagonal

± xIIJJ Input or output of elements of a row

± yIIJJ Input or output of elements of a column

± zIIJJ Input or output of the elements of several columns

- uIIJJ Enter zeros into elements of a row

- vIIJJ Enter zeros into elements of a column

- wIIJJ Enter zeros into elements of several columns

+ uIIJJ Transfer elements in the I^{th} row into respective positions in the $(I-1)^{th}$ row, where I > 1

+ vIIJJ Transfer elements in the J^{th} column into respective positions in the $(J-1)^{th}$ column, where J>1

+ wIIJJ Convert double precision floating point numbers stored in several columns.

SEQUENCE AND FORMAT OF TYPE-INS AND TYPE-OUTS

The input and output of data occurs by columns. (The operation on a row is logically the same as the operation on several columns having only one row each).

All data type-outs are as follows:

The data inputs are:

The above are examples of the type-out of a column, since . ii is incremented.

The method can be further illustrated by the following examples of the type-outs from a 4×5 matrix. Remember, though, that actual type-outs are all vertical.

z0203(tab)s		$\times 0101(tab)s$
a ₂₃ a ₂₄ a ₂₅		a_{11} a_{12} a_{13} a_{14} a_{15}
^a 33 ^a 34 ^a 35		
^a 43 ^a 44 ^a 45		$\times 0202(tab)s$
43 44 45		^a 22 ^a 23 ^a 24 ^a 25
z0101(tab)s		
a ₁₁ a ₁₂ a ₁₃	a ₁₄ a ₁₅	<u>y0105(tab)s</u>
^a 21 ^a 22 ^a 23	a ₂₄ a ₂₅	a ₁₅
^a 31 ^a 32 ^a 33	a ₃₄ a ₃₅	^a 25
a ₄₁ a ₄₂ a ₄₃	a ₄₄ a ₄₅	^a 35
		^a 45
z0405(tab)s		<u>x0405(tab)s</u> <u>y0405(tab)s</u>
^a 45		^a 45 ^a 45

RETURN TO TYPE-IN MODES

A return to the first type-in mode is possible only if the second type-in is gated: scf.

Return to the second type-in is achieved by using one of the two following commands:

- 1. Type ZZZ(tab)s during input when data type-in is gated.
- 2. Type sc6f During and before completion of output with the Compute Switch in the OFF position.

INPUT OF SINGLE ELEMENTS (aij) IN RANDOM LOCATIONS

In general the format of the manual instruction of the second type-in is ±HIIJJ. However, if H is eliminated this type-in may be used to enter single elements into random locations in an accumulator.

For example: To enter a number into the third column of the second row, type:-

After the verification type-out the program returns to the second typein ready to accept another manual instruction.

This feature may be used to advantage in the following ways:

- 1. To correct or change single elements.
- 2. To enter a matrix which has a predominance of zero elements. In this case, to avoid typing the zero elements, clear the accumulator first and then type in the non-zero elements in the manner described above.

SUPPRESSION OF VERIFICATION TYPE-OUT

The verification type-out may be suppressed by turning the Punch Switch to the ON position. (To avoid simultaneous punching and typing turn off the punch motor switch on the inside of the punch assembly.) It is to be noted that with the Punch Switch ON, no zero suppression will occur.

EDITING OF TYPE-INS AND ERRORS:

In the manual modes the following tests are performed:

No.	Error	Consequence
1	Operand, A > Amax	HALT in line 19 (1.16.31.)
		Turn Compute Switch OFF and back to GO to return the program to the 1st type-in. (Test No. 2 was not yet passed)
2	The Operand is too small to hold the matrix specified by the 1st	Halt in line 19 (2.16.31.)
	type-in of A. NN. MM.	Turn Compute Switch OFF and back to GO to return the program
	(Test No. 1 has been passed)	to the 1st type-in.
3	Following a type-in of W. I IJJ (tab)s, a decimal number greater than 10 ³⁸ .	The i and j of the incorrect elements are typed out without interrupting the operation. On return to the 1st type-in these elements may be corrected manually by typing in the correct data.

PUNCHED CARD - INPUT - OUTPUT ROUTINE (OP = .03)

There are two type-ins for manual instructions in this routine. At both type-in's the characteristic neons on the G15D panel show the following: ON; ON, which is the equivalent of a binary three, the op-code number of this Operation Routine.

Both type-ins are preceded by one or two bells respectively. On entering this Operation Routine in the manual mode, one ring bell will occur and the first type-in will be gated.

The First type-in: ± A(tab)s specifies the accumulator number or operand. The minus sign has the following significance:

If the Second type-in is negative, which means data input, the sign is disregarded.

If the second type-in is positive, which means data output, then + A will cause a data card to be punched for every Zero-or-non-Zero element, and -A will cause data cards to be punched only for non-Zero elements.

The Second type-in: ± HIIJJ(tab)s has the following significance:

In general (a) a positive instruction means output and a negative one input.

(b)
$$0 < II \le u7$$

 $0 < JJ \le u7$
 $X < H \le Z$

If $H = \pm Z$, then the operation is performed on a matrix or part thereof whose elements are in the following rows and columns, or i and j respectively:

$$II \le i \le N$$
$$JJ \le j \le M$$

If $H = \pm Y$, the operation is performed on a column such that

$$II \le i \le N$$

$$JJ = j$$

and similarly, if $H = \pm X$, the operation is performed on a row, and

$$II = i$$

$$JJ \le j \le M$$

The following is a summary of the manual operations:-

± XIIJJ	Input or	output	of the	elements	of a	a row.	
± YIIJJ	Input or	output	of the	elements	of a	a colum	n.
$\pm ZIIJJ$	Input or	output	of the	elements	of s	several	columns.

CARD HANDLING

When placing a deck of cards into the hopper be sure that the first and last card in the deck are a First Card and Last Card respectively.

EDITING TYPE-INS AND DATA

The following tests are performed in the manual mode:

NO.	Error	Consequence
1	Operand, A > Amax or A = 0	HALT in line 01 (1.16.31). Turn Compute Switch OFF and back to GO to return program to 1st type-in.
2	The Operand is too small to hold the matrix specified by the First Card.	HALT in line 01 (2.16.31). Turn Compute Switch OFF and back to GO to return program to 1st type-in.
	(Test No. 1 has been passed)	Remove cards from hopper, run out remaining cards into stacker and place deck into hopper.
3	H < X	HALT in line 06 (3.16.31). Turn Compute Switch OFF and back to GO to return program to 2nd type-in.

The following tests are performed both in the Manual and Automatic Mode. The Computer halts on a Punch Switch OFF test ringing a continuous bell. The bell may be silenced by turning the Compute Switch OFF. After taking the remedial action outlined below turn Punch Switch to ON and back to OFF and the program will continue as described below.

4	No first Card in the deck.	Remove cards from the hopper, manually run out all the remaining cards into the stacker. Obtain a First Card and place deck into hopper. After manipulating the Punch Switch the program will proceed to read the First Card again, and continue
		normally from thereon.

NO. Error

5 The decimal number on a data card is greater than 10^{38} .

Consequence

The faulty card is the top card in the stacker. Remove card, manipulate the Punch Switch and the program will continue to read and store cards from the deck. Several incorrect cards may be detected in a deck. After reading the Last Card of the deck the program will go unconditionally to the 1st type in. All incorrect data cards may now be key punched again and be read in under manual control as a supplementary deck using the same First and Last Cards.

THE SCOUT

If the INTERMAP Program is in the manual mode or if the completion of the present operation is of no significance, proceed immediately to step 1.

If the INTERMAP Program is not in the manual mode of one of the service type Operation Routines, then turn the Compute Switch to B.P. The computer will come to a halt immediately before the search for the next Operation Routine would begin. The light configuration on the G15D neon panel will be as follows: Command Line: 03, Characteristic 3, S = 21, D = 31. Whatever operation was in progress is now completed. Proceed to step 1.

Remember that the data type-out is breakpointed, (OP = 02) and if a data type out is programmed do not turn the Compute Switch to B.P. until the type-outs are completed.

STEP

- 1. The Scout has self rewinding feature and must therefore be mounted on a tape magazine.
- 2. To load, type: < P > and turn compute switch to GO. This is followed by a type-out of the Command Counter which is the location in line 18 of the INTERMAP Instruction last executed.
- 3. The computer is now at a programmed HALT (S = 16, D = 31), and the Operator has two options.
 - 4. Option 1. To return to INTERMAP P. P. R and type-out the last executed instruction, turn Compute Switch to B. P. then back to GO, and proceed to (5).
 - Option 2. Turn Compute Switch to GO, and a type-in will be gated.

 Type: .ab(tab)s, where .ab is the location in line 18
 of an INTERMAP Instruction which is to be executed
 next. Proceed to (5).
- 5. The program magazine will be rewound automatically, and the computer will then come to a HALT (S = 16, D = 31). If the G15D Version or the PR-1 Version is used, remove the Scout magazine from the photo-reader and replace by the respective program magazine. There is no need to replace the Scout magazine if the MTA-2 Version is used.
- 6. Turn the Compute Switch OFF then back to GO. The INTERMAP Control takes over in accordance with the option chosen in (4) above.

THE MEMORY DUMP

The memory dump must be loaded right after the execution of an instruction and prior to the search for the next Operation Routine.

SINGLE INSTRUCTION LEVEL

On the Single Instruction Level this is best done after the automatic return to INTERMAP P.P.R.

MULTIPLE INSTRUCTION LEVEL

On the Multiple Instruction Level this may be done in several ways:

- (a) In the manual mode before typing sc3f in order to execute the next instruction.
- (b) In the automatic mode turn the Compute Switch to B.P. The computer halts at the following light configuration: C = 3, S = 21, D = 31, Command Line 3. Proceed to Step 1, below.

There are several other halts in the program at which the memory dump may be performed. However, the operator must be fully aware at what stage in the execution of an INTERMAP Instruction these halts occur.

The following halts occur before the execution:

- (a) Following the type-out of the flag on the changing program magazines.
- (b) Before reading in of data tapes (Punch Switch ON).
- (c) At the first type-in in any service type operation routine.

The following halts occur during the execution:

- (a) All Error Halts
- (b) The halt described in "Reading Date from Punched Paper Tape", Section 4.8, Step 8.
- (c) Type in gated for double precision date type-in (OP = 02).

The following halt occurs after the execution:

(a) After reading of data tapes (Punch Switch ON).

If the Command Counter, is set at .ab during a memory dump, then the next INTERMAP Instruction to be executed on reading the memory dump is ab + 1. It follows, that if a memory dump is performed before or

during the execution of INTERMAP Instruction .ab, this instruction will not be completed. It may, however, be repeated by following instructions in Step 6 below.

For a memory dump after the execution of INTERMAP Instruction .ab follow instructions in Step 1 below: -

STEP

INSTRUCTIONS

- 1. Place Rewound Scout and Memory Dump Magazine on photoreader.
- 2. Type < P > then repeat < P > and turn Compute Switch to GO.
- 3. After the neon lights have become steady, a type-in will be gated. Type: LL (Tab)S, where LL is the highest channel number which contains pertinent data. The memory dump will now take place.
- 4. To read the memory dump back in, place it on the photoreader and type: <P>. Turn the Compute Switch to GO and the memory dump will load itself and come to a HALT (S = 16, D = 31), after all the data and lines .03, .04 and .18 have been restored to the memory.
- 5. Here two options exist:
 - Option 1: Turn Compute Switch to GO and the INTERMAP Sequence will be resumed on the same Instruction Level as before the memory dump was performed.
 - Option 2: Turn Compute Switch to B.P. and the computer will halt in Command Line 03; (C = 3, S = 21, D = 31). This halt occurs after the execution of .ab but before the execution of ab + 1.

If it is desired to continue on a different level or with an INTERMAP Instruction other than ab + 1, the Scout may be used.

6. If it is required to reset the Command Counter before performing the memory dump, use the Scout and follow its operating instructions from step 1 to 4, taking option 2. Type in: cd(Tab)s where .cd is the first INTERMAP Instruction to be executed when the memory dump is reloaded. From here continue with step (2) above.

READING DATA FROM PUNCHED PAPER TAPE

GENERAL

The following applies to all Operation Routines which permit a paper tape Source.

- 1. If the Punch Switch is ON, the computer comes to a halt on a Compute Switch Test ($S=17,\ D=31$) immediately prior to a programmed reading of punched data tape.
- 2. Remove the INTERMAP Program magazine from the photo-reader (in case of the PR-1 version, this may also be the master PR-1) and replace by data tape.
- 3. Move Punch Switch to OFF position (center) then back to ON. Data tape will be read in followed by another halt on a Punch Switch Test (S = 17, D = 31).
- 4. Replace INTERMAP Program magazine and turn Punch Switch to OFF position. Normal operations will be resumed.
- 5. The Punch Switch should be in the OFF position when a type-out occurs. The following type-outs may occur:
 - a. Programmed data type-out (OP = .02).
 - b. INTERMAP P. P. R and Error Indications.
 - c. Accumulator Selection Routine.
 - d. Typing out of a flag when changing program magazines in the G15D Version of INTERMAP.
- 6. Any type-out while the Punch Switch is ON will cause a stop code to be punched on tape. If this should occur, it is recommended to remove this portion of the tape before valid data is punched out.
- 7. If an INTERMAP Sequence, or a part thereof, does not have any programmed punched tape output, extraneous punching may be avoided by turning the punch motor off.

Ideally the Punch Switch is always OFF and sufficient independent sources of input are provided to run an INTERMAP Sequence fully automatically and unattended.

POST MULTIPLIERS

The following rules have to be observed only when using the comprehensive Matrix Multiplication Routine (OP = .25).

- 8. Following (2) above move Punch Switch to OFF position then back to ON. The Label only will be read in whereafter the computer will come to a regular HALT (S = 16, D = 31).
- 9. Replace data tape by program magazine, then move Compute Switch to OFF and back to GO. More program tape will be read in after which the computer will halt again on a Punch Switch Test.
 - 10. Replace program magazine by data and continue with (3) above.
- 11. In the case of: $A_1T_{IN} = A_1$, see Table 5.12.1. The program makes n passes, i.e., reads the source tape n times. Between each pass the tape is automatically rewound and must therefore be mounted on a tape magazine in order to function properly. This does not affect any of the other operating instructions.

The rules outlined in (8) (9) and (10) above apply only to the G15D Version and the PR1 Version of INTERMAP. There is no HALT in the MTA-2 Version and in the PR1 Version.

In the MTA-2 Version, where the program source is on magnetic tape, the additional program which has to be read following the Label does not require the basic photo-reader. The HALT described in (8) above has therefore been removed.

PROGRAM TO MAGNETIC TAPE LOADING ROUTINE (M.T.L.R.)

- 1. Select a magnetic tape unit and turn Selector Switch to 4.
- 2. Use A. P. 61 M. T. S. R. to locate the next available serial number on this tape. This number becomes the First Available File Number or FAN.
- 3. Place rewound M.T.L.R magazine on the photoreader and type: $\langle P \rangle$, then turn Compute Switch to GO. The program is read in and a type-in gated.
- 4. Now place the rewound first magazine of the INTERMAP MTA-2 Version on the photoreader and type: FAN(Tab)S. The reading of punched tape and its transcription on magnetic tape will commence.
- 5. The program tape of the MTA-2 Version must either be mounted on three magazines, referred to as Mag. I, II and III, or it may be read with a read hub and a capstan drive if the three parts have been spliced together.
- 6. If the program is on one continuous punched tape the M.T.L.R will stop automatically after the last block of tape, called the Last Loader or Return Flag has been read.
- 7. If mounted on magazines, the computer will halt about 20 second after the last block of tape on a magazine has been read. Replace the magazine by the next magazine, either Mag. II or III, turn Compute Switch OFF and back to GO and reading will continue.

At the end of the third magazine the Last Loader or Return Flag will be encountered which will terminate the tape reading as in (6) above.

- 8. After the Last Loader has been read the magnetic tape will automatically be reversed and lined up with INTERMAP P.P.R (OP = .00) and the next available serial number is typed out, see Table 7.1.1.
- 9. The check sums of all punched tape blocks are compared to the check sums of these same blocks after being written onto and read from magnetic tape. If they agree, the check sum is typed out.

Thus, the M.T.L.R. is self-checking to the extent that no read in error occurs on the photoreader. It is not affected by changes to the MTA-2 Version.

- 10. If an error occurs; that is, if check sums do not agree in (9) above, the following will take place:
 - a. The computer comes to a HALT. Turn Compute Switch OFF and back to GO.
 - b. The program makes a second attempt to read from magnetic tape and compare check sums. If this attempt succeeds,

- the loading operation will go to completion as described in (6), (7) and (8) above.
- c. If the error re-occurs, the next available serial number will be typed out and the magnetic tape is lined up immediately following this serial number. The computer then comes to another HALT.
- 11. The failure of the program described in 10 (c) would indicate an MTA-2 failure to read and/or write correctly.
- The M.T.L.R may be restarted by turning the Compute Switch OFF and back to GO and proceeding with Step 4 above. The FAN is now the next available serial number typed out in 10 (c).

THE OPERATION ROUTINES

This section gives a detailed description of all Operating Routines. INTERMAP is designed to make it possible to add to the Operation Routines Library any number of new routines.

The number of Operation Routines which may be used at any one time is limited to 64. The OP-code numbers, except for .00, .01, .02 and .03, which are service type Operation Routines have no significance. They have been assigned in what is considered to be a logical sequence which minimizes tape search and the exchange of program magazines on the photoreader.

For specific applications, special INTERMAP Program Tapes may be prepared which contain only those Operation Routines needed, thereby further improving operating speeds.

INTERMAP PROGRAM PREPARATION ROUTINE (I.P.P.R.) (OP = .00)

INTERMAP P. P. R. has the following functions:

Program Preparation

Control Functions

Error Indications

PROGRAM PREPARATION

- 1. Accepts the decimal type-in of INTERMAP Instructions and stores them in their binary form.
 - 2. Accepts the type-in of hexadecimal constants.
 - 3. Lists INTERMAP Instructions.
 - 4. Punches out INTERMAP Sequence tapes.
 - 5. Reads and stores previously prepared INTERMAP Sequence tapes.

CONTROL FUNCTIONS

Initiates computation, starting from any specified INTERMAP Instruction, on either the Single Instruction or the Multiple Instruction Level.

All operations are initiated by a type-in of the form:

± H.ab (tab) s.

where

 $U \le H \le Z$

is an Op-code initiating one of the six functions listed above

and

00≤a b≤U7

is a command sequence number indicating a word location in line 18. (INTERMAP Instruction .ab is stored in 18. U7-ab, because the INTERMAP Sequence is stored starting at 18:U7 and in reverse order.)

ERROR INDICATIONS

Following every ERROR HALT in the program (see 3.3) control will unconditionally be returned to I. P. P. R. followed by a type-out of the INTERMAP Instruction during which the failure occurred.

Any remedial step which may be taken to correct the error can be initiated from I. P. P. R.

ACCUMULATOR SELECTION ROUTINE (OP = .01)

Accumulator Cells are initially set up by this routine. They consist of the following:

Nmax - which is the maximum number of rows and accumulator can hold.

F.L. & L.L. - First Location and Last Location, of $/a_{1,\,1}/$ and $/a_{Nmax}/$ respectively.

The order of the matrix presently contained in the Accumulator.

Nmax is stored internally as $2Nmax=D_j$, the column increment (see sec. 2.2.1). N, is stored as 2N for computational purposes.

There are two methods of setting up accumulator cells. The first method required only that the size of the accumulator be specified by the First and Last Locations, where the Last Location is an arbitrarily chosen location and is not a function of N and M (L. L. \neq f(N, M)).

This method relies on N, M and Nmax being set up automatically, whenever this accumulator becomes the destination of a matrix, as a result of passing the compatibility tests.

In the second method, it is required that all parts of the accumulator be defined. This situation may arise if:

- l. Staggered accumulators are used which all require the same fixed $2N\max$ or $D_{\rm i}$.
 - 2. The Last Location is a function of N and M, namely

$$L.L. = F.L. + 2Nmax(M-1) + 2(N-1),$$

where

N≤Nmax and

M=Mmax

It can be proved that an accumulator in which L. L. = f(N, M) always represents a rectangular array.

The Accumulator Selection Routine has two type-ins, called the first and second type-in, each of which is preceded by one or two bells respectively.

In case of either type-in the characteristic C on the G-15D neon panel will show .01 in binary which is the op-code number for this Operation Routine.

First type-in:

A (tab) S

where, A is the Accumulator Number, and

 $1 \le A \le Amax$

Second type-in:

± H. abcd (tab) S

where $9 \le H \le$ may be one of seven different instructions, and abcd is LLTT when reference is made to the First and Last Location, or NNMM when reference is made to: N, M or Nmax.

Accumulator Cells are in 4-word groups, one cell to each accumulator (see Memory Allocation, Section 2.1).

The contents of these four words are as follows:

.00	2Nmax
.01	F.L. and L.L.
. 02	2N
. 03	М

The second type-in will cause data to be entered into these locations in the 4-word cell pertaining to the accumulator selected by the first type-in.

These type-ins are in the following form -

For:	Type:	Internal Storage:
Nmax	ZNN(tab)s	Set 2Nmax = (.00)
First Location	WLLTT(tab)s	Set L.L. = 0 then Set F.L. = LLTT = (.01)
Last Location	XLLTT(tab)s	Set L.L. = LLTT = (.01)
N and M	9NNMM(tab)s	Set $2N = (.02)$ and $M = (.03)$
N and M	YNNMM(tab)s	Set entire cell equal to zero then set $2N = (.02)$ and $M = (.03)$

The simplified flow chart in Figure 5.2.1 illustrates the functions of this Operation Routine.

It can be seen from the flow chart, that as long as the second type-in is positive, no computation and testing will take place. Thus a cell may be set up by repeated type-ins.

The following type-ins for example will set up accumulator No. 1 as shown in Figure 2.4.1.

l Bell	l(tab)s	Ready Acc. No. 1 for the following changes:
2 Bells	W1006(tab)s	Set First Location = 10.06
2 Bells	X 1064(tab)s	Set Last Location = 10.64
2 Bells	90304(tab)s	Set $N = 3$ and $M = 4$
2 Bells	z09(tabs)	Set Nmax = 9
2 Bells	v (tabs)	Type-out Acc. Cell No. 1

which is followed by the following type-out: (Carriage Return)

Acc. No.	<u>F.L.</u>	$\frac{L.L.}{}$	$\frac{N}{N}$	<u>M</u>	Nmax
1	10.06	10.64	3	4	9

1 Bell (Program returned to First type-in.)

In this case the programmer has to compute manually or estimate the Last Location before typing it in.

The next example is identical to the above type-ins with the exception that N = 4 and M = 4:-

l Bell	l(tab)s	Ready Acc. No. 1 for the following changes:
2 Bells	W1006(tab)s	Set First Location = 10.06
2 Bells	X1064(tab)s	Set Last Location = 10.64
2 Bells	90404(tab)s	Set $N = 4$, and $M = 4$
2 Bells	z09(tab)s	Set Nmax = 9
2 Bells	v: (tab)s	Type out Acc. Cell No. 1

followed by this type-out: (Carriage Return)

Acc. No.	F.L.	$\frac{L. L.}{}$	$\frac{N}{}$	$\frac{M}{}$	$\frac{Nmax}{}$
1	10.06	10.64	4	4	9

It will be noted that in both examples the Last Location is the same, i.e., in neither case is the Last Location a function of N and M. However, reference to Figure 2.4.1 shows that in the second example the Last Location

should be at 10.66, 67 and not at 10.64, 65. This is also born out by formula (3), section 2.2.

L. L. =
$$(10x108+6) + 18(3) + 2(3) = 1146$$
 or 10.66 .

Obviously then, setting up an Accumulator Cell by the method demonstrated above (L.L. \neq f(N, M)) does not detect errors such as these. This error would be detected at a later stage by the Compatibility Tests.

The more common and safer way of setting up Accumulator Cells is by the second method. Assume, for example, that it is required to set up Accumulator No. 1 to the following specifications:

F.L. = 10.06

$$N = 9$$

 $M = 4$
L.L. = $f(N, M)$.

(1 Bell) now type: l(tab)s Ready Acc. 1 for the following changes

2 Bells y0904(tab)s Clear cell then set N=9 and M=4.

2 Bells -w1006(tab)s Set L. L. = 0 then
Set F. L. = 10.06, then proceed
to testing and computing L. L.

A type-out will follows as shown: (Carriage Return)

Acc. No.	$\underline{\mathbf{F} \cdot \mathbf{L}}$.	<u>L.L.</u>	$\frac{N}{}$	\underline{M}	Nmax
1	10.06	10.76	9	4	9

The accumulator set up thus is rectangular and L.L. = f(N, M) is correct.

This accumulator is now properly set up to store matrices of the following order:

1≤N≤9

 $1 \le M \le 4$

As long as Nmax remains unchanged the address of any element a_{ij} in any matrix of an order in this range will be the same.

Other matrices in which either N or M exceed the respective limits of 9 and 4 may be stored provided that NM \leq 36. Of course, since this involves a change of 2Nmax during the Compatibility Test, the address of elements aij for these matrices will no longer be the same.

It will be noted that this Operation Routine favors rectangular or square accumulators. This is achieved as follows:

The manual instruction y NN. MM(tab)s clears the accumulator cell before storing N and M in their respective locations. Thus 2Nmax will always be equal to zero following such an instruction. Test No. 1 on the flow chart, i.e.: 2N≤2Nmax will always be answered "NO", and consequently 2Nmax will be replaced by 2N.

The manual instruction wLL.TT(tab)s unconditionally clears the Last Location. Thus, test No. 2, i.e.

$$F.L. + 2Nmax(M-1) + 2(N-1) \le L.L.$$
 will also be answered

will also be answered "NO" and the change

$$F.L. + 2Nmax(M-1) + 2(N-1) \longrightarrow L.L.$$

will take place.

Now, if any of the above two changes has occurred, then test No. 3 will always be answered "yes" and a type-out of the cell will follow. Thus, the programmer will always be aware of any internal automatic changes in an accumulator cell. Following the type-out there is a return to "I Bell."

The type-out following any automatic change may also be used by the INTERMAP Programmer to map out his memory. It will eliminate the manual computation, and may be used to prepare charts.

The following example presents the actual type-ins for a problem which requires:

1. The use of a maximum of six accumulators, or $A_{max} = 6$.

2. Five accumulators of varying orders of Nmax and Mmax.

Acc. No. 1 -
$$5x6$$

Acc. No.
$$6 - 4x4$$

Acc. No. 2 -
$$16 \times 16$$

Acc. No. 5 -
$$9x9$$

Acc. No. 3 -
$$12x13$$

3. The accumulators are to occupy as little as possible space in the memory. The Last Location of one accumulator should therefore be immediately followed by the First Location of the next.

SAMPLE TYPE-IN AND TYPE-OUT

6 su s				1	sy0506	s-w0416 s		
1	4.16	4.74	5	6	5	6	sy0404	s-w0476 s
6	4.76	4.u6	4	4	4	2	sy1616	s-w0500 s
2	5.00	9.78	16	16	16	5	sy0909	s-w0980 s
5	9.80	11.24	9	9	9	3	sy1213	2-w1126 s
3	11.26	14.12	12	13	12			
Acc.	Acc. F.L. L.L. N M Nmax A YNNMM —WLLTT						-WLLTT	

This example demonstrates various points.

- 1. The use of six accumulators permits the storage of data starting from location 04.16, 17.
 - 2. Fewer than Amax accumulators may be set.
- 3. The physical location of the data in the memory is independent of the sequence of the accumulator numbers.
- 4. The First Location of Acc. 6 was typed in after the Last Location of Acc. 1 was computed and typed out.

All INTERMAP Operation Routines have safeguards to prevent the inadvertent destruction of data. One of these safeguards is the Amax-Test. This test prevents data from being written into Accumulator Cells, or data being interpreted as Accumulator Cells.

Also, on reading in the INTERMAP Program those locations which may contain Accumulator Cells (03.u to 03.u7 and 04.00 to 04.19) are never disturbed, and l≤Amax≤7 is always set to its maximum of Amax=7. However, the First and Last Locations of Accumulator No. 1 are set at 04.20 and 17.u6 respectively, thus permitting any operation involving just one accumulator (No. 1) without calling in this Operation Routine first.

TYPEWRITER INPUT-OUTPUT ROUTINE (OP = .02)

In this routine all input and output takes place through the typewriter to and from any part of an accumulator.

This routine may also be used as a matrix shrinker or, generally to rearrange data within an accumulator.

This routine contains a double precision floating point conversion routine (Decimal to Binary and Binary to Decimal). Provisions are made so that the binary to decimal conversion may be used independently of input/output operations. For instance, if off-line prepared decimal data has been read into an accumulator from punched tape, it may then be converted to binary by the use of this Operation Routine.

Operations are initiated by two type-ins as follows:

First type in - ANNMM(tab)s where $0 \le A \le Amax$,

and N and M are the order of the matrix.

Second type-in - \pm H.II.JJ where $0 \le II \le .u7$ and

 $0 \le JJ \le .u7$ and

9≤H≤z

Both type-ins are preceded by one or two bells respectively. In case of either type-in the characteristic C on the G15D neon panel will show .02 in binary which is the op code number for this Operation Routine.

PUNCHED CARD - INPUT - OUTPUT ROUTINE (OP=.03)

In this Operation Routine all input and output takes place through the medium of punched cards. The order of a matrix to be read in is also contained on a card, called the First Card.

Each deck of cards, on input or output consists of:

One First Card.

which takes the place of the Label on

data tape

Data Cards,

which contain one decimal double precision, floating point number each, in addition to its row and column number, i and j respectively. These cards may be in any sequence.

One Last Card,

which serves to terminate any input

operation.

The operations which are possible with this Operation Routine are the Input and Output of a:

> matrix row, or column

or parts thereof.

Manual instructions consist of two type-ins:

First Type-in:

 \pm A (tab) s where $0 \le A \le A$ max and

Second Type-in: ± H.II.JJ, where 0<II≤u7

0<JJ≤u7

x≤H≤Z

Both type-ins are preceded by one or two bells respectively. For either type-in the characteristic C on the G15D neon panel will show: ON, ON, which is a binary "Three" and is equal to the op-code number of this Operation Routine.

CARD OUTPUT

Card output consists of a First Card, Data Cards and a Last Card. The sequence of the punch out is the same as on type out, namely by columns. The punch out of zero-elements is optional.

CARD INPUT

Every deck of cards must start with a First Card and end with a Last Card. The Data Cards, however, may be in any arbitrary sequence.

Identically to the typewriter Input Output Routine the input (and output) of specific portions of a matrix may be performed. However, on input, punched cards provide flexibility as demonstrated by the following example: Assume an accumulator of dimensions N and M and given a manual instruction: -ZIIJJ. This means the input of elements in the following range of i and j: -

II \leq i \leq N, where II > 1 and JJ \leq j \leq M, where JJ > 1

It follows that it is not only unnecessary to arrange the cards in sequence, but there is also no need to remove from the deck those portions of the matrix which are to be omitted. This permits the Punched Card - Input Output Routine to be used to sort data. Also, the correction of errors merely requires the reading of the corrected cards between the same First and Last Card.

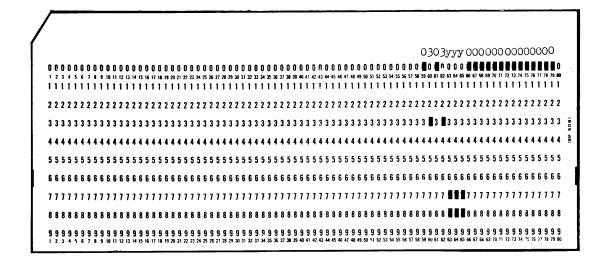
PUNCHED CARD FORMAT AND PREPARATION

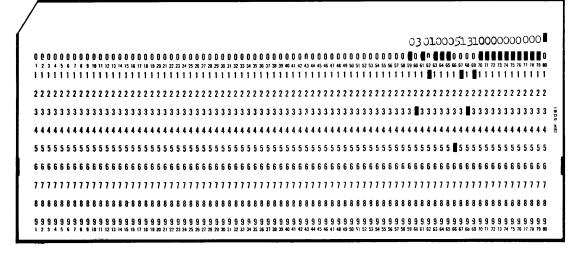
In Figure 5.4.2 are samples of the three types of cards described above. All three cards have the same format and use columns 59 to 80 as shown in Figure 5.4.1 below:

	Columns																			
	51 - 60	61 - 70							71 - 80											
	1234567890	1	2	3	4	5	6 7	8	3 9	0	1	2	3	4	5	6	7	8	9	0
First Card	NN	M	м	Z	Z	Z	0 0	(0	0	0	0	0	0	0	0	0	0	0	+
Data Card	i i	j	j	0	0	0	ΕE	Ι	D D	D	D	D	D	D	D	D	D	D	D	±
Last Card	0 0	0	0	у	у	у	0 0	(0	0	0	0	0	0	0	0	0	0	0	+
			_	 - F`	laį	g 🔻	-													
								L												

Figure 5.4.1 - INTERMAP Card Format

Note the Flags in columns 63 to 65 by which the program distinguishes the three types of cards.





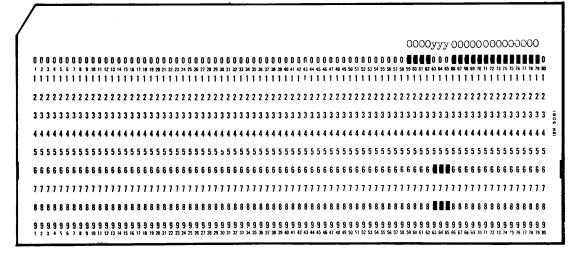


Figure 5.4.2 - Three Types of INTERMAP Punched Cards

PROGRAM CONTROL CARD

One program control card (to be mounted on the program drum of the IBM 026) is needed to prepare any one of the three types of cards. Figure 5.4.3 shows a sample program control card

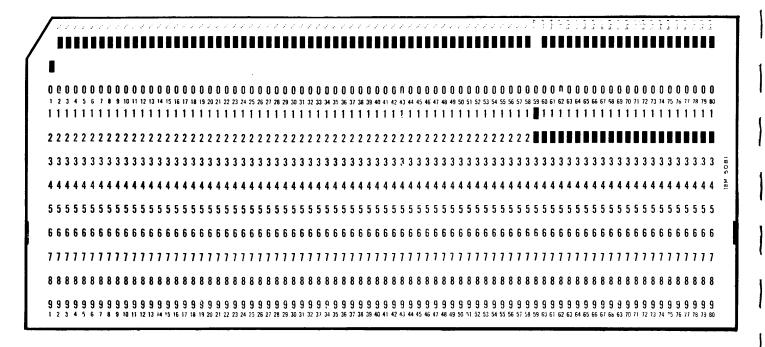


Figure 5.4.3 - IBM 026 Program Control Card

CA-2 ACCESSORY AND AUXILIARY EQUIPMENT

Figure 5. 4. 4 shows the 80:80 CA-2 wiring diagram for use with INTERMAP. The types and numbers of auxiliary card equipment to be used depends on the requirements of each G15D installation and on equipment availability.

For instructions with regards to board wiring and connecting the auxiliary equipment to the CA-2 Accessory, refer to the CA-2 Reference Manual.

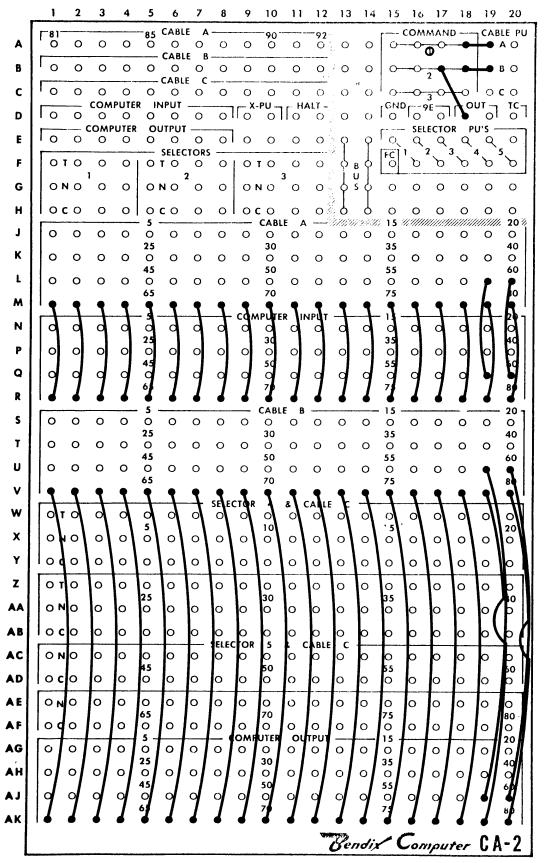


Figure 5.4.4 - CA-2 Wiring Diagram

The punched card output may be substituted by a printout from the IBM 402 for which Figure 5.4.5 and Figure 5.4.6 show the board wiring diagrams for the CA-2 and the IBM 402 respectively.

The punched paper tape for the IBM 402 printer should be prepared as follows:

- 1. Wire CA-2 and 402 board as shown on respective diagrams, and insert board in each respective unit.
 - 2. Punch paper tape for 402 printer as follows:
 - a. Cut off tape at line 102.
 - b. Clip upper left corner.
 - c. Punch following holes on tape.

<u>Line</u>	Column
4	1
47	12
55	1
98	12

- d. Glue ends as indicated.
- e. Install on IBM 402.



1DM [†]	IBM 402, 403 ACCOUNTING MACHINE CONTROL PANEL	
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 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
5 FIRST READING 15 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	A 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	DIGIT SELECTORS CO-SELECTOR I PU'S
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	B 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	98765432101120
	C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	$\begin{array}{c ccccccccccccccccccccccccccccccccccc$	A A A A A A A A A A A A A A A A A A A
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		POSO ZO PO O O O O O O O O O O O O O O O O
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	F 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	OMIC O GOO O O O O O O O O O O O O O O O O
5 2ND LINE ZONE ENTRY 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	G 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	COMPARING EXIT H 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	OMA 0 73 P35 10 0 10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 3RD LINE ZONE & PRINT ENTRY 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	-HEADH-SINGLECARD-
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	J O O O O O O O O O O O O O O O O O O O	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	K O O O O O O O O O O O O O O O O O O O	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	L 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 SECOND READING 15 20 -	M O O O O O O O O O O O O O O O O O O O	-SKIP TO - 2 Z O CO 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
45 50 55 60	N 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	020000000000000000000000000000000000000
5 THIRD READING 15 20	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	ေ 3 ေ
45 50 55 60	P 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	(FC OR 0 IMPULSE)
5 NORMAL ALPHAMERICAL PRINT ENTRY 20 20 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	Q O O O O O O O O O O O O O O O O O O O	050 0 0 0
5 NUMERICAL PRINT ENTRY 20	NUMERICAL PRINT ENTRY 40	06000
COUNTER EXIT	S O O O O O O O O O O O O O O O O O O O	0 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
$\left \begin{array}{c c} J_{2}AJ \end{array}\right \left \begin{array}{ccccc} J_{2}BJ \end{array}\right \left \begin{array}{ccccccccccccccccccccccccccccccccccc$	7 48	8 2 2 5 6 A B A B A B A B C D C D C D C D C D C D C D C D C D C
	U	
	V	70100 ₩ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
010 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	$\begin{bmatrix} w & \circ &$	0110 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
	X ONO O O O O O O O O O O O O O O O O O	COUNTER EXIT SUPPRESSION +
	Y 0 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	Z O O O O O O O O O O O O O O O O O O O	P=SW NEGATIVE BALANCE TEST EXIT
$\left \begin{array}{c cccccccccccccccccccccccccccccccccc$	AA 48 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	NB-AC NEGATIVE BALANCE CONTROL
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9 9	BB AD 1 1 8C 1 1 1 8D 1 1 1 BC 1 1 1 BB	2 2 2 6 6 6 4 4 4 8 8 8 2 2 2 6 6 6 6 8
$ \begin{vmatrix} 2c & & & & & & & & & & $	CC 25 1 2 30 THIRD EADING 2 20 IN 2	O O O O O O O O O O O O O O O O O O O
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0		E O O O O O O O O O O O O O O O O O O O
45 50, 55 60 0000000000000000000000000000		CR SYMBOL EXIT
(63-65)	 	CHANNEL ENTRY TOTAL
		YCLES
<u>(59-60)</u> <u>(61-62)</u>	нн <u>(66-67)</u> <u>(68-79)</u> <u>(83)</u> оо	1 1
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 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 42 43 44	45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
Zero Suppression # 15, 18, 23, 27, 40 Must Be Raise	ed On Numeric Side FORM LAYOUT (821)	(BC OR 10 IMPULSE) Figure 5.4
	IIII	

1.5 Page 5.4.7

REPORT HEADINGS_____ FIELD HEADINGS ____ CARD COLUMNS ____ TYPE BARS ____ SHORT HMRLCK LEVERS LONG HMRLCK LEVERS HAMMERSPLIT LEVERS ____ DEMOUNTABLE TYPE ____ CLASS OF TOTAL_

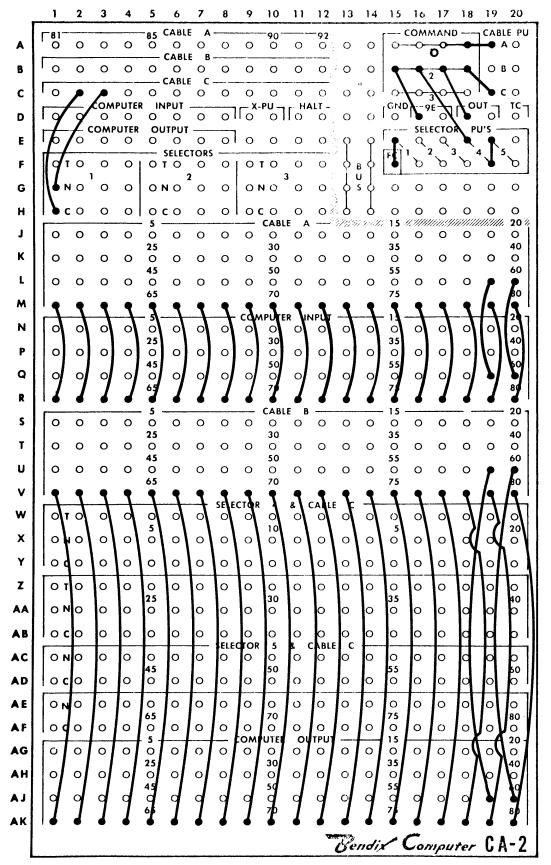


Figure 5.4.6

INCREMENTER (OP=.04)

The purpose of the Incrementer is to permit continuous and automatic operation of INTERMAP. The Incrementer takes the place of what in other interpretive systems would be performed by an index register. The INTER-MAP Incrementer is an independent Operation Routine. It modifies INTER-MAP Instructions permanently and not at the time they are executed.

The Incrementer's main purpose is to facilitate unattended running of INTERMAP Sequences over long periods of time. In most of these cases, the use of magnetic tape equipment may be assumed. Emphasis is therefore placed on functions which are most useful if magnetic tape is used for data storage. These are contained in modes 0 and 1.

Other functions of the Incrementer are defined by modes 2 and 3.

The instruction code used with the Incrementer falls into the category of Special Instructions, see Section 3.1, meaning that the significance of the various fields in the instruction is different from all other INTERMAP Instructions. Only the OP-code functions in the same manner as in all other INTERMAP Instructions.

The following are the four modes in which the Incrementer operates:

- C = 0 Increment Source Mode
- C = 1 Increment Destination Mode
- C = 2 Modifier and Unconditional Transfer Mode
- C = 3 Accumulator Order Mode

In explaining these four modes the following definitions will be useful.

- .ab = The location in line 18 of the INTERMAP Instruction or dummy to be incremented or modified.
- .cd = The location in line 18 of the incrementing or modifying instruction.

All these locations are relative, i.e. equal to 108-TT. See sec. 5.1.

INCREMENT SOURCE OR DESTINATION MODES: C = 0 and 1

Definitions:

(A)cd = Increment, $1 \le A \le 7$

(SRC)cd = .0ab

 $(DST)cd = Limit \le 127$

 T_S and T_0 are defined below

In these modes the Source (when C=0) or the Destination (when C=1) of an instruction in .ab is modified by (A), the operand, of the Incrementing Instruction in .cd. The incremented field, $(SRC)_{ab}$ or $(DST)_{ab}$ is then compared to the Destination of the incrementing instruction which is the limit to which incrementation takes place. If the limit has not been exceeded, the next instruction to be executed will be in cd+1; and the incremented instruction is returned to .ab. If the limit has been exceeded, the Source or the Destination respectively will be initialized or set unconditionally equal to .001 and the next instruction to be executed will be in cd+2.

As was mentioned above, these incrementing modes were specifically designed to facilitate magnetic tape operations. There is a basic difference between initiallizing a magnetic tape Source or a magnetic tape Destination. A matrix stored on magnetic tape under a given File Number may be read an unlimited number of times. However, only one matrix may ever be written following one File Number. The initiallization of a magnetic tape Destination would cause data to be written over previously recorded information. This is avoided by a corresponding change in the First File Number, as explained below.

To insure proper magnetic tape operation the T_S and T_D fields in the instruction should conform to the table below.

Mode		nt Source = 0	Increment Destination C = 1				
Medium	(T _S) _{cd}	$(T_{ m D})_{ m cd}$	(T _S) _{cd}	(T _D) _{cd}			
MAG. TAPE	0	0 (x	≠ 0	$(T_D)_{ab}$			
ALL OTHERS	0	0 (x	≠ 0	0			

x) The value of (T $_{\rm D}$) $_{\rm cd}$ has no significance.

The flow chart in Figure 5.5.1 illustrates the functioning of the two incrementing modes. It will be noted in Part II of the flow chart that alternatives other than the ones outlined in the above table exist for the values of

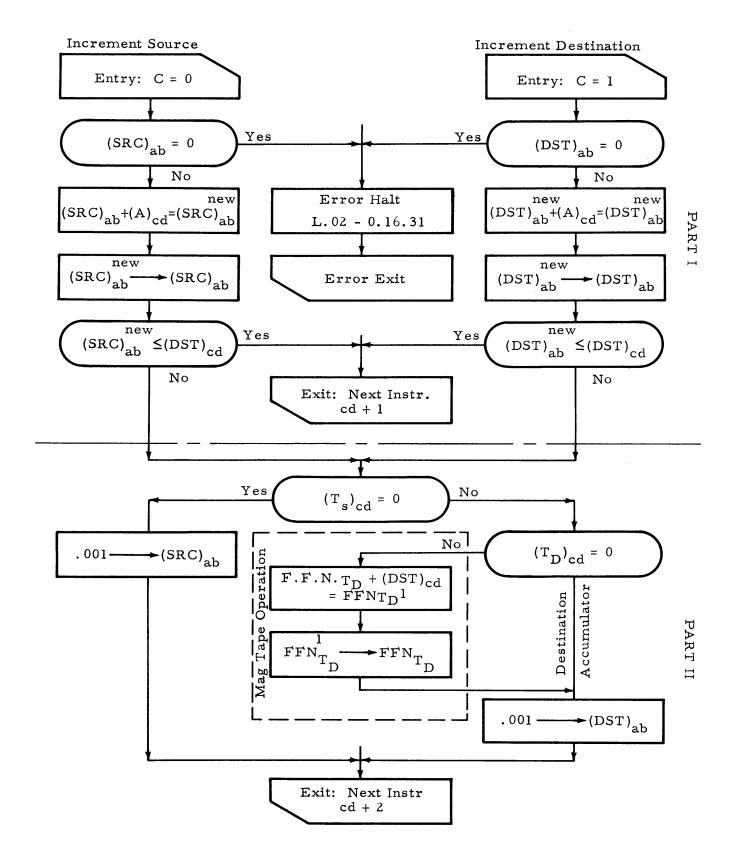


Figure 5.5.1 - Incrementer: Flow Chart for Increment Source or Destination Modes

 $(T_S)_{cd}$ and $(T_D)_{cd}$. The effect of any of these alternatives on the incremented instruction and a First File Number is predictable from the flow chart, however, no practical use has been found so far for any.

It has been stated above that the Limit or (DST)_{cd} may not exceed 127, thus the number of repetitions of a single loop is limited to 127. The following example will illustrate both a method for repeating a loop more than 127 times as well as the mechanics of changing the First File Number to retain continuity of an operation which results in the writing of data on magnetic tape.

Example	intermap Sequence							<u>e</u>
	OP	С	A	T _S	SRC	$^{\mathrm{T}}\mathrm{_{D}}$	DST	
- 10	20.	0.	0.	0.	000.	0.	001	Read punched tape into Acc. 1.
11	21.	0.	l.	0.	000.	0.	000	Invert matrix in Acc. 1.
12	20.	0.	0.	0.	001.	2.	001.	Acc. 1 to MTA/2/001.
13	04.	l.	l.	1.	012.	2.	100	$(DST)_{12} + (A)_{13} \le (DST)_{13}$?
L ₁₄	04.	2.	0.	0.	010.	0.	010	YES: Unconditional transfer to Instruction ab = 10.
15	04.	0.	1.	0.	018.	0.	002	NO: $(SRC)_{18} + (A)_{15} \le (DST)_{15}$?
16	04.	2.	0.	0.	010.	0.	010	YES: Unconditional transfer to ab = 10.
17	00.	0.	0.	0.	000.	0.	000.	NO: End of sequence. Return to I. P. P. R.
18	00.	0.	0.	0.	001.	0.	000	Dummy Counter. (SRC) = 001.

Given that the First Available File Number is 80 and that $F.F.N_2 = 79$, then each time the instruction in .12 is executed, a matrix is written following a File Number = $(FFN)_2 + (DST)_{12}$. (Section 1.4 for File Number System)

Thus the File Number used in the

1st loop is 79 + 1 = 80, and in the 2nd loop is 79 + 2 = 81 and in the 100th loop is 79 + 100 = 179

After the 100th loop the test in .13; namely $(DST)_{12} + (A)_{13} \le (DST)_{13}$? will be answered "NO," as a consequence of which the following will take place:

$$FFN_2 + (DST)_{13} = 79 + 100 = 179 \text{ and } 179 \longrightarrow FFN_2, \text{ also } .001 \longrightarrow (DST)_{12}$$

After incrementing the second counter the program returns to .10 and the next File Number used in the

101st loop is
$$179 + 1 = 180$$
, and in the 200th loop is $179 + 100 = 279$.

An important rule which has to be observed in these two modes is that a zero-field may not be incremented. As explained in Table 3.1.4 the change of a Source or Destination from zero to non-zero is not an incrementation but is equivalent to changing the interpretation of the entire instruction. An instruction to increment either a zero Source or a zero Destination will be terminated by an Error Halt, as shown on the flow chart.

MODIFIER AND UNCONDITIONAL TRANSFER MODE C = 2

Definitions:

(A)
$$_{cd}$$
 has no significance $(SRC)_{cd} = .0ab$ $(DST)_{cd} = .0ef$ T_{S} and T_{D} have no significance

In this mode the non-zero fields of the modifier in .ef replace the corresponding fields of the instruction in .ab and the control is transferred to ab. Thus INTERMAP Instructions can be modified other than through constant increments.

In the special case where $(SRC)_{cd} = (DST)_{cd}$ it can be seen that no modification will actually occur and that the net effect of this instruction is an unconditional transfer to .ab.

ACCUMULATOR ORDER MODE C = 3

Definitions:

$$(A)_{cd}$$
 = Accumulator $1 \le A \le Amax$

$$(SRC)_{cd} = .0ab$$

$$(DST)_{cd} = .0gh$$

 $^{\mathrm{T}}\mathrm{_{S}}$ and $^{\mathrm{T}}\mathrm{_{D}}$ have no significance

In this mode the N and M associated with the accumulator specified in (A)_{cd} will be substituted into the Source and Destination respectively of the instruction specified in (SRC)_{cd}. Following this an unconditional transfer to the instruction specified in (DST)_{cd} takes place.

This mode provides the link between incoming data of varying dimensions and an INTERMAP Sequence written in parametric terms.

An Error Halt will occur when A > Amax.

STORE AND TRANSFER ROUTINE (OP = .20)

This routine transfers data (matrices) from a Source to a Destination, each one of which may be an Accumulator, Magnetic Tape or Punched Tape. The Store and Transfer Routine is not an algebraic routine and has no operand. In a Store and Transfer instruction (A) has no significance and should be made equal to zero.

The data being transferred by this routine may consist of double precision words of any type or format. Such data may come from a punched tape source. Refer to section 1.3 on Labels and data tapes.

The transfer of data may be performed in four modes, namely:

C = 0 Clear and Add Mode

C = 1 Clear and Subtract Mode

C = 2 Clear Mode

C = 3 Transpose Mode.

Compatibility Test:

Whenever the Destination is an accumulator a compatibility test has to be passed (see section 2.4). The purpose of such test is to ensure that:

- 1. The Source data can be accommodated in the Destination Accumulator.
- 2. That no information outside of the Destination Accumulator is inadvertently disturbed.

FORCED TRANSPOSITION

If the Destination is on tape, a test is performed to ensure that the number of rows does not exceed 54. When N > 54 the length of a block of tape would exceed 108 words which is in excess of the capacity of the G15D's Input/Output buffer. In this instance the matrix will be automatically transposed and the label will be flagged to indicate such a forced transposition. It can be shown that if N > 54 then M < 54.

The purpose of the flagged label is to unconditionally offset the effect of the forced transposition if this tape is used as Source Tape. No other Operation Routine accepts a flagged label tape.

The Store and Transfer Routine treats flagged label tapes in a logically correct way, though in some instances it may not achieve the intended purpose of the programmer. These cases occur when both Source and Destination are on tape and are described in Table 5.6.1.

CLEAR AND ADD AND CLEAR AND SUBTRACT MODES (C = 0 and C = 1)

In these modes a single transfer of data from Source to Destination takes place, with a row to row and column to column correspondence. In the Clear and Add mode no change is performed on the data. In the Clear and Subtract Mode a sign change of every element in the matrix takes place on transfer.

CLEAR MODE C = 2

In this mode a null matrix is transferred to the Destination irrespective of the contents of the Source. The order of the null matrix is the same as that of the Source.

The Clear Mode is designed to clear accumulators. It will, however, function logically with any other Destination. Generally, to clear a Destination accumulator the INTERMAP Instruction should be written with Source equal to Destination. This guarantees that the Source and the Destination are uncoditionally compatible.

TRANSPOSE MODE C = 3

In this mode square as well as rectangular matrices may be transposed provided that the following conditions are met:

- 1. The Source and Destination may not be the same accumulator.
- 2. If the Destination is an accumulator which is part of an arrangement of staggered accumulators, this accumulator must be large enough so as not to require a change of 2Nmax(D) when the compatibility tests are passed (see section 2.4).

These conditions must be met in order to retain meaningful data. Where these conditions cannot be met the use of the Matrix Transposition Routines (OP = .24) is required.

Table 5. 6. 2 shows the possible combinations of data sources and destinations in algebraic INTERMAP Instructions.

For an N x M matrix

Source: Tape	Destination: Tape							
	C = 0	C = 1	C = 2	C = 3				
Flagged Label: N blocks of M elements	A duplicate of flagged label and N blocks	As in C = 0, but including a sign change	A duplicate of the flagged label and N blocks of zeros	A duplicate of the label without the flag and the N blocks of data				
Unflagged Label: M blocks of N elements	A duplicate of unflagged label and M blocks	As In C = 0, but including a sign change	A duplicate of the unflagged label and M blocks of zeros	A duplicate of the label but with a flag and M blocks of data				

Table 5. 6. 1 - Store and Transfer Routine: Tape to Tape Operation

	From: (SRC)	To: (DST)		Remark	s
1)	Accumulator			3
2	Accumulator	Mag. Tape		2	
3	J	Punched Tape		2	t.
4)	Accumulator			3
5	Mag. Tape	Mag. Tape	1	2	
6	J	Punched Tape		2	
7	۱	Accumulator			3
8	Punched Tape	Mag. Tape		2	
9	J	Punched Tape		2	

Remarks:

- 1. Source and Destination may not be on the same tape unit.
- 2. No part of the memory, except the input-output buffer (L. 19) is affected.
- 3. The compatibility test has to be passed before data is transferred. After transfer $\rm N_D$ = $\rm N_S$ and $\rm M_D$ = $\rm M_S$

Table 5. 6. 2 - Combinations of SRC and DST in an algebraic INTERMAP Instruction

DECIMAL DATA TAPE INPUT ROUTINE (OP = .26)

This Operation Routine operates only in one mode, the Clear and Add Mode, and permits only one operation; namely:

From: Source - Flexowriter punched data tape

To: Destination - Accumulator

For a discussion of Flexowriter Labels and data tape refer to section 1.3.

This Operation Routine is the same as the Store and Transfer in the following respects:

- a. The same compatibility tests are performed,
- b. The Coding rules are the same, and
- c. Error Halts function in the same manner.

No data tape of a matrix having more than 53 rows can be read. Reference to the instructions for preparing Flexowriter tape on page 20 of the INTERCOM 1000 Operating Manual will show that the Z or W- codes which are required at the end of each data block (or column) prevent the full use of all 108 words in the input output channel.

MATRIX INVERSION AND EQUATION SOLVER (OP = .21)

This routine operates on the matrix specified by the Accumulator Field of an INTERMAP Instruction. It may be operated in one of two modes:

INVERSION MODE - (C = 0)

Only a square matrix may be inverted. If, however, a number of column vectors are stored along with the matrix such that an $N \times M$ matrix results where M > N the Inversion mode produces an inverse in the first N columns and solutions in the last (M-N) columns.

Example: Given:

$$A_{x1} + B_{x1} = +Q$$

$$C_{x1} + D_{x2} = R$$

then

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} Q \\ R \end{bmatrix}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} Q \\ R \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} Q \\ R \end{bmatrix}$$

however, if the original equations are written thus:

$$A_{x1} + B_{x2} - Q = 0$$

$$C_{x1} + D_{x2} - R = 0$$

and the following matrix is "inverted," namely

then the resulting matrix in the accumulator is

$$\begin{bmatrix} a & b & x_1 \\ c & d & x_2 \end{bmatrix}$$

this can be expanded, so that if the original equations were several systems such that

$$A_{x11} + B_{x21} = Q$$
 $A_{x12} + B_{x22} = S$
 $C_{x11} + D_{x21} = R$ $C_{x12} + D_{x22} = T$

then after "inverting" the matrix

the result is

a b
$$x_{11}$$
 x_{12}
c d x_{21} x_{22}

In recapitulation the Inversion mode may produce the solutions to simultaneous equations in addition to computing an inverse.

EQUATION SOLVER MODE C = 1

In this mode equations may be solved as shown in the above example with the following exceptions:

1. The order of the matrix must be such that

2. No inverse is produced and in the result

$$\begin{bmatrix} a^{1} & b^{1} & x_{11} & x_{12} \\ c^{1} & d^{1} & x_{21} & x_{22} \end{bmatrix}$$

the elements a¹, b¹, c¹ and d¹ have no meaning and must be disregarded.

COMPUTING TIMES

The advantages of the Equation Solver Mode over the Inversion mode is in its speed. The following formulae express the computing time in minutes in terms of the order of the matrix.

(1)
$$t = .004 \text{ N}^3$$
 for an inversion of an Nth order matrix, having N x N non-zero elements.

(2)
$$t = .002 N^3 + .004 N^2$$

for the solution of one set of N simultaneous equations.

For matrices with a predominance of zero elements these computing times are very sharply reduced. It is difficult, however, to express these mathematically.

The table below shows some actual computing times for two sample matrices. The "Susie Matrix" is a square matrix having no zero elements at all, and the "Mimi Matrix" has three non-zero elements to a row (on either side of the main diagonal).

	S	usie Matrix	''N	limi Matrix''
N & M	Inversion	Equation Solver \otimes	Inversion	Equation Solver
5	0.5 min.	0.4 min.	0.2 min.	0.16 min.
10	4.0 min.	2.4 min.	0.8 min.	0.5 min.
15	13.2 min.	7.6 min.	1.8 min.	l.l min.
20	32.0 min.	17.4 min.	3.3 min.	1.9 min.
25	63 min.	33.2 min.	5. 2 min.	2.9 min.
27	79 min.	41.8 min.	6.1 min.	3.3 min.

 $[\]bigotimes_{M = N+1}$

MATRIX INVERSION AND EQUATION SOLVER (OP = .21)

Technical Information

THE INTERMAP INSTRUCTION

Fields 4, 5, 6 and 7 should be equal to zero, however, their contents will be ignored.

C = 2 and C = 3 will cause error indications.

MATRIX TRANSPOSITION ROUTINE (OP = . 24)

The Matrix Transposition Routine transposes a matrix within one accumulator without benefit of buffer storage, provided compatibility requirements are satisfied.

Ordinarily, compatibility tests are performed between the matrices defined in the SRC and the DST fields of the INTERMAP Instruction. In the transposition instruction these fields are zero and the test is performed on the operand, which is contained in the accumulator specified by the field A. The Matrix Transposition Routine, therefore, effects only the operand and its accumulator.

In discussing the compatibility tests of the Matrix Transposition Routine, it is convenient to think of the matrix A as the source and A^T as the destination.

One condition controlling transposition within an accumulator is that the size of the operand accumulator must be large enough for the transposition of a square matrix whose dimension is the largest of the two dimensions of the operand matrix.

CASE I:

Assume that accumulator (a) in FIGURE 5.9.1 contains a 3 x 5 matrix which is to be transposed. Obviously, 2Nmax which is the horizontal or column increment may never be changed without completely changing the matrix. The transposition has to be performed on a 5 x 5 square matrix which will not fit the present shape of the accumulator, whose Last Location is 04.68 and not 04.72 as would be required in this case. This operation will terminate in an Error Indication.

CASE II:

The transposition of a 5×3 matrix in Accumulator (a) fails for the same reason as in Case I.

CASE III:

The transposition of any matrix in Accumulator (a) whose largest dimension is equal or smaller than 4 is possible.

Cases I to III describe the operation in the Regular Mode. However, there is also a Permissive Mode in which these operations would have had different consequences.

In the Permissive Mode the Matrix Transposition Routine will change the Last Location so as to make the accumlator rectangular for the given 2Nmax if the execution of the Transposition Instruction depends only on the Matrix Transposition Routine. It follows, that in this mode Cases I and II above would not fail and the Last Location would be changed to L. L. = 04.78, as shown in (b). Cases III could be restated to the effect that: A matrix in accumulator (a) whose largest dimension is equal or smaller than 5 may be transposed in the permissive mode.

Of course, even the Permissive Mode cannot guarantee that every matrix may be transposed within its own accumulator. A 3×6 matrix in accumulator (d) or a 6×3 matrix in accumulator (c) demonstrates such a case. It can be seen by the dotted lines that such a matrix could not be transposed in these accumulators as shown.

In general, if:

 L_S = The largest dimension of the operand

 M_{D} = The horizontal dimension of the operand

Then error indications will occur when:

 L_S Nmax

 $L_S M_D$

This Operation Routine has been written to detect every illegal instruction in the Regular Mode. In the Permissive Mode more leeway is given. However, a programmer uses this mode at his own risk, because changing (Increasing) the Last Location of an accumulator may jeopardize valid data in an immediately adjoining accumulator.

Since there are no distinguishing marks to a staggered accumulator, it is also the programmer's responsibility to exercise care in transposing staggered accumulators. Assume that the accumulator shown in (b) contains two staggered accumulators, the First Location of one being 04.00 and of the other 04.08 and the Last Location for both being 04.78. and both of these accumulators contain 4×5 matrices. It would be impossible to transpose the lower one of the two matrices, but on transposing the upper one the first row of the lower matrix will be destroyed.

It can be seen that even though matrix transposition in itself presents no difficulties from a computational point of view, it is logically quite complex. The programmer will do well to familiarize himself with accumulator maps in order to visualize the effect of an instruction.

The following is a summary of the modes:

Regular mode: C = 0 Transpose

C = 1 Transpose and negate

Permissive mode: C = 2 Transpose permitting change of Last

Location

C = 3 Transpose and negate, permitting change
 of Last Location

04. 00	. 16	. 32	. 48	. 64	04.00	. 16	. 32	. 48	. 64		05.00	. 16	.32	. 48	
. 02	. 18	. 34	. 50	. 66	. 02	. 18	. 34	. 50	. 66	•	. 02	. 18	. 34	. 50	† —
. 04	. 20	. 36	. 52	04. 68	. 04	. 20	. 36	. 52	. 68		. 04	. 20	. 36	. 52	
. 06	. 22	.38	. 54		. 06	. 22	. 38	. 54	.70		. 06	. 22	.38	. 54	 <u> </u>
. 08	. 24	. 40	. 56		. 08	. 24	. 40	. 56	.72		. 08	. 24	. 40	. 56	— ·
. 10	. 26	. 42	. 58	<u>_</u>	. 10	. 26	. 42	. 58	. 74		. 10	. 26	. 42	. 58	 + -
. 12	. 28	. 44	.60		. 12	. 28	. 44	.60	. 76		. 12	. 28	. 44	.60	 -
.14	.30	. 46	.62		. 14	.30	. 46	.62	. 78		. 14	.30	. 46	. 62	
		(a)					(b)						(c)		
]	L _S > M	$^{\mathrm{I}}\mathrm{D}$	

Figure 5. 9. 1 - Examples of Matrix Transposition (Sheet 1 of 2)

.05. 00	. 08	. 16	. 24	. 32	.40	. 48	. 56	
. 02	. 10	. 18	. 26	. 34	. 42	. 50	. 58	
. 04	. 12	. 20	. 28	. 36	. 44	. 52	.60	
. 06	.14	. 22	.30	. 38	. 46	. 54	. 62	
 		 	l ∟	 				
(d)								

L_S > Nmax

Figure 5. 9. 1 - Examples of Matrix Transposition (Sheet 2 of 2)

MATRIX MULTIPLICATION ROUTINE (COMPREHENSIVE) (OP = . 25)

As in all other algebraic Operation Routines the operand (Premultiplier) is the contents of the accumulator specified in the third field (A) of the INTERMAP Instruction. The operator (Post-multiplier) is designated by the Source and the product by the Destination.

Any one of these three may be a Source or a Destination:

Accumulator

Magnetic Tape

Punched Tape

and the combinations and remarks shown in Table 5. 6. 2. generally apply.

Compatibility tests ensure that the product can be accommodated in the Destination, if it is an accumulator; or in the input-output buffer, if it is magnetic or punched tape. Tests are also performed to ensure that the basic rules of matrix multiplication are satisfied, namely:

nxm mxp nxp

 $A \times B = C$

The Matrix Multiplication actually contains two distinct multiplication systems. One, the column system, generates the product matrix by columns and the other, the row system, generates the product matrix by rows.

A list of the possible cases is shown in Table 5.10.1. Also note that the Matrix Multiplication Routine operates in four modes, namely:

C = 0 Normal mode

C = 1 Scalar

C = 2 Pre-multiplier mode

C = 3 Post-miltiplier mode

THE NORMAL MODE

INTERMAP was written to operate with the greatest degree of efficiency in G15D installations which have no auxiliary equipment available. Particular emphasis was placed on permitting the multiplication of matrices of such order so that only two or even one matrix may be stored on the memory drum at any one time. These are represented by the following cases:

1.
$$A_1 A_2 = A_2$$

			~ .	<i>a</i> 3			
Remarks to normal mode	Case	C = 0 Normal	C = l Scalar	C = 2 Pre- Diagonal	C = 3 Post- Diagonal		
These cases use the column system	A ₁ A ₂ =A ₃	OK	OK	OK	OK		
and require no buffer storage,	$A_1 A_1 = A_2$	OK	OK	EH	EH		
since the Destination is always	$A_1 T_{IN} = A_2$	ОК	OK	OK	OK		
different from either the oper-	$A_1 A_2 = T_{OUT}$	OK	OK	OK	ОК		
and or the Source	A ₁ A ₁ =T _{OUT}	OK	OK	EH	EH		
These cases require some buffer storage since the Destination and Source are the same accumulator	A ₁ A ₂ =A ₂	ОК	OK OK	OK OK	Illegal but no E. H. pro- vided EH		
or input/output buffer. The column system is used	A ₁ T _{IN} =T _{OUT}	OK	OK	OK	EII		
The new system is used in these two cases which have to be buffered since the	A ₁ A ₂ =A ₁	OK	EН	EH	OK		
product and the pre-multiplier use the same accumulator	A ₁ T _{IN} =A ₁	ОК	EH	ЕН	OK		
punched). Subscripts in accumulators	accumulator tape (mag. or dicate number of sused in operate Halt, See sec. 3	In these three modes the column system is used and no buffer storage is required.					

Table 5. 10. 1 - Table of Operations - Matrix Multiplication (Comprehensive)

2.
$$A_1 A_2 = A_1$$

3.
$$A_1 T_{IN} = A_1$$

The first two cases permit the multiplication of two matrices of up to 20 x 20 dimensions. The third permits the multiplication of two 28 x 28 matrices. Neither one of those three cases requires that the product be a tape output.

The third case is not efficient from a computing time point of view since it makes p - passes of the post-multiplier tape. Between each pass the tape is completely reversed. Thus if the Source is on punched paper tape, this tape has to be mounted on a tape magazine in order to function properly.

Faster alternatives for cases 2 and 3 would be:

4.
$$A_1 T_{IN} = A_2$$

5.
$$A_1 A_2 = T_{out}$$

6.
$$A_1 T_{IN} = T_{OUT}$$

These cases, however, do involve tape operation which is less efficient without auxiliary equipment.

On discussing the permissible order of the multiplier it was assumed, so far, that the product matrix be of a magnitude which permits its storage in the G-15D memory. However, as long as the product is in a tape-Destination the limit imposed on its magnitude is only in its number of rows. Since output takes place by columns, the maximum number of rows is 54, which is equivalent to block of tape 108 words long.

The following are permissible operations:

Operand	Source	Tape Destination
54 x 1	l x 54	54 x 54
54 x 5	5 x 100	54 x 100

Product tapes of a magnitude as the above examples may not be read back into the memory in their original form. Such a tape, if used as a Source, will induce an Operation Routine to read M blocks of tape as specified in the Label. If it is required to use such tapes it will be necessary to partition them manually and splice in new Labels. Such products should be on punched paper tape rather than be written on magnetic tape.

If it is required to obtain a decimal type-out of such data tape, it is suggested to do this under the control of an INTERCOM 1000 program. It

would be possible to partition such tapes through a special INTERCOM 1000 program. Such a program would, however, require that the entire tape be repunched. This program is not provided as part of the INTERMAP package.

THE SCALAR MODE

In this mode, the Matrix Multiplication Routine assumes element A_{11} in the operand (A) to be a scalar multiplier for the matrix specified by the Source. The operand matrix may be of any magnitude, including a 1×1 matrix. All the rules that apply to our $N \times M$ matrix also apply to a 1×1 matrix.

It follows from the above that the following two cases are not permissible in this mode; and are terminated by Error Halts (see Table 5. 10. 1):

$$A_1 A_2 = A_1$$

$$A_1 T_{1N} = A_1$$

Two cases which may not be generally useful are:

$$A_1 A_1 = A_2$$

$$A_1 A_1 = T_{out}$$

In these cases a number will be squared if the operand is a $l \times l$ accumulator, and if greater than that a matrix will be multiplied by its first element.

PRE-DIAGONAL MODE

In this mode the pre-multiplier or operand is a vector column whose elements are the elements of a diagonal matrix. The diagonal pre-multiplier may also be thought of as a vertical projection of a N \times N diagonal matrix. The dimensions of the pre-diagonal are N \times 1. However, the horizontal dimension M of the operand is of no consequence, in this mode and several diagonal pre-multipliers may be stored consecutively in one accumulator, as for example:

OPERAND ACCUMULATOR

a ₁₁	^b 11 06. 08	c ₁₁
a ₂₂	b ₂₂	c ₂₂
06. 02	06. 10	06. 18
a ₄₄	b ₄₄	c ₄₄
06. 06	06. 14	06. 22

The 3 x 3 matrix in the operand accumulator consists of 3 columns, each of which is the vertical projection of the 3 x 3 diagonal matrices A, B and C respectively.

It will depend on the First Location (F. L.) of the Operand Accumulator which of the three diagonal matrices will be the pre-multiplier.

If: F. L. = 06.00 the premultiplier is A, if

F. L. = 06.08 the premultiplier is B, and if

F. L. = 06.16 the premultiplier is C.

The First Location may be changed under the control of the Accumulator Selection Routine and the Incrementing Routine.

The same principle applies to a diagonal post multiplier if stored in an accumulator, as well as for scalar multipliers.

POST-DIAGONAL MODE

A post diagonal multiplier is also a vertical projection of a diagonal matrix and is in most respects similar to a prediagonal multiplier. The post diagonal multiplier is in the Source and may therefore also be on tape.

However, if on tape it would not be possible to have several post diagonal multipliers on the same tape, similar to the above example of the operand accumulator. Irrespective of the contents of the Label only the first block of tape (or column) following the label would be read.

In each, the pre-diagonal or the post diagonal mode, there are four cases which are not permissible and which will be terminated by Error Halts. These are listed in Table 5. 10. 1. In the case of A_1 A_1 for which Error Halts exist in all four modes.

MATRIX MULTIPLICATION ROUTINE (Short) (OP = . 22)

Similarly to the comprehensive Matrix Multiplication Routine (OP=. 25), see Section 5. 10, the Operand is the premultiplier, the Source is the post-multiplier and the product is in the Destination.

A Source or a Destination may be an:

Accumulator

Magnetic Tape

Punched Tape

and the combinations and remarks shown in Table 5. 6. 2 generally apply with the exception of mode 3 as explained below.

Compatibility Tests ensure that the product can be stored in the Destination, if it is an accumulator; or in the input output buffer, if it is magnetic or punched tape. Tests are also performed to ensure that the basic rules of matrix multiplication are satisfied, namely:

Below is a list of the operations and their respective modes which may be performed by this multiplication.

MODE	OPERATION				
C = 0	$A_1 A_2 = A_3$				
	$A_1 A_1 = A_2$				
C = 1	$A_1 A_2 = T_{out}$				
	$A_1 A_1 = T_{out}$				
C = 2	$A_1 T_{in} = A_2$				
C = 3	$A_{1} T_{in} = T_{out}$				

All of these operations are also found in the comprehensive multiplication routine on Table 5.10.1. By limiting the scope of the short multiplication routine its efficiency and computing speed have been improved. In return for this some error detection devices had to be eliminated, and care has to be exercised in coding INTERMAP Instructions having an OP-Code of .22.

The mode controls the type of operation which this Operation Routine will attempt to perform. The following are the characteristics of the FOUR modes:

Mode	Source		Destination
C = 0	Accumulator	to	Accumulator
C = 1	Accumulator	to	Tape
C = 2	Tape	to	Accumulator
C = 3	Tape	to	Tape

The following examples will illustrate what may take place if the instruction is incorrectly coded.

Example 1:

Incorrect instruction 22. 1. 1. 0. 002. 0. 003.

Here the intention is to multiply accumulator 1 by accumulator 2 and store the product in accumulator 3. However, C = 1 overrides this and assumes that the Destination is File Number 3 on magnetic tape Unit 0, which is the basic unit. Since the basic magnetic tape may not contain data, but may contain the MTA-2 Version of INTERMAP, this is prevented by Error Halt number 7, see sec. 3.3.

Example 2:

Incorrect instruction 22. 0. 1. 0. 002. 1. 003

The product of accumulators 1 and 2 is to be stored on magnetic tape number 1, File Number 3. In this case, where C = 0, the program will store the product in accumulator 3, irrespective of T_D , if the operation passes the compatibility test. However, the portion of the memory which may be destroyed is predictable.

If the instruction had been written thus; 22. 0. 1. 0. 002. 1. 009, where accumulator number 9 does not exist, the consequences are rarely predictable. Such an instruction may destroy the INTERMAP Program and Control and necessitate starting the problem from the beginning.

Operations in the Tape to Tape Mode, C = 3, differ somewhat from other modes in that line 18 or a part thereof is used for buffer storage. This has to be taken into consideration when storing an INTERMAP Sequence in line 18.

The number of elements stored in the line 18 buffer (starting from 18.00,01 and up) is equal to the number of rows in the premultiplier.

For example, if the operand is a 27 x 27 matrix, then 27 elements or $2 \times 27 = 54$ words are needed for buffer storage in line 18, using up locations (18.00.....53). This leaves room for a 54 instruction INTERMAP Sequence occupying (18.u7.....54).

MATRIX ADDITION ROUTINE (OP = . 23)

The Addition Routine, adds to the operand in "A" a matrix from a source "SRC" and stores the sum in the destination "DST."

The Source and Destination may be any of the three:

Accumulators

Magnetic Tape

Punched Tape

permitting all the combinations outlined in Table 5. 6. 2.

The Addition Routine operates in four modes. If

[A] = operand

[B] = Source matrix

[C] = Destination matrix, then when

C = 0 the function is [A] + [B] = [C]

C = 1 the function is -[A] + [B] = [C]

C = 2 the function is [A] - [B] = [C]

C = 3 the function is -[A] - [B] = [C]

Similarly to all other INTERMAP Operation Routines all operations which have an accumulator as a Destination have to pass the compatibility tests. Generally, matrices can be added only when of the same order. In INTERMAP this rule is expressed as follows:

a. The dimensions of the Source matrix govern in all instances.

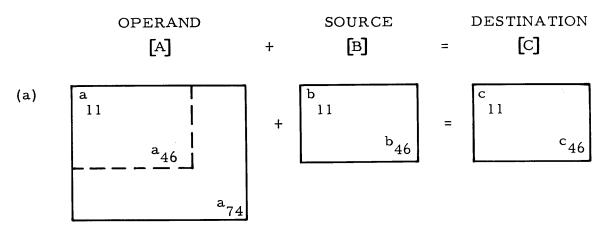
b. An addition can be performed if: $^{\bigotimes}$ $N_0 \ge N_S$ and $M_0 \ge M_S$

c. After the operation is complete $N_S = N_D$ and $M_S = M_D$

d. The portion of the Operand matrix which is being added to the Source Matrix is of the same dimensions as the Source Matrix.

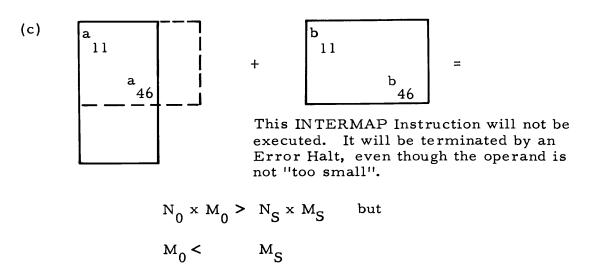
 \bigotimes_{N_0} and M_0 refers to the Operand.

These rules are illustrated in the examples below:



In this case the dimensions of the operand [A] are larger than the dimensions of the source [B]. The dimensions of the destination [C] are identical to those of the source. Finally, $a_{11} + b_{11} = c_{11}$

This is a special case of (a) above, where the dimensions of the operand and the source are identical.



DOT PRODUCT MULTIPLICATION ROUTINE: (OP = . 10)

The Dot Product Routine is the same as the Matrix Addition Routine with the exception that like elements are multiplied by each other instead of being added. Technically, the Dot Product Routine is a modified Matrix Addition Routine. It follows exactly the same rules and tests which have been outlined in the description of the Matrix Addition Routine.

Some of the uses the Dot Product Routine may be put to are:

- a. Extract any row, column, diagonal or other group of elements by "Multiplying" a matrix by a matrix which contains one's in the positions of the elements to be extracted and zeros in all other positions.
- b. If it was required for instance to extract the main diagonal of a matrix of an order not to exceed N , store a unit matrix of order N in the Operand, use the Dot Product Multiplication.
- c. To expand on (b) above, if the diagonal was all that was required in this operation, make Source and Destination the same accumulator. The diagonal may be "shrunk" into one row or column by the use of the Matrix Shrinker which is part of this Typewriter Input Output Routine.

EQUIPMENT

The equipment which may be used with INTERMAP is of two kinds:

- a. Equipment, which depends on which of the three versions of INTERMAP being used, and
- b. Optional Equipment which will operate with any of the three versions of INTERMAP, depending on availability and the requirements of the problem.

The following is the required equipment for the three versions of INTERMAP:

The G-15D Version:

This version requires no auxiliary equipment at all. Both program and data tapes are read from the same photo-reader and the two INTERMAP magazines have to be interchanged manually.

The PR-1 Version:

This version requires one auxiliary photo-reader (PR-1, C=1), and the change from the first to the second magazine is performed automatically. The first INTERMAP magazine is mounted on the G-15's basic photo-reader, and the second INTERMAP magazine is mounted on the master PR-1.

The MTA-2 Version:

This version requires one magnetic tape unit for the storage of the INTERMAP program. The G-15's basic photo-reader is available for data. The Selector Switch on this magnetic tape unit has to be set at 4 or C=0.

OPTIONAL EQUIPMENT

The total auxiliary equipment, both optional and required, which may be used with the INTERMAP Program is not to exceed:

- 4 magnetic tape units (MTA-2)
- 3 auxiliary photo-readers (PR-1)
- 1 card coupler and assorted card equipment (CA-2)

The maximum number of optional equipment not including the required equipment, which may be used with any of the three INTERMAP versions is as shown below:

> G-15D Version: Up to: 3 MTA-2's $(1 \le C \le 3)$ 3 PR-1's $(1 \le C \le 3)$

1 CA-2

3 MTA-2's $(1 \le C \le 3)$ PR-1 Version: Up to:

2 PR - 1's (C = 2 or 3)

1 CA-2

3 MTA-2's $(1 \le C \le 3)$ 3 PR-1's $(1 \le C \le 3)$ MTA-2 Version: Up to:

1 CA-2

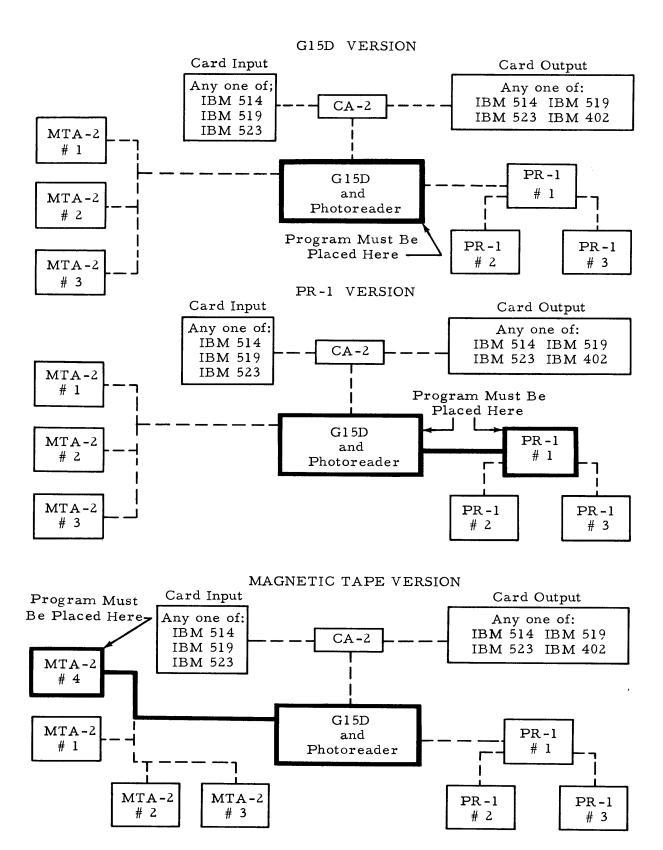


Figure 6.0.1 - G15D And Accessory Equipment Which May Be Used With INTERMAL Indicates Required Equipment For Program Only Indicates Optional Equipment For Data Use Only

DIFFERENT INTERMAP VERSIONS:

The difference between the programs of the three INTERMAP versions is mostly in their respective Controls. Other than in the control portion of the program, only two blocks of tape differ in all of the three versions, see check sums in section 0.3.

Functionally, the difference between these versions is only in the type of equipment used as a source for the INTERMAP Program with resulting advantages in speed and facility of handling.

By speed we mean the time required to search the library type program tape for each Operation Routine (op-code) and to change paper tape magazines. By far the fastest version in this respect is the magnetic tape version. Computing speeds for the various algebraic matrix routines are the same.

Facility of handling refers to the need or its absence for the interchanging of program and data magazines on the photoreaders. However, truly automatic, that is unattended operation is impossible unless independent media of input are available for program and data.

The most efficient operation is possible with the Magnetic Tape Version, because its access to the INTERMAP Program is the fastest. It permits the exclusive use of the basic G-15D photoreader for the input of data. The G-15D typewriter and tape punch are, of course, the output media.

The auxiliary equipment requirements for fully automatic operation depend on the type of problem. If a problem requires considerable intermediate storage beyond the capacity of the G-15D memory), data has to be stored on magnetic tape from where it can be recalled automatically under program control.

Figure 6.0.1 shows the many different combinations in which auxiliary equipment may be assembled to serve any particular need.

It is, however, important to remember that there is no INTERMAP operation which cannot be performed on the basic G-15D with no auxiliary equipment whatsoever.

THE G-15D VERSION:

This version is contained on two tape magazines both of which have to be read from the basic G-15D photoreader. Magazines are changed manually. Whenever a change of magazine is required, an automatic type out will occur followed by a HALT.

If the first magazine is to be replaced by the second magazine, the type out is: .zzzzzz2 and if the second magazine is to be replaced by the first magazine, the type out is: .zzzzzzl.

The first three blocks of tape contain the INTERMAP Control. Only one of the blocks, however, will be permanently stored in line 03 of the memory.

THE MAGNETIC TAPE (MT-A2) VERSION

This version of INTERMAP requires that the INTERMAP Program be stored on magnetic tape. The Selector Switch on the magnetic tape unit which is used for this purpose must be set at 4 (or C=0). This tape unit cannot be used simultaneously for data and INTERMAP Program.

The operation of storing the INTERMAP Program on magnetic tape is separate operation, which is described under: "Program to Magnetic Tape Loading Routine." (Section 7.2.1).

The system of serial numbers and file codes used by INTERMAP is compatible with Applications Project 61 Magnetic Tape Service Routine. The INTERMAP Program may be written on any magnetic tape which stores programs or data in a format compatible with this service routine.

THE PR-1 VERSION

This version of INTERMAP is contained on two tape magazines similar to the G-15D Version. INTERMAP Control is part of the first magazine.

The first magazine is mounted on the basic G-15D Photoreader and the second magazine is mounted on the auxiliary photoreader (PR-1) which responds to a characteristic of C=1. This is the master PR-1.

During the op-code search of the library tape the program automatically switches from one magazine to another. No need exists to remove these tape magazines unless it is required to read data tape from any one of the two photoreaders.

The PR-l version may be expanded to use more auxiliary photoreaders if the length of the program tape should be increased by the addition of additional Operation Routines.

THE SCOUT AND MEMORY DUMP

The Scout and the Memory Dump are two independent service routines which for convenience are produced on one tape. Both routines are not part of the basic INTERMAP Program and may be read in independently. They may be used with any of the three available INTERMAP Versions.

THE SCOUT

The Scout, which operates from line 19, has the following functions:

- 1. It will type-out the number (.ab) of the last INTERMAP Instruction executed following, which it may do either (2) or (3) below.
- 2. Return to the control of INTERMAP P.P.R. to type out the decimal equivalent of the last executed INTERMAP Instruction (See Error Halts, (c) and (d), sec. 3.3).
- 3. Or, by typing in .cd, the location in line 18 of any other INTERMAP Instruction, the program will proceed to execute this instruction, followed later by .cd + 1 on the Multiple Instruction Level or a return to I.P.P.R. on the Single Instruction Level.

The Scout may be put to the following uses:

- 1. To determine the last instruction executed in an INTERMAP Sequence.
- 2. To return unconditionally to INTERMAP P.P.R. for changes in the program or for the purpose of investigating intermediate results.
- 3. To transfer unconditionally to any other part of an INTERMAP Sequence.

The Scout does not permit the type-in of INTERMAP Instructions. It may only return to I.P.P.R. or to an instruction which has already been stored in line 18. It will not change the Instruction Level.

THE MEMORY DUMP

The Memory Dump permits the storage of a variable portion of the G15D memory onto a self loading punched paper tape. The dump may consist of a minimum of three blocks and may have up to sixteen blocks depending on the quantity of data stored. It will always dump:

Line .03 - Control and Accumulator Cells

.04 - Accumulator Cells and Data

.18 - Intermap Sequence and Data

If the dump is performed after the complete execution of an INTERMAP Instruction, it will when reloaded resume operations with the next instruction in the INTERMAP Sequence. If the program was operating on the Single Instruction Level, it will on reloading return to I.P.P.R.

PROGRAM TO MAGNETIC TAPE LOADING ROUTINE

This is a service routine used only for the MTA-2 Version of INTER-MAP. Its purpose is to load the INTERMAP Program onto magnetic tape. It is independent of the basic INTERMAP Program and is not effected by changes in the INTERMAP Program Tape. The program tape may be of any length, that is, it may contain fewer or more Operation Routines than are shown in Table 7.1.1. Table 7.1.1 is an example of what would take place if a punched tape containing all the Operation Routines shown is loaded on magnetic tape. Note, that only serial number line 116 has a block searching routine, also, that the last block on the program tape must be the Last Loader or Return Flag (OP=.64).

As this loading routine writes programs on magnetic tape, it compares the check sum of each block on magnetic tape with the check sum obtained when reading the punched tape. Check sums are typed out and may be compared to the list of check sums in Section 0.3 of this manual. It follows that this routine is self-checking provided no read-in errors occur.

After completion of the loading operation, the next available serial number is typed out. The magnetic tape is then reversed and the read head is aligned with the loader for the INTERMAP P.P.R. (OP=.00).

File Number	Storage									
101	First Available File Number line with block searching routine									
	Type: 101 (tab)s and the following lines will be stored by the Program to Magnetic Tape Loading Routine.									
	The File Numbers are contained in the loaders which do not have a block searching routine except as shown (File No. 116).									
102	Loader	for	OP=00	followed	bv	4	lines	of :	program.	
103	11		OP=01	11	11	3	11	11	11	
104	11		OP=02	11	11	5	11	11	11	
105	11		OP=03		11		11	11	11	
106	11		OP=04		11	2	1.1	11	11	
107	11	11	OP=10	11	11	3	11	11	11	
108	11	11	OP=20	11	11	3	11	11	11	
109	11	11	OP=21	11	11	3	11	11	11	
110	11	1.1	OP=22	11	11		1.1	11	11	
1 111	11	11	OP=23	11	11	3	11	11	11	
112	11	11	OP=24	11	11	3	11	11	11	
113	11	11	OP=25	11	11	7	11	11	11	
114	11	11	OP=26	11	11		11	11	11	
115	11	11	OP=64	Last	Lo	oad	er or	· Re	turn Flag.	
116	Next available serial number line with block searching									
				yped out.						-

Table 7.1.1. Storage of MTA2 Version on Magnetic Tape