PROJECT NO. 105

TITLE:

INTERCARD PROGRAMMING MANUAL FOR THE BENDIX G-15D

COMPUTER

EQUIPMENT AFFECTED: G-15D

APPLICATION SECTION

PREPARED BY:

Bendix Computer Division

CHECKED BY:

APPROVED BY: T. Yamashita

DATE:

18 June 1959

TABLE OF CONTENTS

		Pago
I,	Introduction —	
II.	Data Input	20.00
	Sample Data Sheet	20.10
III.	Command Input	
IV.	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7	
	a) Memory	40.00
	b) The Codo Shoot	· ·
	Sample Code Shoot	
	c) The Assombly Program	
	d) Operations	
	1) Arithmetic 2) Fixed Point Arithmetic — —— 3) Indexing 4) Logical Control — —— 5) Block 6) Card Input — —— 7) Card Øutput 8) Printed Øutput — —— 9) Magnetic Tapos	40.31 40.32 40.33 40.34 40.35
	o) Machine Lenguage Subroutines	40.40
v.	Timing	
VI.	Computer Operations —	80,00
	Table of Operations	90.00

for

THE BENDIX G-15D COMPUTER

I. Introduction

These notes assume that the reader is familiar with the G-15D computer and with IBM card equipment. Necessary information for the use of the G-15D with a floating point interpretive system is presented.

Input is from IBM cards via the IBM 523. Output is both IBM cards via the 514 punch and line printing on the IBM 402 printer. All input-output is 100 lines and/or cards per minute.

Commands are written with alphabetic operation codes, numeric addresses, and alphanumeric comments. An assembly program converts these symbolic commands to the proper binary machine language and punches binary cards. Programs are then run by reading the binary program cards followed by the data cards into the IBM 523 reader. Results are printed on the IBM 402 printer.

This system is intended as a replacement for Intercom using high speed input-output via the IBM card and printing devices. However, the interpretive routines are written in a very efficient manner and new commands added so that the computing speed will be about 4 times faster than Intercom. The overall capacity of the G-15D will be increased 10 to 20 times over that of the present mode of operation.

II. Data Input

Data is prepared on the data sheets provided in either fixed or floating point. These are keypunched on IBM cards, 5 pieces of data to a card. The floating form is the base 50 method already in use with Intercom on the G-15D and familiar to those who remember the IEM 701. The first location must be punched on every card, blank number fields are not stored, and a zero is entered by writing one zero as the first digit in the number field. If the signs are positive, they should be left blank, and it is not necessary to write trailing zeros in number fields. Fixed point numbers may be positive or negative and have a maximum value of 99. These are written as a two digit number with the remainder of the field blank. (see the data sheet and cards).

FIRST S N, S N ₂ S N, S N ₃ S N ₄ S N ₅ SEG. N 00000000000000000000000000000000000	FIRST 45 LOC. 35 25 S	NUMBERS	IDENT.	PAGEOFJOB NOPROGRAMMER
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1				
Loc.				
111111111111111111111111111111111111	<u></u>		N, S	N ₂ S N ₃ S N ₄ S N ₅ SEQ. N
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	2 3 4 3 6 7 6 8 10 11 12 13 14 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2	79 30 31 32 33 34 35 16 31 38 39 31 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
77777777777777777777777777777777777777	4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5	4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4	4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8	7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8	7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
######################################				

III. Command Input

- a) Programs are written initially on code sheets. These are keypunched on IBM cards, one command to a card including comments. The code used is discussed in detail in Section IV. These are called the symbolic cards (see the code sheet and cards).
- b) Binary cards are prepared by the Assembly program. One channel of commands (approximately 100) are punched on 9 cards with a check sum. These can be read into the computer at execution time by the operator or by a stored command (see the binary cards).

IV. Programming

a) Momory

The interpreter and input-output routines occupy long channels 00 \$\ightarrow\$ 06 and 4-word channels 20 -22. While reading or punching binary cards, all of long channels 16 and 17 are used. While reading decimal data cards or printing decimal results, words 00 -49 of long channels 16 and 17 are used. While reading or writing on magnetic tapes long channel 19 and 4-word channel 23 are used. This leaves the following portions of memory available to the programmer:

Long Channels 07-15

Long Channels 16 & 17 (with IBM input-output exceptions)

Long Channel 18

Long Channel 19 (with mag.tape input-output exceptions)

Certain registers located in the interpreter channel 06 are of interest to the programmer and may be addressed by the programmer. These are:

These registers may be filled from cards by writing the correct locations on the data sheets. The contents of these registers may also be printed by addressing the print command. The only other reference to channels 00-06 permitted by the assembly program is the use of the out command to channels 00-03.

The "N" register used in indexing and the AMQ register used in computing may also be addressed by the programmer. These are:

(2000) = N Register (2400) = AMQ Register

There registers may not be filled from data cards and may not be printed except by storing their contents in a long channel.

b) The Code Sheet

- Col.1) An "x" in column 1 halts the card reader. This normally indicates to the assembly program that this is the last card (END) of those to be assembled.
- Col.24,25) These two columns are left blank except where it is necessary to refer to a command with another command. The assembly program assigns the word locations by counting. Any two digits from O1-99 may be written without regard to the order or sequence of the numbers.
- Col.28-30) The two and three letter operation codes are written here. Two letter codes should be written to the right, leaving Col.28 blank.
- Col.32-35) The four digit address is written here. The first two digits indicate the memory channel except when they are blank. If the first two digits are blank, the second two digits refer to another command having the location (Col.24,25) indicated by the latter two digits. For some operations the address has a special meaning and these exceptions are all noted in the operations section.
- Col.37,38) These two columns are left blank except to indicate an indexed command and certain special uses explained in the operations section. If an entry is made, it should always appear as a two digit number.
- Col.46-63) Commonts may be alphanumeric and up to 18 characters (inside paper perforations) may be used including spaces.

 These are not needed to assemble a program.
- Col.70-74) This 5 digit sequence number is for the convenience of the programmer and may be used to properly sort the cards, insert commands, etc. These are not needed to assemble a program and need not be in order for assembly. However, they are printed by the assembly program.

(29-17-H)

																					PA	4GE					.0)F				J) B	1	10). .			_			-
	į	DE	C۲	(VC)	-	···					_ [DA	TE	•						PR	0	GF	₹,	11	11	٨I	EF	₹.								سندات	مجسد			-
100	LASI	LØ	C.	ø	PF	₹.		AD	DI	R.		T	K								C,	ØM N	ME	N	TS	5											T	-		SEO		R
2	<u>-</u>	24	25	28		30	32	<u> </u>			3	3 3	7	38	46																					6	3 7	<u>'0</u>				74
L											٠																															
L	4	<u></u>	_			_						1																									1					
L	\dashv		-			_						+						-																			\downarrow					٠
H	4		4			-					ــــ	+		_																				-			\dotplus					
-	+		\dashv			-		٠.				+																									+					
-	+		\dashv			-						+		-										-												**	+				4	-
-	†		\dagger			1		-				\dagger		\dashv					_																		\dagger		 1			
	\dagger		7			7				ليب	<u> </u>	t	4-	┪						-																	十					
								·																				•••••									İ				<u> </u>	
	T		T																																		T					
-			.									- -		-					- -				-										<u>-</u>			- -			<u> </u>			
CR.													LO	c.	OP	R.	AC	DR	.	NO	EX					(ON	IME	NT	9			!			9	BEQ.	NO.				
0	0	000	00	00	0 0 10 11	12 13 1	14 1 9	0 0	1 18 1	9 29	21 2	23 20	25	ŀ	000	0	N N	34 25	11	1 24			48	47 48	49 5	0 51	u u	34 5	3 36	57 58	\$2 6	81 62	13/61	65 66	67 66	63 10	00	73 74	75 76	17 71	7 8	•
	1	1 1 1 2 2 2										1		1	1				11				1										- 1				1122		İ			
1		2					_						- 1	- 1	2	11				- 1													1			1	33		ĺ			
1		4 4 4		44									- 1		4 4 4				11	- 1			4	4 4	4 4	14	4 4	4 4	14	4 4							44		1			
5	5	5 5 5	5 5	5 5	5 5	5 5 :	5 5	5 5	5 :	5 5	5 5	5 5	5		5 5 5		5 5	5 5	5	5			5	5 5	5 5	5 5	5 5	5 9	5 5	5 5	5 5	5 5	515	5 5	5 5	5 5	5 5	5 5	5 5	5 5	5 9	1
6	6	6 6 6	6 6	6 6	6 6	66	6 6	6 6	6 6	6 6	6 6	6 6	G		6 6 6	П	6 6		Ш	6			1										- 1			-	6 6					
T		777	11	11	71	7 7 7	17	11					-		111	11	7 7			7								7 7					1			- 1	7 7 8 8		1			
1		8	99	99	99	999	9 9	99		3 6		8 8			8 8 8 : 9 9		3 8	0 0		8				8 8 9 9		•							- 1			9 0	9 9	g é	2 9	9 9	9 9)
i	12	9 9 9		# # 1 H L 1	0 11 1 2720	2 13 1	4 15	16 17	16 1	9 2%	21 *2	22/21	وادد	5 2; l·	8 29 3	i i	• • • •	H 35.		• ;	;3 40 41	42 43 44 4	17/46	47 48	49 5	0 51 5	52 53	54.5	3 .6	5/ 58	29 66	61 62	E) 64	65 CA	67 (3	60170	71 72	73 74	15 76	11 71	79 6.	•
	ļ		+		_4_	\downarrow			-	_		_		\downarrow						_																	╀				L	
	L		+			+						L		\dashv						_																	╀				<u> </u>	-
	-		┼			+						┝		+																							十					-
	┝		+		-	+						-		\dagger						_																	十				<u> </u>	•
	\dagger		\dagger			\dagger						-		\dagger		•	-										-										\dagger					
	\vdash		\dagger			\dagger			_			_		+																							†				<u></u>	
			\dagger			\dagger			-			_		†																							Ţ					
	T		†			†							<u> </u>	1																							\int			لبسي		
	+		+-			+					-			+		_																						****				

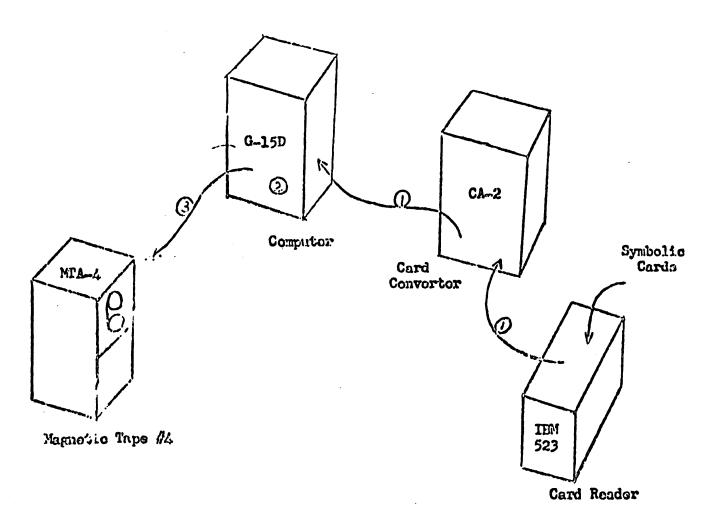
c) The Assembly Program

The assembly program reads the symbolic cards as punched from the code sheets into the G-15D. Locations are assigned beginning at the address of the first card, the origin card, and sequentially increased until another origin card is read into the computer. On this first pass, a location table is propared, and the operation codes found from an operation table. Simultaneously with card roading, the images are written on magnetic tape 4, four cards to a record. The programmer may instruct the assembly program to start a new memory channel of code at any time before the previous channel is filled by an origin (GRG) command. If the GRG is not written, the next channel in sequence and word zero will be assumed by the assembly program. When the assembly encounters an END operation code with an "x" punched in column 1, the address in considered to be the execution test address, and the first pass is terminated. Magnetic tape 4 will now re-wind to the beginning of the commands recorded and Pass II begins the IEM 402 listing, one command to a line, while the tape is being read simultaneously. After all channels of commands have been listed, the binary cards representing the machine language equivalent will punch.

All binary cards punched will be read into the computer by the operator to verify the fact that these are punched correctly. If errors occur, the cards can be punched a second time from magnetic tape 3.

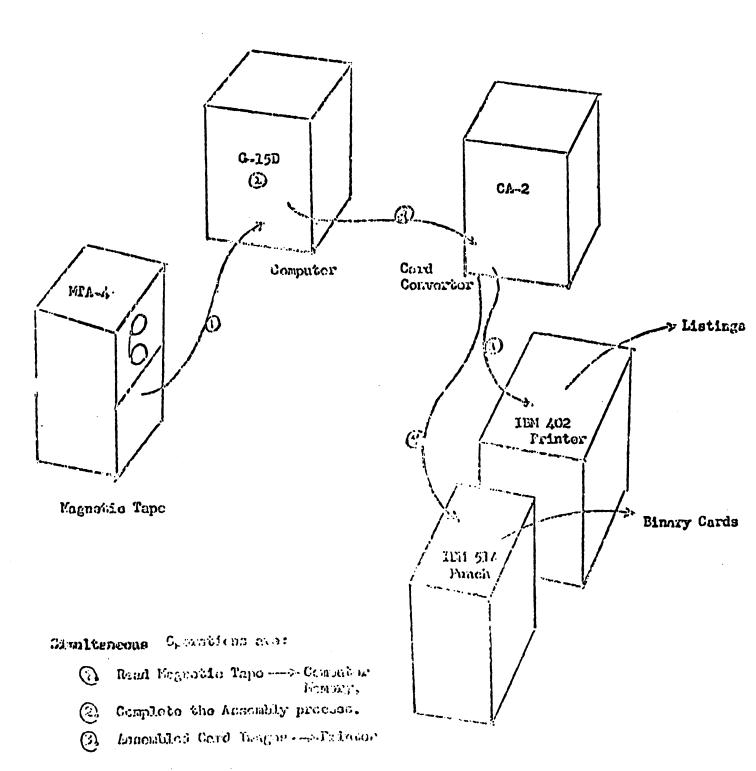
Assembly time is exactly 50 commands per minute, the time being limited to 100 cards per minute input and the 100 line per minute listing speed. A schematic of Pass I and Pass II phases of the assembly program are shown in Figures 1 and 2.

^{*} Location of the first command to be obeyed upon leading.



Three simultaneous operations are:

- (i) Symbolic Cards --> Computer Comovy
- (2) Computer converts 675 & Desirul to Binary.
- 3 Card Images from Memory-Magnetic Tape



A Court own ponens () reasy 300 december.

d) Operations

The operations preceded by an asteriak (*) are indexable. Those preceded by (4*) are indexable only if the index register contains a multiple of 4:

1) Arithmetic:

- * RA Reset and Add. The addressed quantity is stored in the AMQ.
- * RS Rocot and Subtract. The addressed quantity is stored, with its sign changed, in the AMQ.
- * SR Sond Result. The (AMQ) is stored at the addressed location.
- * AD ADd: The addressed quantity is added in floating point arithmetic to the (AMQ). The result is stored in the AMQ.
- * SU SUbtract. The addressed quantity is subtracted in floating point arithmetic from the (AMQ). The result is stored in the AMQ.
- * AP Add Positive. The absolute value of the addressed quantity is added in floating point arithmetic to the (AMQ). The result is stored in the AMQ.
- * SP Subtract Positive. The absolute value of the addressed quantity is subtracted in floating point arithmetic from the (AMQ). The result is stored in the AMQ.
- * MY Multiply. The (AMQ) is multiplied by the addressed quantity in floating point arithmetic. The result is stored in the AMQ.
- * DV DiVido. The (AMQ) is divided by the addressed quantity in floating point arithmetic. The result is stored in the AMQ.
 - SQT SQuare RooT. The square root of the floating point number in the AMQ is stored in the AMQ. The address of this command must be blank.

2) Fixed Point Arithmetic Bg

- XAD Fixed And. The addressed quantity is added in fixed point arithmetic to the (AMQ). The result is stored in the AMQ.
- XSU Fixed Subtract. The addressed quantity is subtracted in fixed point arithmetic from the (AMQ). The result is stored in the AMQ.
- XMY Fixed Multiply. The (AMQ) is multiplied by the first 8 binary bits of the quantity in fixed point arithmetic. The result is shift left 8 binary places and stored in the AMQ.

3) Indexing

- LXR Load Index Register. The addressed quantity is stored in the index register indicated by the two digits in the K field. If K = 00 or blank, all three index registers are loaded from consecutive locations. The addressed first word location must be a multiple of 4 with a one remainder (i.e. 01,05,09 etc).
- *LNR Load N Register. The addressed quantity is stored in the N register.
 - IXR Increase Index Register. The index register indicated by K is increased by the increment in the address field.
- IXN Increase Index Register and M Test. The index register indicated by K is increased by the increment in the address field. This value is then compared to the contexts of the N register. If the values are not equal, calculations proceed. If the values are equal, the index register is set to zero and calculations proceed. This command will prevent a transfer if the next command in sequence is TXN and the values are equal.

4) Logical Control

- 4 TXN Transfer if IndeX not equal to N Registor. Transfer control to the addressed location if the index was not equal to N on the previous command. This command must follow the IXN command.
- 4 TRA TRAnsfer. Control is unconditionally transferred to the addressed location.
- 4* TMA Transfer and Mark A. Control is unconditionally transferred to the addressed location and the location of the next command is marked for use in a return command.
- 4* TMB Transfer and Mark B. Same as TMA to permit two marked places A and B for two levels of sub-routines.
- 4 TZE Transfer if ZEro. Same as TRA if the (AMQ) = 0. If $(AMQ) \neq 0$ calculations proceed.
- 4* TPL Transfer is Plus. Same as TRA if the (AMQ) = (+). If (AMQ) = (-) calculations proceed.
 - Transfer if Tape Operating. Same as TRA if a magnetic tape is in operation. If no tape is in operation, calculations proceed.
 - Transfer if Card Operating. Same as TRA if NEW cards or 402 printer is operating. If no card equipment is operating, calculations proceed.
 - TSS Transfer if Sense Switch on. Same as TRA if the sense switch is on. If the sense switch is off, calculations proceed.
 - Transfor <u>CUT</u> of the Interpreter. Control is transferred to the addressed location in <u>machine language</u>. OUT 0000 may be used to end a job.
 - RMA Roturn to Mark A. Control is returned to the location marked by the TMA command. The address of this command must be blank.
 - RMB Roturn to Mark B. Control is returned to the location marked by the TMB command. The address of this command must be blank.

5) Block

- PDL Process Pata Line. The addressed line is processed starting at word location zero. Word zero is destroyed and the addressed word is the last one changed. The AMQ is unaffected.
- PDA Process Pata & Load AMQ. The addressed line is processed starting at word location zero. Word zero is destroyed and the addressed word is the last one changed. The contents of the addressed location prior to precession is stored in the AMQ.
- CDL Copy Data Line. A block of data is copied from the line indicated by the address to the line indicated by the two digits in the K field. The block copied starts at word zero and continues through the addressed word, this being the last word copied.
- CLL Copy Line to Line. The entire line indicated by the address is block copied to the line indicated by the two digits in the K field.
- CAL Copy AMQ to Line. The (AMQ) is block copied to each word of the addressed line starting in word zero. The addressed location is the last word copied.

6) Card Input

The execution of any card input command is automatically delayed until the previous card or print operation is completed. The address of these commands must be blank.

RBC Read Binary Cards. Nine (9) binary cards are read into channel 17. If the cards being read are binary program decks prepared by the assembly program, the check sum is computed and compared with that on the cards. If no error occurs, channel 17 is moved to the proper program channel and the next binary deck of 9 cards is read. This process automatically continues until the last program deck is loaded into the memory. Automatic computation then starts at the location specified by the address of the END command.

If the cards being read are <u>hinary data</u>, ten (10) cards are read. The first card is the binary data control card. The last 9 cards containing the information are read into channel 17 and the check sum is computed. If no error occurs computations resume with the next command. All of channels 16 and 17 are used by this command.

- RBP Read Binary Cards and Proceed. Nine (9) binary cards are read into channel 17 and computing proceeds simultaneously. All of channels 16 and 17 are used by this command.
- RDC Read Data Cards. Decimal data cards are read, converted to binary and stored at the memory locations given on the cards. Card reading continues until the last card, indicated by an "x" punch in column 1, has been read and stored in the memory. Computations then resume with the next command. Words 00 -> 49 of channels 16 and 17 are used by this command.

7) Card Output

The execution of a <u>cord</u> <u>cutput</u> (PUN) command is <u>not</u> automatically delayed if the previous card or print operation is
in progress. For this reason the (PUN) command should be preceded by the card operation halt (HLC) command if there is any
possibility that the previous card operation is still in
progress.

- PUN Pullch Binary Cards. The addressed line is check cusmed and then punched on nine (9) binary cards. Committing proceeds while the cards are being punched. All of channels 16 and 17 are used by this command.
- HLC Halt if Card coupler is in Operation. Computing is halted until the previous card or print operation is completed.

8) Printed Output

The execution of any print command is automatically delayed until the previous card or print operation is completed. Words 00 -> 49 of channels 16 and 17 are used.

- PRI PRINT. IBM 402 printing starts at the addressed location. Four (4) numbers are printed on a line and zeros print as blanks. Mixed fields of both floating and fixed point numbers are printed with the signs following the numbers. If the K field on the code sheet is blank, the number of words printed is controlled by the print register (0612). An entry in the K field everrides the print register and determines the number of words to be printed. The contents of the print register is unchanged. A complete line is always printed and computing proceeds while the last line is being printed.
- PRC PRint if Checkout. Printing occurs exactly as above during checkout runs. This command is ignored on production runs.
- REP Restore Paper. The 402 paper is restored to a point just below a perforation to permit hole punching for a 3-ring binder. The address of this command must be blank. Computing proceeds while the paper is being restored.
- SPP SPace Paper. The 402 paper is spaced two lines. The address of this command must be blank. Computing proceeds while the paper is being spaced.

9) Magnetic Tapes

Magnetic Tapes 2,3, and 4 are available to the programmer. Tape 1 is the library tape and is normally read only at the beginning of a problem if system programs are required from the library routines. The file protection switch avoids accidental destruction of Tape 1 information. Tapes do not erase old information when writing new information. Control cards are provided to automatically position the tapes to be used to a blank section of tape. These are placed in front of the binary program cards and become part of the program deck.

- WRT MRito Tapo. Channel 19 is recorded on the addressed tape. Computing proceeds and channels 19 and 23 are used while recording.
- RDT Read Tapo. The addressed tape is read into channel 19. Computing proceeds and channels 19 and 23 are used while reading.
- BST Eack Space Tape. The addressed tape is backspaced 1 block. Computing proceeds and channels 19 and 23 are used while backspacing. The previous contents of channels 19 and 23 are destroyed.
- EFT End of File Tape. A file mark is written on the addressed tape. Computing proceeds but channels 19 and 23 are not used.
- SFT Sparch Forward Tapo. The addressed tape is searched forward at high speed, stopping when a file mark is found.

 Computing proceeds while tape is searching but channels 19 and 23 are not used.
- SBT Sparch Rackward Tapo. The addressed tape is searched reverse at high speed, stopping when a file mark is found. The reading head stops in front of the file mark. For this reason a search forward (SFT) command is necessary before reading the first block in the file. Computing proceeds while searching but channels 19 and 23 are not used.
- HLT Halt if Tape Operating. Computing is halted if a tape is in operation. As soon as the tape operation is completed, the computer remains halted for the additional number of drum cycles indicated by the address of this command. (see Table I). After this delay, operation automatically resumes with the next command.
- STT Stop Tape Motion. This command stops any tape operation which may be in progress. Computing proceeds and channels 19 and 23 are not used. The address of this command must be blank.

Table I) Delay Times for Magnetic Tapes

Command which must		Required Delay Following									
be delayed	WRT	EFT	RDT . BST	SFT.SBT							
WRT	0	0	0	8							
EFT	4	0	0	8							
rdt BS T	4	4	0	8							
SFT SBT	4	4	0	0							

Note: Table I applies if the previous tape command referred to the same tape as the command being delayed. Zero cycles are required if a new tape is referred to by a command. The intervening HLT however is still required because 2 tapes cannot operate simultaneously.

Other Notes on the Use of Magnetic Tapes

- 1) A non-zero piece of data must appear in one of the four locations 1900, 1901, 1902 or 1903 before writing L19 on tape.
- 2) An HLT command with 0 delay may be followed by at least 4 or 8 instructions in lieu of the 4 or 8 cycles delay required by the above table. This makes use of the delay time to compute.
- 3) A block of tape reads or writes in approximately 2 seconds. Between 40 and 80 commands can be executed during this time interval. The TTØ command provides for iterative loop executions for the full tape operating time. If this is done the HLT command is still required before the next tape operation. The TTØ command may be followed by at least 4 or 8 commands in lieu of the HLT delay from the table.
- 4) Searching speed is approximately 8 times faster than the readwrite speed. The search time is approximately 0.25 seconds per block.
- 5) Line 19 is cleared by WRT, and L19 cannot be referred to by another command while writing or reading until the process is completed.
- 6) After an EFT command, an SBT command may or may not detect the file mark just written. To insure that a backward search is initiated, use:

EFT HLT	4}	Writes file mark and delays 4 cycles
SBT STT	}	Insures search over mark just written
Set	•	Searches back to previous mark.

•) Machine Language Subroutines

Machine language subroutines are available on binary cards for the required elementary functions. Some of the commonly used routines are written and are available on file. Nemonic operation codes are accepted by the assembly program for convenience. The programmer may elect to place a package of subroutines in channel 19 if magnetic tapes are not used. However, if a magnetic tape operation is to take place, care must be exercised to preserve the subroutine and place it back into channel 19 for further use. These routines use channel 23 for temporary working space.

- 1) Trigonometric Package. This package is available for radian arguments. The number of circles is defined as the number of revolutions subtracted from the argument to reduce the angle to less than 2 N radians.
 - SIN The sine of the floating argument in the AMQ is stored in the AMQ. The number of circles is stored as a fixed point number in 2300. The address of this command must be blank.
 - CØS The cosine of the floating argument in the AMQ is stored in the AMQ. The number of circles is stored as a fixed point number in 2300. The address of this command must be blank.
 - ART The inverse tangent of the floating argument in the AMQ is stored in the AMQ. The address of this command must be blank.
- 2) Logarithmic Package. This package is available for log bases 2,10 and c.
 - LyG The logarithm (power to which base must be raised to equal the argument) of the floating argument in the AMQ is stored in the AMQ. The address of this command must be blank.
 - EXP The exponential (base raised to the argument power) of the floating argument in the AMQ is stored in the AMQ. The address of this command must be blank.

V. Timing

A very good approximation to the time required to assemble a problem is given by:

$$t = \frac{1}{40} + 1 \quad (min)$$

where i is the total number of commands to be assembled.

The time required to run a problem can be approximated by:

$$t = \frac{1}{1200} + \frac{p+c}{100} + 1$$
 (min)

where i = total number of commands to be executed

c = number of cards to be read

p = number of lines to be printed

For more accurate timing, the charts that follow provide cycle times for each operation. (a drum cycle on the G-15D is .0295 sec). Access time to the commands is not included in the charts. For average access, 0.25 cycles must be added for each command executed. Command access may be accurately timed by consideration of the method used by the interpretor. Four commands are copied into channel 22 by a TRA command. This requires 1 cycle if the 4-word group begins with a word location in the range 00-88. The 4-word groups above 92 require 2 cycles. The conditional transfers copy the 4-word group to channel 21 and then test the condition. If the condition is met, channel 21 is moved to channel 22 affecting the transfer. A conditional transfer takes place in 1 cycle if the 4-word group begins with a word location in the range shown in the tables. For any transfer operation for which an actual transfer is effected, access to the first four commands that follow is zero. However all subsequent 4-word groups must be charged with 1 cycle for access.

1) Arithmetic & Indexing Operations

a) Operations requiring access time

The range in the table is defined as the word time of the effective address of the command. Average cycles are for random access.

Ø P	Range i		Execution Cycles	Total Average			
R	Not Indexed	Indexed		Not Indexed	Indexed		
RA	0-93	20-93		1.06	1.26		
RS	0-90	20-90	0	1.09	1.29		
SR	0-96	20-96		1.03	1.23		
AD	0-85	20-85	O If either number zero	2.14	2.34		
SU	0 - 83	20 - 83	l Normally	2.16	2.36		
AP	0-84	20-84	1	2.15	2.35		
SP	0-80	20-80	(see note 1)	2.19	2•39		
MY	0-91	20-91	O If either number zero	2.08	2.28		
DΛ	0-89	20-89	l Normally	2.10	2.30		
LNR	0-70	20-70	0	1.29	1.49		
XAD	0-1	, 1		1.	58		
XSU	0-1	‡ 3		1.56 1.78			
XMY	0-3	L9					
	X1 0-0)6	o	1.93			
	X2 0-0	08		1.	91		
LXR	X 3 0-0) 		1.	95		
	All Ol or	• 05		1.	92		
PDL	0-9	96		1.	03		
PDA	0-9	73		1.06			
CDL	0-9		0	1.03			
CAL	0-9	96		1.03			

NOTE 1). If the difference in the magnitudes of the numbers is greater than 217, one additional cycle is required. Also if more than 5 leading binary zeros are introduced in the fraction by differencing two numbers of equal magnitude; one additional cycle is required.

R.G. Noel 5/17/60

b) Operations requiring no access time

ØPR	FIXED CYCLE TIME
SQT	
IXR	2
IXN	_
CIL	

2) Transfer Operations

a) Operations requiring access time

0	Cycle	for 1 Access	Execution Cycles	Tota Average	l Cycles
	Not Indexed	Indexed		Not Indexed	Indexed
TXN	0-76	20-76		1.20	1.40
TRA	0-88	20-88		1.08	1.28
TMA	0-68	20-68		1.28	1.48
IMB	0-64	20-64	0	1.32	1.52
TZE	0-72	20-72		1.24	1.44
TPL	0-68	20-68		1.28	1.48
TTØ	0-56	DO NOT		1.40	
TCØ	0-52	INDEX		1.44	DO NOT
TSS	0-111			1.52	INDEX

b) Operations requiring no access time

ØPR	FIXED CYCLE TIME
ØUТ	
RMA	1
RMB	

3) Input Output Via Card Coupler

a) Card Input-Output

ØРП	FIXED CYCLE TIME	CARD CYCLE TIME
RBC	44	180 (9 cards)
RDC	50	20/card
RBP	,	Simultaneous with computing. (220 for
PUN	4	9 cards)

b) Printing

¢ PR	FIXED CYCLE TIME	LINE CYCLE TIME
PRT	50	20/line. Last line simultaneous with
PRC		computing.
REP		Paper is positioned simultaneously with
SPP	~	computing.

c) Card Coupler Test

Ø PR	FIXED CYCLE TIME	DELAY TIME
HLC	1	Waits until previous card or print operation is completed.

4) Magnetic Tapes

a) Reading and Writing

Øpr	FIXED CYCLE TIME	DATA TRANSMISSION TIME
WRT		Simultaneous with com- puting. Approximately
RDT	1	67 cycles for a channel of 108 words.
BST		·

b) Searching

ØРR	FIXED CYCLE TIME	SEARCH TIME
SFT	1	Simultaneous with computing. Approxi-
SBT	•	mately 12 cycles/channel

c) Tape Control

Øpr	FIXED CYCLE TIME	EXECUTION TIME
EFT	1	0
STT	1	Tape motion stops simultaneously with computing.

d) Tapo Tost

øpr	FIXED CYCLE TIME	DELAY TIME
HLT	1	Waits until previous tape operation has completed + the number of cycles in the address field of this command.

5) Elementary Functions

¢pr	FIXED CYCLE TIMES	
SIN	1 for arg. = 0	
cøs	3 for small arg.	
LØG	5 normally	
EXP	5	
ART	6 if arg - 1 7 if arg - 1	

VI. Computer Operation

The chart in the following page presents a flow diagram for easy reference when operating the computer. Checkout, production and assembly runs are all loaded the same way. If the sense switch is to be turned on, the PUNCH switch on the typewriter base should be set to the left. For assembly the card punch must be connected to the CA-2 with cards cycled into the hopper and magnetic tapes. 3, and 4 must be turned on. The execution of some operations which produce non-admissable results will cause the computer to ring bells continuously. These are:

<i>∮</i> ₽ R	CAUSE OF TROUBLE	
AD AP SU SP MY	Overflow* Underflow*	
XAD XSU XMY	Overflow*	
DV	Overflow* Underflow* Division by zero	
SIN C¢S EXP	Argument too large	
LØG	Argument = 0 or (-)	
RDC	Intercard channels 00 -> 06 do not check sum.	

If balls ring, or if the computer stops for any other reason, the DEBUG procedure will effect a print-out on the 402 with sufficient information to detect the error.

^{*} If (0615) is zero, bells ring. If (0615) \pm 0, fixes overflow and continues.

Danie Intill-cin

1652	CII1	11	CH ₂	1.12
1656	1771	p	OC.	¥Ÿ
1660	rr	X1	72	2.3
1664	CHEUM	C1	C2	C3

CHWW₁ = Location of last command obeyed. This will print 2222 if the interpreter 4 word channel 22 is destroyed or if the computer is stopped in the middle of printing.

CHWW2 = Location of last transfer and Eark A (TMA). This will print 0404 if no TMA command this executed.

NR = Contents of the N register = (2000)

AMQ = Contents of the AMQ register = (2400). If overflow or underflow or divide check occurs, the contents in the AMQ will be:

MY	Multipliened
DV	Dividend
ΛD	
ΛP	absolute largest
SU	of the numbers
SP	being added.
KIN	Argumont
CC3	
ART	<u></u>
rco	
YMY	
XAD] , .
XSU	Garbago
EXP	

SS = Sense switch status; 00 if off, 01 if on.

ØV = Overflow indicator; 00 if underflow, 8934028 if overflow.

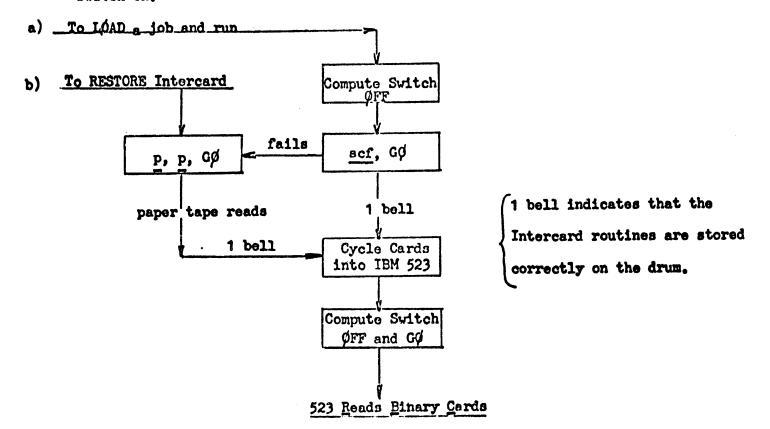
PR = Contents of the FrintRegister = (0612)

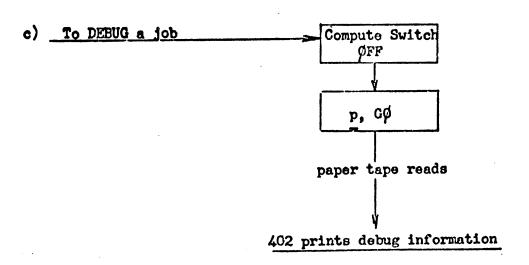
X1, X2, E3 = Contents of the Index Register = (0609), (0610) & (0611)

CKSUM = Channel 00 -> 06 status; 00 if checksum o.k., =0 if not o.k. C1,C2,C3 = Contents and the Control Register = (0613), (0614) & (0615)

OPERATING CHART

In the charts below ϕFF and $G\phi$ refer to the compute switch on the typewriter base. Underscored lower case letters mean to type these letters with the Enable switch on.





The paper tape always backspaces itself. It is never necessary for the operator to manually rewind the paper tape.

	-	TABLE II OPERATIONS
	*RA *RS *SR *AD *SU *AP *SP *MY *DV SQT	Roset & Add Arithmotic Roset & Subtract Sond Result ADd SUbtract Add Positive Subtract Positive Multiply DiVide SQuare Roof
	XAD XSU XAY	Fixed And Fixed Point Fixed Subtract Fixed Multiply
4	LXR LNR IXR IXN	Load IndoX Rogister Joad N Rogistor Increase IndeX Rogistor Increase IndeX & Compare to N
4 4 4 4	TXN TRA TMA TMB TZE TPL TCØ TCS ØUT RMA RMB	Transfer if Index not equal to N TRAnsfer Transfer and Mark A Transfer and Mark B Transfer if Zero Transfer if Plus Transfer if Tape Operating Transfer if Cards Operating Transfer if Sonso Switch on Transfer Out of the Interpreter Roturn to Mark B
	PDL PDA CDL CLL CAL	Process Pata Line Precess Pata Line Precess Pata & Load AMQ Copy Pata Line Copy Line to Line Copy AMQ to Line
	REC RBP RDC PUN HLC PRT PRC REP SPP	Card Input_Output Read Binary Cards Read Binary Cards & Proceed Read Data Cards Punch Binary Cards Halt if Card Coupler Operating PRinT FRint if Checkout Reptors Paper SPace Paper
	WRT RDT EFT BST SFT SBT HLT STT	MRite Tape MeaD Tape ReaD Tape Fad of File Tape Each Space Tape Search Forward Tape Search Backward Tape Halt if Tape Sporating & Delay Stop Tape Motion

[#]Indexable

^{4&}quot; Indoxable in multiples of 4.