pb 250

Cinch Interpreter

PBC 1006

pb Packard Bell Computer

A SUBSIDIARY OF PACKARD BELL ELECTRONICS

1905 ARMACOST AVENUE • LOS ANGELES 25, CALIFORNIA • GRANITE 8-4247

March 15, 1961

PREFACE

The CINCH Manual is provided for information pertaining to coding and does not intend to cover the subject in its entirety. Certain additions, deletions and amendments will be incorporated in later editions.

CONTENTS

I. INTRODUCTION

The Packard Bell PB 250 Computer is a high-speed, solid state, general purpose digital computer with a memory comprised of a group of nickel steel magnetostrictive delay lines, along which acoustical pulses are propagated. To facilitate the rapid programming of scientific and engineering problems and to allow personnel generating these problems to carry out the required programming, a floating point interpretive system, designated CINCH, has been developed.

CINCH uses an arithmetic process known as "floating point", in which the scaling of numbers is handled automatically by the interpretive system. In fixed point the programmer must keep track of radix points when performing arithmetic operations, but by using the floating point feature of CINCH, such problems are eliminated.

II. ELEMENTS

A. MEMORY

The CINCH interpretive memory may contain up to 4,095 words. One word is required to represent a command; two words are required to represent a number. The allowable memory addresses are 0000 through 4,095 sequentially provided a sufficient amount of PB 250 memory is available. Each delay line contains 256 locations. It is the programmer's responsibility to confine the program to conform to the size of the particular machine. CINCH will work for any interpretive size memory up to 4,095 words. Address 0000 refers to the accumulator, therefore neither commands nor numbers may be located in 0000. An address specifies the location of a particular number or command in memory. Numbers must always be stored in even numbered locations. Commands may be stored in either even or odd numbered locations.

B. NUMBERS

All numbers used in CINCH must be less than 1.0×10^{38} in absolute value. Larger numbers formed during computation will cause a halt in computation with a line number of 05 displayed. If computation is continued (by restarting) such numbers will be reduced to less than 10^{38}. Numbers smaller in magnitude than 1.0×10^{-38} are set to zero. Each number input to the computer may consist of the following:

1) a sign
2) up to 10 decimal digits

3) a decimal point

4) an exponent and its sign.

1, 3 and 4 are optional.

The exponent consists of one or two decimal digits, and specifies the power of ten by which to multiply the number. Every number being input must be followed either by a tab code or a carriage return code.

Examples:

14	6+21 means $+6x10^{21}$
-6317	1.2-14 means $+1.2x10^{-14}$
+2.08	+1+6 means $+1x10^{6}$
6.8	-1234567-10 means $1234567x10^{-10}$

When stored in memory, each number takes up two words. The first word must have an even address and it is the programmer's responsibility to make sure that this is always the case.

If the numbers in the preceding example were stored sequentially starting with location 1000, a decimal representation of memory would appear as follows:

Location		Location	
1000	$+.1400000000 \times 10^{2}$	1008	$+.6000000000 \times 10^{22}$
1002	$-.6317000000 \times 10^{4}$	1010	$+.1200000000 \times 10^{-13}$
1004	$+.2080000000 \times 10^{1}$	1012	$+.1000000000 \times 10^{7}$
1006	$+.6800000000 \times 10^{1}$	1014	$-.1234567000 \times 10^{-3}$

If these numbers were to be printed out, the format would be:

.1400000000+02	.6000000000+22
-.6317000000+04	.1200000000-13
.2080000000+01	.1000000000+07
.6800000000+01	-.1234567000-03

C. COMMANDS

The command format consists of up to 8 alpha-numeric characters followed by a carriage return. Tabs (and code deletes) are ignored. A command has the form:

T	OP	I	ADDR	CR

TRACE TAG (T)	T if present, space if not present
OP CODE (OP)	2 numeric characters.
INDEX TAG (I)	1 alphabetic character (A-G) if present, space if not present.
ADDRESS (ADDR)	One, two, three or four numeric characters.
TERMINATION CR	A carriage return follows each instruction.

Commands are normally performed sequentially, starting with a given location. Certain commands will interrupt the normal sequence and cause control to be transferred out of the normal sequence depending on given conditions. The trace tag, where present, indicates that this command is to be traced during the time the program is being executed if the BREAKPOINT switch is down. The index tag, if present, specifies one of seven index registers lettered A through G. The contents of the base of the specified index register are added to the address of the command to form an effective address for execution. The op code specifies what operation is to be performed. The address, where needed, specifies a memory location which enters into execution of the command.

Examples:

Command	Meaning
10 0126	Clear the accumulator and add the contents of location 0126.
02A1348	Set the base of index "A" equal to the contents of location 1348.
22D0046	If the base of index register "D" contains 0012, then the effective address of this command is 0058 and the command reads: Subtract the contents of location 0058 from the contents of the accumulator.
T42 0319	Transfer to location 0319 if the contents of the accumulator are positive and print the location of this command, the command itself, and the contents of the accumulator.

D. ALPHA-NUMERIC FORMAT

A string of any number of Flexowriter characters, excluding colon (:) and semicolon (;), may be stored, three per word. The termination character which is either a colon or semicolon, is stored in the last word.

Example: TITLE PAGE (requires 4 words)

It is important to remember that spaces, carriage returns, tabs, punctuation marks, upper and lower case shifts, etc., are all characters and must be counted when filling memory with alpha-numeric information. Also remember to count the termination character.

E. PROVISION FOR RELOCATION

It may be desirable to write a program with relative addresses so that it may be relocated at will anywhere in interpretive memory. CINCH does this by permitting the programmer to specify the base address for loading the program (S-code) prior to reading it into memory. CINCH will then add the base address, set by the S code, to the address of commands in the program. Since there may be commands which will not be relative, an X-code is used to designate non-relative information. The relocation codes are explained in paragraph IV. A.

III. INDEX REGISTERS

The index registers perform a number of important functions. They can be used for "looping", that is, doing iterative operations, counting and tallying, and address modification. CINCH incorporates seven index registers. Each index register consists of a base, a modifier and a limit which can be set and reset separately. When executing an indexed command which is not manipulating the contents of an index register, the base address is added to the command address to give the effective address. The command is left unchanged in memory. For example, if the base of index register "F" was set to 329, then the command

CAD F 0026

would place the contents of location 0355 into the accumulator. The contents of the base of an index register must not be negative.

Sample problem illustrating the use of index register:

> Find the sum of all numbers stored in location 0400 to 0450 and store the answer in 0126. The program starts in location 0100.

Location	Program Tape	Symbolic	Remarks
	L0100		Set location.
0100	C02A0108	SIB A0108	Set base = 0000
1	03A0110	SIM A0110	Set modifier = 2
2	04A0112	SIL A0112	Set limit = 50
3	10 0400	CAD 0400	First number

Location	Program Tape	Symbolic	Remarks
4	12A0000	ADD A0402	Next number
5	06A0104	MIT A0104	Increment base of index by 2. Go back to add next number if 0452 has not been reached.
6	60 0126	STA 0126	Store answer when done
7	00 0000	HLT 0000	Stop (finished)
8	D+0	+0	Constants
10	+2	+2	
12	+50	+50	
	E		End of tape signal

IV. INPUT/OUTPUT

A. INPUT

There are two modes of operation, internal and external. After loading CINCH into memory (see Operating Instructions) the computer will be in the external mode. All external (manual) operations are described in Section V.

It is possible to enter information into the computer by way of the keyboard or paper tape. Reading is controlled by one of the external operations R, K, or S (see Section V) or by the RPT or RTK commands. The information read may consist of a combination of the following configurations:

	Code Letter
Alpha-numeric information	A
Commands	C
Decimal numbers	D
Location definitions	L
End of data signals	E
Transfers (Go To)	G
Non-relative address	X

Each configuration is specified by a code letter. When CINCH encounters an A, C, or D it assumes that all following information is of the given type, until another relevant code letter is read. A space in the code letter position, if present, is ignored. Tabs, in any position, are ignored. A defines alpha-numeric information, C specifies that the following are commands, D denotes data. Types L, E, X, and G are instructional and are not stored in memory. Code deletes are ignored on input.

When a code letter appears on a tape or is typed, the next character to be read must follow immediately unless separated by a tab. There may be no spaces, carriage returns, etc., between the code letter and the next character. Reading of information may be interrupted by one of the instructional code letters. These have the following effect:

Location Setting	Lnnnn	Set location counter and read the following information into memory starting with location nnnn. This code may not appear on a relocatable tape.
Go To	Gnnnn	Stop reading inputs and go to location nnnn.
End of Data	E	End of data signal. Stop reading inputs and transfer to location following RPT or RTK command, or back to the keyboard if under external control.
Non Relative Address	X	Do not add the contents of the relocation index (set with an S external command) to the address portion of this instruction (this applies only to relocatable programs).

An example of a standard program as it would appear on tape is as follows:

Note the L, C, and D, and E codes.

Example:

Find one root of the quadratic:

$$X = \frac{-B + \sqrt{B^2 - 4AC}}{2A}$$

where A, B, and C are stored in locations 0602, 0604 and 0606 respectively, and X will be put in location 0608. If $B^2 - 4AC < 0$ then $X = 0$. The program starts in location 0100.

Location	Program Tape		Symbolic	Remarks
	L0100			Set location
0100	C10 0118		CAD +4	C defines commands
0101	14 0602		MUL A	4•A
0102	14 0606		MUL C	4•A•C
0103	60 0610		STA Temp	Temporary Storage
0104	10 0604		CAD B	
0105	14 0604		MUL B	B^2
0106	22 0610		SUB Temp	$B^2 - 4AC$
0107	41 0114		TAN Neg	Out if $B^2 - 4AC < O$
0108	57 0000		SQR 0000	$\sqrt{B^2 - 4AC}$
0109	22 0604		SUB B	$-B + \sqrt{B^2 - 4AC}$
0110	15 0120		DIV +2	/2
0111	15 0602		DIV A	$(-B + \sqrt{B^2 - 4AC}) / 2A$
0112	60 0608		STA X	Ans. to X
0113	00 0000		HLT	Stop when done
0114	10 0122	Neg	CAD + 0	
0115	60 0608		STA X	X = 0
0116	00 0000		HLT	Stop
	L0118			Set location
0118	D4		+4	Constants
0120	2		+2	D defines data
0122	0		+0	
	E			End of tape, return control to keyboard.

B. OUTPUT

Output of data, commands, or alpha-numeric information may be on either the typewriter or punch. Command format was described in paragraph I. C, and alpha-numeric in paragraph I. D. Numeric data may be output in two forms: as four digit positive integers and as data.

The commands TXT and PXT will type or punch the contents of the base of a specified index register followed by a tab. The number must be positive and less than 4,095. Leading zeros are not suppressed.

Data is output by the commands TNT, TNC, PNT, and PNC and has the form:

±.nn---nn±ee

The fractional part of the number (n---n) will contain up to 10 digits preceded by an optional sign and decimal point. The exponent will consist of a sign and two digits.

Ordinarily, ten digits will be typed. However, the SFL command (Set Fraction Length) may be used to designate the number of digits. Note that the number is rounded only in the tenth place. Therefore, if less than ten digits are typed, truncation will occur.

A number is always followed by a carriage return or a tab code, whether printing or punching.

To punch a tape with the PNT or PNC commands that may later be read by the RPT command, the programmer must ensure that there is a "D" on the tape preceding the first piece of data punched. This can be accomplished either manually or by the PAC command.

Examples of input data tape:

```
L1000
D+123.456
  -1.234+02
   1.2345-03
   1234567-10
E
```

This tape stores data in the interpretive memory as follows:

Location	Data
1000	$+.1234560000 \times 10^{3}$
1002	$-.1234000000 \times 10^{3}$
1004	$+.1234500000 \times 10^{-2}$
1006	$+.1234567000 \times 10^{-3}$

If the contents of the preceding locations were printed by the TNC command, they would appear as follows:

```
 .1234560000+03
-.1234500000+03
 .1234500000-02
 .1234567000-03
```

There are no instructions to type or punch command format, but this is possible by external control. See the description of the M external operation in Section V for further details.

C. TRACING

If the BREAKPOINT switch is down, all commands with a trace tag will be printed. In addition, the location of the command and the base of the specified index register if any, will be printed as 4 digit numbers. The contents of the accumulator after execution of the command, will be printed in data format.

Example:

```
1037    T41C0167    0013    + .1234514239-14
```

V. EXTERNAL OPERATIONS

After CINCH has been loaded into memory the computer will be in external mode. The light on the Flexowriter will come on and the computer will be ready for one of the nine external operations.

It is possible to transfer to external mode at any time by depressing the ENABLE switch. The parity must then be cleared with the BREAKPOINT switch. Depressing the "I" key on the Flexowriter keyboard, and raising the ENABLE switch will effect the transfer of control. The Flexowriter light will come on and a series of characters may be typed to prepare CINCH for one of several functions. These are as follows:

Read Standard or Binary Tape	R	Read a tape punched in standard input or binary format, (containing commands, data, alpha-numeric information, etc.). The code letter E on a standard tape returns control to the keyboard for another external operation. The binary tape must have been previously punched by CINCH. In this case, control will return to the keyboard if the check sum compares. If not, the computer will halt with a line number of 03. Clearing parity will return control to the keyboard.
Punch Binary Tape	P	The operator must type an initial address, space, and a final address, followed by a carriage return. The information located in memory between these addresses will be punched out in 8-bit binary. Control is returned to the keyboard.

Read Standard Keyboard	K	Read the keyboard in standard input format. The operator may then type any information (as described in paragraph IV. A. The code letter E will return control to the keyboard for another external operation of the type described in this section.
Read Relocatable Standard Tape	S	The operator must type an address followed by a carriage return. The relocation base address will then be set to this address and a relocatable tape will be read. The code letter E on the tape returns control to the keyboard.
Go To	G	Control will transfer to the location currently in the program counter. This code is used to exit from external control or to resume full speed operation after single stepping.
Set Program Counter	$	The operator must type an address followed by a carriage return. This sets the program counter to the location given. The next command executed (for example under single step mode or following a "G") will come from this location.
Single Step	Z	Perform the next command. Initially, the program counter must be set by the $ external operation. Each time the Z key is depressed, the computer will execute one command and advance the program counter. Control remains with the keyboard.
Type Program Counter	Q	The computer will type the contents of the program counter. The address shown will be that of the last command performed, unless that command was a transfer, in which case it will be the location of the next command. When

single stepping, the program counter contains the address of the next command to be executed.

Memory Print Out	M	The operator must type an initial address, a space, and a C or D. The information in the location specified will be typed out in either Command (C) or Data (D) format. If the BREAKPOINT switch is down, only one location will be typed. If the BREAKPOINT switch is up, sequential locations will be typed out until the switch is depressed. If the address of 0000 is given, only the contents of the accumulator will be typed. Control is then returned to the keyboard for another external operation. Listable tapes (standard input format) may be produced by turning the punch on while executing a memory printout. The information will simultaneously output on the keyboard and punch. If the tape being punched is to be read back into CINCH, then the operator must also punch a C or a D on the tape immediately preceding the first command or piece of data, and a G or an E at the end of the tape.

VI. COMMAND LIST

A. INTRODUCTION

The following is the complete list of available commands. AC refers to the accumulator. Sw is a switch used by CINCH. ( ) refers to the contents of the named location enclosed in parentheses. The address portion of all commands with the exception of those referring to an index register are indexable. Where the index position, i, is specified, it is an integral part of the command and must be given. M refers to any word in memory, including the AC (address = 0000), and specifies the address used in performance of the command.

B. ARITHMETIC

CAD M — Clear and Add (10)

Clear the AC and add (M) to the (AC). (M) are unaffected.

CSU M — Clear Subtract (20)

Clear the AC and subtract (M) from the (AC). (M) are unaffected.

CAA M — Clear Add Absolute (11)

Clear the AC and add the absolute value of (M) to the (AC). (M) are unaffected.

ADD M — Add (12)

Add (M) to (AC). The result replaces (AC). (M) are unaffected.

ADA M — Add Absolute (13)

Add the absolute value (M) to (AC). The result replaces (AC). (M) are unaffected.

SUB M <u>Subtract</u> (22)

Subtract (M) from (AC). The result replaces (AC). (M) are unaffected.

MUP M <u>Multiply</u> (14)

Multiply (AC) by (M). Result replaces (AC). (M) are unaffected.

DIV M <u>Divide</u> (15)

Divide (AC) by (M). Result replaces (AC). (M) are unaffected. If (M) = 0, the computer halts, with a line number of 06.

DVM M <u>Divide Memory</u> (25)

Divide (M) by (AC). The result replaces (AC). (M) are unaffected. If (AC) = 0, the computer halts with a line number of 06.

C. INDEX REGISTERS

SIB i M <u>Set Index Base</u> (02)

Set the base of index i to (M). (M) are unaffected.

SIM i M <u>Set Index Modifier</u> (03)

Set the modifier of index i to (M). (M) are unaffected.

SIL i M <u>Set Index Limit</u> (04)

Set the limit of index i to (M). (M) are unaffected.

MIT i M <u>Modify Index and Transfer</u> (06)

The modifier of index i (m_i) plus the current value of the base of i (b_i) becomes the new value of b_i. If the new b_i has not passed the limit of i (L_i) then control is transferred to M. Otherwise the range is satisfied and control continues in sequence. This condition can be explained as follows:

$b_i = b_i + m_i$. If $m_i \geqslant 0$ and $b_i \leqslant L_i$ then transfer is to M.

If $m_i < 0$ and $b_i \geqslant L_i$ then transfer is to M.

Otherwise no transfer occurs and the program continues sequentially.

D. MANIPULATION AND TRANSFER OF INFORMATION

STA M **Store Accumulator** (60)

Store (AC) in M. The (AC) are unaffected. The original (M) are destroyed. M may not be 0000.

STB i M **Store Index Base** (05)

Store base of index i in M. The base is unaffected. The original (M) are destroyed. M may not be 0000.

E. DECISIONS AND TRANSFERS

TRU M **Transfer Unconditionally** (40)

Transfer control to M. The next command will be executed from location M. M may not be 0000.

TAN M **Transfer Accumulator Negative** (41)

Transfer control to M if $(AC) < 0$. Otherwise continue in sequence. M may not be 0000.

TAP M **Transfer Accumulator Positive** (42)

Transfer control to M if $(AC) \geqslant 0$. Otherwise continue in sequence. M may not be 0000.

TAZ M **Transfer Accumulator Zero** (43)

Transfer control to M if $(AC) = 0$. Otherwise continue in sequence. M may not be 0000.

TNZ M **Transfer on Non-Zero** (44)

Transfer control to M if $(AC) \neq 0$. Otherwise continue in sequence. M may not be 0000.

CAM M **Compare Accumulator and Memory** (30)

Set sw = (AC) - (M). (AC) and (M) are unaffected.

TCL M **Transfer if Comparison Switch Low** (31)

Transfer control to M if $sw < 0$. Otherwise continue in sequence. M may not be 0000.

TCH M <u>Transfer if Comparison Switch High</u> (32)

Transfer control to M if sw $>$ 0. Otherwise continue in sequence. M may not be 0000.

TCE M <u>Transfer if Comparison Switch Equal</u> (33)

Transfer control to M if sw = 0. Otherwise continue in sequence. M may not be 0000.

TCU M <u>Transfer if Comparison Switch Unequal</u> (34)

Transfer control to M if sw $\neq$ 0. Otherwise continue in sequence. M may not be 0000.

TSI i M <u>Transfer and Set Index</u> (07)

The base of index i is set with the location of the TSI command and control is transferred to M. M may not be 0000.

EXT M <u>Exit</u> (45)

Exit from control of CINCH. This command is used to transfer to a PB 250 machine language program. M is given as a standard CINCH address and the operator must know the actual location this represents. See paragraph VII. C for a complete description of the use of this command.

F. CONTROL

HLT <u>Halt</u> (00)

Computation stops with a line number 04 displayed and the computer control panel PARITY light comes on. To continue execution of the program the ENABLE and BREAKPOINT switches must be depressed. Upon raising the ENABLE switch, the program will continue.

NOP <u>No Operation</u> (01)

No operation is executed. Control continues in sequence.

G. INPUT/OUTPUT

RPT M **Read Paper Tape** (17)

Read paper tape prepared in standard input format, into memory, starting with location M. Cease reading upon signal from tape. (See paragraph IV. A.)

RTK M **Read Typewriter Keyboard** (16)

Read keyboard in standard input format into locations starting with M until signal from keyboard. (See paragraph IV. A.)

TNT M **Type Number and Tab** (26)

Type data word from location M followed by a tab. (See paragraph IV. B.)

TNC M **Type Number and Carriage Return** (27)

Type data word from M followed by a carriage return. (See paragraph IV. B.)

PNT M **Punch Number and Tab** (36)

Punch data word from M followed by a tab. (See paragraph IV. B.)

PNC M **Punch Number and Carriage Return** (37)

Punch data word from M followed by carriage return. (See paragraph IV. B.)

TAC M **Type Alpha-Numeric** (46)

Type alpha-numeric characters starting with location M until ; or : is encountered. (See paragraph II.D.) M may not be 0000.

PAC M **Punch Alpha-Numeric** (47)

Punch alpha-numeric characters starting with location M until ; or : is encountered. (See paragraph II.D.) M may not be 0000.

SFL M **Set Fraction Length** (35)

Initially, the TNT, TNC, PNT, PNC, commands, the external operation M, and the Trace assume that data is to be typed or punched with a fractional part of 10 digits. This value is permanently changed by M, $(1 \leqslant M \leqslant 10)$ when this command is executed. The fraction is rounded as if M = 10. Therefore, if $M < 10$, the fraction is truncated.

TXT i **Type Index Register and Tab** (63)

Type the contents of the base of the specified index register as 4 digits followed by a tab. The largest number permitted is 4095.

PXT i **Punch Index Register and Tab** (64)

Punch the contents of the base of the specified index register as 4 digits followed by a tab. The largest number permitted is 4095.

H. FUNCTIONS

SIN M **Sine** (50)

Sine (M) replaces (AC). (M) are unaffected. (M) must be in radians.

COS M **Cosine** (51)

Cosine (M) replaces (AC). (M) are unaffected. (M) must be in radians.

ATN M **Arctan** (52)

Arctan (M) replaces (AC). (M) are unaffected. (AC) will be in radians.

LNE M **Log_e** (53)

$\text{Log}_e |(M)|$ replaces (AC). (M) are unaffected.

LOG M <u>Log_{10}</u> (54)

$\mathrm{Log}_{10}\ |(M)|$ replaces (AC). (M) are unaffected.

EXP M <u>Exponential e^{M}</u> (55)

$e^{(M)}$ replaces (AC). (M) are unaffected.

TEN M <u>Exponential 10^{M}</u> (57)

$10^{(M)}$ replaces (AC). (M) are unaffected.

SQR <u>Square Root</u> (56)

$\sqrt{|(M)|}$ replaces (AC). (M) are unaffected.

VII. SUBROUTINES

A. INTRODUCTION

A certain class of programs are called "subroutines". These programs are self-contained units which may be entered repeatedly from various parts of the main program. The return to the main program is usually to the location following the entry command. Two types of subroutines are acceptable to CINCH. These are: CINCH language programs and machine language programs. The more desirable form is CINCH language, which may be used for most subroutines written by the programmer. Machine language should be reserved only for those subroutines whose speed is a critical factor and where the use of CINCH language would be awkward.

B. CINCH LANGUAGE SUBROUTINE

As an example of a subroutine, the programmer may frequently need to compute the tangent of an angle in the program, and this function is not part of CINCH. The programmer may code a routine to evaluate it in CINCH language and store the routine anywhere in memory. By using the TSI command, the operator can enter it from many different places in the particular program.

This command stores its location in the base of the index register specified and then transfers to the address given. In this way, the subroutine "remembers" where it came from and can control its return to the main program at the appropriate location.

For example, assume there is a subroutine to evaluate the tangent of an angle. This subroutine is located in 0124-0175 and expects the argument to be in the accumulator at entry. After evaluation, control returns to the main program with the answer in the accumulator. The last command executed in the subroutine is TRU F 0001. Therefore, to enter this subroutine, the program must have loaded the argument into the accumulator and then given TSI F 0124, which loaded the base of index F with the address of the TSI command and transferred to the tangent subroutine. After evaluation of a tangent the

TRU F 0001

will return control to the command immediately following the TSI command.

Example:

$$X = \frac{a \cdot TAN\ (b+c) - d}{TAN\ e}$$

Location		
1000	CAD b	
1001	ADD c	b + c
1002	TSI F 0124	TAN (b + c)
1003	MUL a	a•TAN (b + c)
1004	SUB d	-d
1005	STA Temp	⟶ Temp.
1006	CAD e	
1007	TSI F 0124	TAN e
1008	DVM Temp	Temp/ TAN e
1009	STA X	= X

It is frequently desirable to have a subroutine relocatable, that is, coded with only relative addresses so that it may be located anywhere in memory. Of the two types of subprograms acceptable to CINCH, those coded in CINCH language and those written in machine language, only CINCH language programs may be relocated. CINCH language programs look like standard programs except that the commands with non-relative addresses must be flagged with an X code.

To relocate a given program tape with relative addresses, the operator gives an S external command. This sets the base address for reading the tape. All commands on the tape that do not have an X code as the first character will have the base address added to the address portion of the command as it goes into memory. The X may only be used to protect commands. Data is automatically protected. Where it is necessary for a C and an X to precede the same command, the C must come first.

C. MACHINE LANGUAGE SUBPROGRAMS

It is possible to write machine language subroutines and link to them from CINCH. However, this is a complicated process and the programmer is cautioned to take extreme care in its performance.

The linkage from CINCH to machine language is with the EXT command,

EXT M

where M is a standard, 4 digit, CINCH address. Conversion to the machine language equivalent of this address is accomplished as follows:

Example: 0139

1. Express address as 12 binary bits: 000010001011
2. 8 low order bits are sector number: 10001011
3. 4 high order bits plus 1100 are line number.

Thus 0139 becomes line 14 sector 213.

The accumulator occupies three words in the short line and has the following structure:

← 22 →		← 22 →	← 15 →	← 7 →
±	High order	Low order	±	Exponent

The programmer may return to CINCH at anytime, however certain sectors in memory must be restored if line 00 has been disturbed. These sectors are as follows:

001 } 002 }	Accumulator
004	Exponent of accumulator
006	Location counter
015	+0000000

Return to the command following the EXT command by transferring to 345 03.

The programmers machine language segment must be in a command line. If it is necessary to move any command line occupied by CINCH, then the user must restore this line before returning to CINCH.

CAUTION

To avoid possible errors in programming it is essential to exercise extreme care when writing in machine language.

APPENDIX A

SAMPLE PROBLEMS

Sample Program No. 1 Calculate: $P = \frac{0.674SIN(y)}{\sqrt{1 - X^2}}$

where $y = e^B - A$ for an unknown number of data sets.

A, B, and X are read from paper tape, the end of data being signalled by G0016.

Location	Program Tape		Symbolic	Remarks
	L0001			Set Location
0001	C17 0018	Z1	RPT	Read A, B, X into 0018, 0020, 0022.
0002	55 0020		EXP B	e^B
0003	22 0018		SUB A	$e^B - A = y$
0004	50 0000		SIN 0000	SIN (y)
0005	14 0024		MUP	0.674 SIN (y)
0006	60 0030		STA Temp	
0007	10 0022		CAD X	
0008	14 0022		MUP X	X^2
0009	60 0032		STA Temp 2	
0010	10 0026		CAD + 1	
0011	22 0032		SUB Temp 2	$1 - X^2$
0012	56 0000		SQR 0000	$\sqrt{1 - X^2}$
0013	25 0030		DVM Temp	$(0.674 \text{ SIN } y) / \sqrt{1-X^2}$
0014	60 0040		STA P	
0015	40 0001		TRU Z1	
0016	00 0000	Z2	HLT	Stop
	L0024			Set Location
0024	DO.674		0.674	Constants
0026	1.0		1.0	
	E			End of Program

Data Tape

D (A--)
(B--)
(X--)
E
(A--)
(B--)
(X--)
E
--
--
--
--
(A--)
(B--)
(X--)
G0016

Sample Program No. 2 Table-Look-Up

Find:

$$y = f(x)$$

where x has been previously calculated and stored in location 0100. Let A and B represent two tables of 25 elements each where

$$A_1 \leqslant X \leqslant A_{25}$$

If the current value of X is

$$A_i \leqslant X < A_{i+1}$$

then

$$y = B_i$$

Table A is stored in locations 0050-0099 and table B is stored in 0150-0199. When completed, store y in 0102 and transfer to location 0135 for the rest of the program.

Location	Program Tape		Symbolic	Remarks
	L0001			Set location
0001	C02F0012		SIB F + 0	Set base of index F = 0
0002	03F0014		SIM F + 2	Set modifier of index F = 2
0003	04F0016		SIL F + 48	Set limit of index F = 48
0004	10 0100		CAD X	X
0005	30F0050	Z1	CAM F A_i	X: A_i
0006	31 0009		TCL Z2	Transfer to Z2 if $X < A_i$
0007	06F0005		MIT F Z1	Step index and compare A_{i+1}
0008	00 0000		HLT	Error Halt
0009	10F0148	Z2	CAD F B_{i-1}	B_{i-1}
0010	60 0102		STA y	$Y = B_{i-1}$
0011	40 0135		TRU 0135	Out
0012	D + 0			
0014	+ 2			
0016	+ 48			
	E			End of tape

Sample Program No. 3

Use of Relocatable Subroutine

Calculate: $$R = \frac{2050\, D^2 L}{Q^5 - 1}$$

where $$D = \frac{2a}{\sqrt{2-a^2}}$$

$$Q = \frac{B}{\log_{10} |k|}$$

and $$k = \left[\frac{b}{\sqrt{2 - b^2}} + b\right]$$

$$L = \frac{a + b}{\sqrt{2 - (a+b)^2}}$$

A and B are read from tape. The function

$$\frac{X}{\sqrt{2-X^2}}$$

will be evaluated by a relocatable subroutine which expects the argument to be in the accumulator, and leaves the answer also in the accumulator. Index register 'G' is used by the TSI command which links to the subroutine. The answers will be typed out as drawn including headings.

RUN	A	B	R
0001	.1000000000+01	-.1000000000+01	.0000000000+00

The subroutine appears as follows:

Relative Location	Program Tape	Symbolic	Remarks
0000	C60 0010	STA X	Argument Stored in X
0001	X14 0000	MUP 0000	X^2
0002	60 0012	STA Temp	
0003	10 0008	CAD + 2	
0004	22 0012	SUB Temp	$2 - X^2$
0005	X56 0000	SQR 0000	$\sqrt{2 - X^2}$
0006	25 0010	DVM X	$X/ \sqrt{2 - X^2}$
0007	X40G0001	TRU G 0001	Return
0008	D + 2	+2	Constant
0010	0	X	
0012	0	X^2	Temp
	E		

The main program reads a and b into location 0002 and 0004 respectively and expects the subroutine to be locations 0050 - 0063.

The data tape is similar to that in problem number one, with the termination signalled by G0135.

Location	Program Tape		Symbolic	Remarks
	L0100			Set location
0100	46 0160		TAC 0160	Type headings
0101	02C0138		SIB C + 1	Run = 1
0102	03C0138		SIM C + 1	
0103	04C0140		SIL C + 2050	Any large number
0104	17 0002	Start	RPT 0002	
0105	10 0004		CAD b	
0106	07 G0050		TSI G 0050	Link to subroutine

Location	Program Tape	Symbolic	Remarks
0107	12 0004	ADD b	$b/\ \ 2-b^2+ b$
0108	60 0146	STA B	= B
0109	54 0146	LOG B	$\log_{10} \lvert B \rvert$
0110	25 0146	DVM B	$B/\log_{10} \lvert B \rvert$
0111	60 0148	STA Q	= Q
0112	10 0002	CAD a	
0113	07G0050	TSI G0050	$a/\sqrt{2-a^2}$
0114	14 0142	MUL + 2	
0115	60 0150	STA D	
0116	10 0002	CAD a	
0117	12 0004	ADD b	a + b
0118	07G0050	TSI G 0050	$(a+b)/\sqrt{2-(a-b)^2} = L$
0119	14 0140	MUL + 2050	2050 L
0120	14 0150	MUL D	2050 LD
0121	14 0150	MUL D	$2050\ LD^2$
0122	60 0152	STA temp	
0123	10 0148	CAD Q	
0124	14 0148	MUL Q	Q^2
0125	14 0000	MUL 00000	Q^4
0126	14 0148	MUL Q	Q^5
0127	22 0138	SUB + 1	$Q^5 - 1$
0128	25 0152	DVM temp	$2050\ D^2L/Q^5 - 1$
0129	60 0006	STA R	= R
0130	63C0000	TXT C	Type run number
0131	26 0002	TNT	Type A
0132	26 0004	TNT B	Type B
0133	27 0006	TNC R	Type R

Location	Program Tape		Symbolic	Remarks
0134	06C0104		MITC start	Run=Run + 1 go to
0135	00 0000	end	HLT	Start
	L0138			Set location
0138	D + 1			Constants
0140	2050			Constants
0142	2			
	L0160			Set location
0160	ARUN			Carriage control func-
0161	(tab A (tab)			tions are (tab) and (c/r).
0162	B (tab) R			This message contains
0163	(c/r);			11 characters and is
				stored in 0160-0163.
	E			End of tape.

To run the previous problem, the following steps must be performed:

1. Load CINCH I (see Operating Instructions).
2. Set program tape in reader.
3. Type R (under external mode), the program will then load.
4. When the Flexowriter light comes on, set subroutine tape in reader.
5. Type S0050 c/r, the subroutine tape will be loaded into locations 0050-0063.
6. When the Flexowriter light comes on, set data tape in reader.
7. Type $0100G, control will transfer to the start of the program. When completed, the computer will stop and a line number of 04 will be displayed.