# POLICY STATEMENT

Users of the computer are invited to make use of the various programming aids and numerical analysis references to be found in the lab.

Suggestions for improvement of laboratory facilities and references will be appreciated by the staff of the laboratory. Comments may be directed to any member of the staff.

Users of the laboratory are to leave the lab as clean as they found it.

Users of the laboratory may not enter the maintenance area.

Violation of the rules of the Laboratory may result in

(a) explusion from the laboratory,
(b) termination of computing privileges,
(c) disciplinary action by the Department, and
(d) disciplinary action by the University.

GOOD COMPUTING

THE UNIVERSITY OF WISCONSIN

DEPT. OF ELECTRICAL ENGINEERING

# MEMORANDUM

From C. H. Davidson

September 10, 19 60

To All EE Staff

Subject Availability of computing facilities

The attached material will be taught in EE 182 the first week of classes this fall, and will be used as part of the experiment run the second week in EE 183. It will be taught also in EE 34 the second week, and used in EE 35 the third week. Subsequently, in both EE 182 and EE 34, homework assignments will be made frequently that require the students to use the digital computer to study and solve problems related to the course content. In addition, EE 42, assuming the majority of students are concurrently registered in either EE 34 or EE 182, will make similar use of the computer in connection with course requirements.

In the light of this new facility on the part of our upper class undergraduates, it is hoped that still other instructors will take advantage of the opportunity to use a general purpose digital computer as a tool in engineering education, analysis, and research. It is realized that for some staff members this will involve first becoming familiar with the computer and its use. To this end, the staff of the computing laboratory promises every assistance in answering questions, writing programs, and running the machine.

Specifically, you are cordially invited to attend either EE 182 lecture, 8:50 or 1:20, on Friday, September 16; or the EE 34 lecture at 8:50 on Thursday, September 22. Material on digital computer programming will be presented, and PAT coding will be discussed.

If any of us can help you in planning for increased computer usefulness, please feel free to ask.

from C. H. Davidson —— to ——> C. W. McClure
D. L. Dietmeyer
A. K. Scidmore
J. F. McNall

D. R. Smith
C. C. Bjerke
B. L. Weinberg

# PSEUDO

# ASSEMBLER

# TRANSLATOR

**PAT**

Prepared by the Staff

of the

Computing Laboratory

of the

Department of Electrical Engineering

University of Wisconsin

Madison, Wisconsin

CHD
9/6/60

## Introduction

PAT is a pseudo assembler translator available to users of the WISC, whereby the computer can be programmed as a three address, floating point, decimal machine with a large class of special functions available as special order types.

Each PAT order contains an order type which identifies the order, and one, two, or three addresses. These addresses refer to three hundred memory positions numbered from 000 to 299 in which instructions or numerical data may be stored. A program is executed sequentially except when transfer instructions are encountered.

## Classes of Orders

There are six classes of orders available to PAT users:

1. Loading
2. Arithmetic
3. Control
4. Comparison
5. Special Function
6. Counting

In the following discussion the symbol -X- refers to the address of X, and (X) refers to the contents of memory position X.

Note that all words prepared for or received from the computer must consist of thirteen characters (excluding spaces or periods) and a comma.

1. The LOAD instruction is a class to itself:

LOAD 000 0 -W- -V- -N-,

There may be only one LOAD instruction in a program, and it must be the first instruction given to the computer. LOAD instructs the computer to store the program that follows into memory positions -W- to -V-. At the end of this program the word "end," must be inserted; when this is encountered the computer stops. When the computer is restarted the first instruction to be executed will be the order stored in -N-.

2. The second class of orders available to PAT users consists of arithmetic instructions. As typified by the ADD instruction:

ADD 000 8 -A- -B- -R-,

each of these six order types specifies an arithmetic operation T, the address of two operands A and B, and an address to which the result R is to be sent.

3. Control instructions specify the address of the next instruction, and thus interrupt the normal sequential operation of the program. The TRA instruction transfers control without stopping; HTR first halts the computer, and transfers only when computation is restarted.

4. Comparison instructions combine some of the features of classes two and three. They call for a subtraction of two numbers, and a transfer of control conditional upon the sign of this result.

5. This class of orders consists of a large number of special function instructions. However, three of these instructions deserve special attention here.

The IN order instructs the computer to load numerical data into a specified band of sequential locations; the OUT order instructs the computer to punch numerical data from a similar band. The format of the numbers necessary for proper operation of these orders is described under Numbers.

The CHK instruction tells the computer to check itself for error-free operation. If any fault is found the computer will stop; if the computer is restarted it will again check itself. If no error is found the computer will return control to the program under progress.

The remaining special functions are all of the type which operate upon a single operand to produce a single result; e.g.,

SQR 010 4 -A- 000 -R-,

delivers to -R- the square root of (A). A complete list of the special functions presently available is found on the order summary sheet; other special functions may be added from time to time. The accuracy of all special functions is at least 0.001% and may be much better in some cases. Further information about special functions may be obtained from members of the staff of the Computing Laboratory.

6. Counting instructions are available only in PATRICK; see "Advanced Problems," page 3.

## Short Memory

Whenever an order requires the result of the immediately preceding order, a special coding must be used. To obtain the immediately preceding result for either the A or B operand of an order (or both), use the address 800, which is known as "short memory." Note, however, that neither the operand nor the result of a special function may be obtained this way; in a special function instruction, and in the order immediately following it, use the actual addresses of the operands.

Advanced Problems

For the student who is interested in solving more advanced problems, such as those involving differential equations, matrices, or statistical procedures, a more powerful version of PAT (known as PATRICK) is available. Beyond this, the basic machine language of the WISC is the most flexible tool for handling more involved programs. While for simple problems most PAT programs will operate almost as fast as programs written in WISC language, as the complexity increases so does the speed advantage as well as the power of WISC.

Details of existing variants of PAT and information on WISC programming may be obtained from members of the staff of the Computing Laboratory.

Numbers

To conform to standard scientific notation, all numbers for PAT consist of a ten digit number with the decimal point assumed to follow the first digit, and a two digit exponent of the base ten. Thus, numbers are represented in the form

$$P \times 10^{Q}$$

where P must be in the range between ten and one in magnitude, and Q must be less than 76 in magnitude.

Both the significant digits and the exponent carry signs, and the first character of a number identifies the signs with the following convention:

| | |
|---|---|
| 0 | positive number; positive exponent |
| 1 | positive number; negative exponent |
| 2 | negative number; positive exponent |
| 3 | negative number; negative exponent. |

The reader familiar with binary numbers should be able to derive this code from the two sign bits involved.

The following examples show the manner in which numbers are prepared for input to the computer and the form in which numbers are delivered as output from the computer. A number is assembled by giving the sign character described above, the two digits of the exponent, and the ten digits of the significant figure.

| Number | Input | Output |
|---|---|---|
| 0 | 000 0.000 000 000, | 0000000000000, |
| 2 | 000 2.000 000 000, | 0002000000000, |
| -75 | 201 7.500 000 000, | 2017500000000, |
| .045 | 102 4.500 000 000, | 1024500000000, |
| -.00467 | 303 4.670 000 000, | 3034670000000, |

# PAT ORDER SUMMARY

## Loading

| | | |
|---|---|---|
| LOAD | 000 0 -W- -V- -N-, | Load program into positions -W- to -V-; start at -N-. |

(program to be loaded must be followed by:)
end,

## Arithmetic

| | | |
|---|---|---|
| ADD | 000 8 -A- -B- -R-, | (R) = (A) + (B) |
| ADA | 000 9 -A- -B- -R-, | (R) = \|(A)\| + \|(B)\| |
| SUB | 000 a -A- -B- -R-, | (R) = (A) - (B) |
| SUA | 000 b -A- -B- -R-, | (R) = \|(A)\| - \|(B)\| |
| MPY | 000 2 -A- -B- -R-, | (R) = (A) x (B) |
| DIV | 000 3 -A- -B- -R-, | (R) = (A) / (B) |

## Control

| | | |
|---|---|---|
| TRA | 000 5 000 000 -N-, | Next instruction from -N- |
| Halt HTR | 000 6 000 000 -N-, | Halt; next instruction from -N- |

## Comparison

| | | |
|---|---|---|
| TNE | 000 e -A- -B- -N-, | Next instruction from -N- if (A) - (B) < 0 |
| TNA | 000 f -A- -B- -N-, | Next instruction from -N- if \|(A)\| - \|(B)\| < 0 |

## Special Function

| | | |
|---|---|---|
| CHK | 000 4 000 000 000, | Check computer; stop if not working properly |
| IN | 001 4 -F- 000 -L-, | Load numerical data into positions -F- to -L- |
| Punch OUT | 002 4 -F- 000 -L-, | Punch numerical data from positions -F- to -L- |
| SQR | 010 4 -A- 000 -R-, | (R) = square root of (A) |
| SIN R | 005 4 -A- 000 -R-, | (R) = sin (A); (A) in radians |
| SIN D | 015 4 -A- 000 -R-, | (R) = sin (A); (A) in degrees |
| COS R | 006 4 -A- 000 -R-, | (R) = cos (A); (A) in radians |
| COS D | 016 4 -A- 000 -R-, | (R) = cos (A); (A) in degrees |
| LOG E | 007 4 -A- 000 -R-, | (R) = $\log_e$ (A) |
| LOG T | 017 4 -A- 000 -R-, | (R) = $\log_{10}$ (A) |
| EXP E | 008 4 -A- 000 -R-, | (R) = $e^{(A)}$ |
| EXP T | 018 4 -A- 000 -R-, | (R) = $10^{(A)}$ |
| ARC R | 009 4 -A- 000 -R-, | (R) = arctan (A); (R) in radians |
| ARC D | 019 4 -A- 000 -R-, | (R) = arctan (A); (R) in degrees |

EXAMPLE

PAT CODING FOR $Y = \frac{1}{2} \sin^2 \theta$ $(\theta = 30^\circ, 45^\circ, -15^\circ)$

BY CWMc DATE 4-27-60 PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | N | X | T | A | B | R |
|---|---|---|---|---|---|---|---|---|
| | | | | DECIMAL | | | | |
| | LOAD | Load Program | | -- | 0 | 038 | 045 | 038 |
| | IN | Bring in constant | 038 | 001 | 4 | 101 | -- | 101 |
| | CHK | Check the computer | 039 | -- | 4 | -- | -- | -- |
| | IN | Bring in θ | 040 | 001 | 4 | 197 | -- | 197 |
| | SIND | Form sin θ | 041 | 015 | 4 | 197 | -- | 198 |
| | MPY | Square sin θ | 042 | -- | 2 | 198 | 198 | -- |
| | DIV | Form 1/2 $\sin^2 \theta$ | 043 | -- | 3 | 800 | 101 | 199 |
| | OUT | Punch θ, sin θ, 1/2 $\sin^2 \theta$ | 044 | 002 | 4 | 197 | -- | 199 |
| | HTR | Stop; ready to repeat | 045 | -- | 6 | -- | -- | 039 |
| | end | | | end, | | | | |
| | | Constant | 101 | -- | 2 | -- | -- | -- |
| | | $\theta_1$ | 197 | 001 | 3 | -- | -- | -- |
| | | $\theta_2$ | 197 | 001 | 4 | 500 | -- | -- |
| | | $\theta_3$ | 197 | 201 | 1 | 500 | -- | -- |
| | | | | | | | | |
| | | | | 000 | 0 | 038 | 045 | 038, |
| | | | | 001 | 4 | 101 | 000 | 101, |
| | | | | 000 | 4 | 000 | 000 | 000, |
| | | | | 001 | 4 | 197 | 000 | 197, |
| | | Listing of problem tape ready | | 015 | 4 | 197 | 000 | 198, |
| | | to be fed into computer. | | 000 | 2 | 198 | 198 | 000, |
| | | | | 000 | 3 | 800 | 101 | 199, |
| | | (Note use of comma at end of | | 002 | 4 | 197 | 000 | 199, |
| | | each word.) | | 000 | 6 | 000 | 000 | 039, |
| | | | | end, | | | | |
| | | | | 000 | 2 | 000 | 000 | 000, |
| | | | | 001 | 3 | 000 | 000 | 000, |
| | | Results produced by computer | | 001 | 4 | 500 | 000 | 000, |
| | | | | 201 | 1 | 500 | 000 | 000, |
| | | | | | | | | |
| | | 0013000000000, 1014999999918, 1010249999959, | | | | | | |
| | | 0014500000000, 1017071067829, 1012500000012, | | | | | | |
| | | 2011500000000, 3012588190450, 1023349364905, | | | | | | |

Certain additional features are available within PAT which make it possible to write more involved programs to solve more difficult problems. Foremost among the additional features are (a) counting instructions and (b) the use of subroutines for solving many standard problems such as simultaneous linear equations, polynomial equations, and ordinary differential equations.

## Counting Instructions

The five counting instructions cause instructions within a program to be modified during the running of the program; in so doing a very powerful program may be obtained.

### DOK

A DOK instruction is executed as a TRA order for a specified number of times and is then bypassed once. It is written as:

DOK:    i:   000 1 --K 000 -N-,

where K must be less than 100.

This instruction is treated as a "TRA to -N-" the first K-1 times it is executed; the $K^{th}$ time this instruction is executed it is not treated as a TRA order, but is bypassed so that control passes to the order in location i+1. At this time the counter associated with this instruction is reset to zero so the instruction may be used again. (A DOK instruction may be reset at any time by a RST instruction, as described below.)

If, as is common, -N- is less than -i-, this instruction has the effect of causing the block of orders between -N- and -i- to be executed K times: hence "DO K times."

For example, the following instruction

056:  000 1 011 000 049,

would cause the block of orders from 049 to 055 to be executed eleven times, and then the order in location 057 would be executed.

MOVE

The MOVE instructions move one word (either a number or an instruction) from one location to another; the MOVE instructions may then be automatically modified. They are written as:

i: 0 00x 1 -A- spp -R-,

where x indicates the addresses which are to be incremented following each execution, with the convention

| | | |
|---|---|---|
| MVA: | x = 1 | Increment -A- only |
| MVR: | x = 2 | Increment -R- only |
| MVAR: | x = 3 | Increment both -A- and -R-; |

p is the two digit increment to be added to the specified addresses;

and s is the sign of the increment: positive increment denoted by s = 0, and negative increment by s = 1.

Each time a MOVE instruction is executed the contents of the current -A- is copied into the current -R-, the A and/or the R addresses of the MOVE instruction are incremented by the appropriate amount, and a record is made of the number of times the MOVE instruction has been executed. (The original values of the A and R addresses may be restored at any time by a RST instruction, as described below.)

For example, the following pair of instructions

031: 001 1 109 001 299,

032: 002 1 299 001 250,

would cause the contents of 109 to be copied into 299 and then into 250 the first time they were executed. The next time they were executed (assuming that they have not been reset in the interim) the contents of 110 would be copied into 299 and then into 251. The next time they were executed the contents of 111 would be copied into 299 and then into 252. Note that the same result could have been obtained by the order pair:

031: 001 1 109 001 299

032: 003 1 109 001 250

It should be noted that neither the operand nor the result of a MOVE instruction may be obtained via short memory. Use the actual addresses of the operands in the MOVE instruction and in the order immediately following.

An error punching will occur and the computer will stop with the PAT idisplay showing if

(a) a MOVE instruction has been executed more than 100 times without being reset, or if

(b) the updated A or R addresses fall outside the limits of PAT.

## RESET

A RST instruction restores a DOK or a MOVE instruction to its original state. It is written as:

RST:    i:   004 1 -Q- 000 000,

where -Q- is the address of the counting instruction to be reset.

The only action when a DOK instruction is reset to that the counter associated with that instruction is set equal to zero.

The action when a MOVE instruction is reset consists not only of setting the counter associated with that instruction equal to zero, but also of restoring the original values of the A and the R addresses.

All MOVE and DOK instructions are reset at the time of loading by PAT.

## Additional Subroutines

Many additional subroutines are available to PAT users in the WISC library. Three of the most useful of these subroutines will be described in detail here; information on others may be obtained from members of the staff of the Computing Laboratory. Because all subroutines were written to be used without PAT, it is necessary to digress to explain the procedure for loading a group of words directly into the computer memory.

### READ PAPER TAPE

A RPT instruction is used in much the same manner as the LOAD instruction which starts every PAT program; however, it may be executed at any time during a program. It is written as:

RPT: i: 000 0 -W- -V- -N-,

where -W- is the address of the first word to be loaded

-V- is the address of the last word to be loaded

and -N- is the address of the next order to be executed.

A RPT instruction causes the computer to load additional data and/or instructions directly into the memory of the computer. Since these additional instructions do not pass through PAT for translation, the user must supply the correct form for the instructions. In particular, it will be necessary to convert all decimal addresses to their binary equivalents (PAT does this automatically for all instructions loaded through PAT). However, since 12-bit binary numbers are awkward for human beings to handle, it is customary to group these bits in groups of four, and represent each group as a single hexadecimal or base 16 character ($2^4 = 16$). These characters consist of the ten decimal digits 0-9 and the first six letters a-f. Here and elsewhere, a subscript on a number indicates the radix or base to which the digits of that number refer: e.g., $0001{:}0010_2 = 12_{16} = 18_{10}$. The attached conversion table gives the decimal-hexadecimal equivalents that will be required for such conversions.

### POLYNOMIAL ROOT SOLVER

PRS is a very powerful subroutine that will quickly evaluate the real and complex roots of a polynomial with real coefficients.

The polynomial must be in a standard form for use with PRS. The coefficients must be listed in order of descending powers of the unknown with the leading coefficient (coefficient of the highest power of the unknown) equal to unity. All the coefficients except the leading coefficient (which is unity) must be stored in a band of n sequential locations (where n is the degree of the polynomial). The band of locations used for the coefficients must be selected with care because PRS not only destroys the coefficients in the process of extracting the roots, but also uses the 2n+4 locations immediately following the coefficients.

Because of the form of information required by PRS, it is necessary to use a RPT instruction to load the three words that comprise the link between the PAT program and PRS. The link is:

```
        L:    100 8 L+2 3ff 35f,
      L+1:    000 5 000 000 001,
      L+2:    110 0 --n -C- -N-,
```
*

where L+2 is the address of the third link word

--n is the degree of the polynomial

-C- is the address of the first coefficient stored

and -N- is the address of the order to be executed after all the roots of the polynomial have been found.

All the addresses in the link must be given in the hexadecimal equivalents for the decimal addresses used.

The roots of the polynomial will be punched out in pairs as soon as they are determined without any further programming. Two numbers will always be punched out (except for the case of the final root of a polynomial of odd degree) and two possibilities must be considered.

(a) If the second number punched out is positive, both roots are real and can be found using the relationships

$$r_1 = x + y$$

$$r_2 = x - y$$

where x is the first number punched out and y is the second number punched out.

(b) If the second number punched out is negative, the two roots are complex conjugates and can be found using the relationships

$$r_1 = x + j\,y$$

$$r_2 = x - j\,y$$

where x is the first number punched out and y is the second number punched out.

PRS uses the entire band of locations from 001 through 174 in addition to the aforementioned band of locations following the coefficients. Therefore, PAT programs using PRS must not place any instructions or data in memory locations below 175.

Special loading procedures (Breakpoint Switches, etc.) must be used to allow PAT to process a program with an RPT instruction and to insure that PRS is available. The person in charge of the computer will be able to assist the user in these matters. It is further suggested that a member of the computing staff be consulted whenever a polynomial of degree higher than nine is being programmed.

- - - - - - - - - - -

* A tape is available which may be fed through the Flexowriter to produce the fixed portion of the link. This tape will stop at each place that the user must supply data which is unique to his program.

PAT CODING FOR PRS Example

BY Charles W. McClure DATE ______ PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | N | X (Decimal) | T | A | B | R |
|---|---|---|---|---|---|---|---|---|
| | | (PRS must be previously loaded) | | | | | | |
| | LOAD | Load PAT Program; | | —— | 0 | 251 | 258 | 251 |
| | | Set PC = 2 | | | | | | |
| | RPT | Bring in link to PRS | 251 | —— | 0 | 259 | 261 | 252 |
| | IN | Bring in $a_4$, $a_3$, $a_2$, $a_1$, $a_0$ | 252 | 001 | 4 | 280 | —— | 284 |
| | RST | Reset MVA | 253 | 004 | 1 | 255 | —— | —— |
| | RST | Reset MVR | 254 | 004 | 1 | 257 | —— | —— |
| | | | | | | | | |
| | MVA | Move $a_i$ to OPSTO | 255 | 001 | 1 | 281 | 001 | 290 |
| | DIV | $A_i = a_i/a_4$ | 256 | —— | 3 | 290 | 280 | 290 |
| | MVR | Move $A_i$ from OPSTO | 257 | 002 | 1 | 290 | 001 | 281 |
| | DOK | Do for the 4 coefficients | 258 | 000 | 1 | 004 | —— | 255 |
| | end, | | | end, | | | | |
| | | | | | | | | |
| | | Link to PRS | 259 | 100 | 8 | 105* | 3ff | 35f |
| | | gives | 260 | —— | 5 | —— | —— | 001 |
| | | [illegible] | 261 | 110 | 0 | 004 | 119* | 0fc* |
| | | | | | | | | |
| | | Polynomial Coefficients $a_4$ | 280 | | | | | |
| | | $a_3$ | 281 | | | | | |
| | | $a_2$ | 282 | | | | | |
| | | $a_1$ | 283 | | | | | |
| | | $a_0$ | 284 | | | | | |
| | | | | | | | | |
| | | OPSTO | 290 | | | | | |

* $119_{16} = 281_{10}$

$0fc_{16} = 252_{10}$

$105_{16} = 261_{10}$

RUNGE-KUTTA STEP

If the solution of a differential equation is desired when the values of the solution and its derivatives are given at some starting point (the typical initial value problem), it is possible to form a polynomial that agrees with the solution at the starting point. If this polynomial is considered to be a reasonable approximation to the actual solution over some small range, the value of the solution (including its derivatives) at a point not far from the starting point can be calculated using the polynomial. This process can now be repeated by replacing the initial values by the calculated values and then determining a new approximating polynomial.

RKS is a subroutine that will calculate the value of the approximating polynomial at the end of an interval by means of a fourth order Runge-Kutta method.* The method used is somewhat more involved than that outlined above in that the derivatives are computed at four points within the interval to obtain a better approximation of the values at the end of the interval. The actual method used is not of concern to the user if he understands how to prepare the differential equations for numerical solution, what program steps must be written uniquely for these equations, and how to establish linkage with RKS.

RKS is able to integrate ahead for only one step at a time; if a master control program is provided it is possible after each step to re-enter RKS and thus continue the integration over any desired range.

Like many integration formulae, RKS will handle directly only systems of first order equations. However, any system of differential equations may be reduced to a system of first order equations by the simple trick of changing variables so that only first derivatives appear in the new system. It is then necessary to solve this system of equations explicitly for these first derivatives.

For example, consider the differential equation

$$y'' - y' - 6\,y = 0.$$

As this equation now stands it would not be possible to use RKS to obtain $y$ numerically. By rewriting the equation in the form

$$(y')' - y' - 6\,y = 0$$

and making the substitutions

$$y_1 = y$$

$$y_2 = y' = y_1'$$

the single second order differential equation may be rewritten as a system of two first order equations in terms of the new variables.

- - - - - - - - - -

* See a good book on numerical analysis -- for example:

1. Crandall, S. H., Engineering Analysis, McGraw-Hill, New York, 1956, pp. 160-187.
2. Levy, H. and E. A. Baggott, Numerical Solutions of Differential Equations, New York, Dover, 1950, pp. 143-4.
3. Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, 1956, p. 214.

$$y_2' - y_2 - 6\,y_1 = 0$$

$$y_2 = y_1'$$

By solving these two equations for the derivatives of $y_1$ and $y_2$ in terms only of $y_1$ and $y_2$, the equations will be in the standard form needed for numerical solution.

$$y_2' = y_2 + 6\,y_1$$

$$y_1' = y_2$$

In order for a user to employ RKS, it is necessary that he supply:
(a) the number of equations being solved;
(b) the storage locations of the variables;
(c) the initial values for all variables, and the value of the increment to be used in stepping the independent variable;
(d) at what intervals results are to be listed;
(e) the number of dependent variables to be listed in the results;
(f) the method of evaluating the derivatives of all the variables.

Items (a), (b), (d), and (e) are given in the linkage to RKS, as described below. It is convenient to be able to specify item (d), the number of integration steps between listings of the results, to maintain a reasonable balance between accuracy and quantity of output. While decreasing the step size increases the accuracy of the integration process, the increased number of answers produced quickly passes the point of diminishing returns. Thus it is desirable when using a small step size to take a large number of steps between listings of the results.

In the preceding example, for instance, if it has been decided (by some "guess" or previous knowledge) that a step size of 0.02 is required to achieve the desired accuracy, it would not improve the accuracy of plotted results to have the computer list some 50 pairs of numbers to cover the range from 0 to 1. By having results listed following every fifth integration, the volume of output (and plotting effort) is reduced to a reasonable size, yet adequate accuracy is retained.

For much the same reason, it is desirable to restrict the listing of the dependent variables [item (e)]. If the original problem were a third order differential equation of $y$ as a function of $x$, for instance, RKS would punch out at each specified interval not only $x$, but also 3 $y$'s, corresponding to $y$, $y'$, and $y''$. By setting $y = 1$ in the L+2 word, the latter two could be suppressed and only $x$ and $y$ would be punched.

It should be noted that RKS will supply the output of results without any further programming.

Item (c), the size of the increment and the initial values of the variables, is supplied by placing the appropriate values in the assigned storage locations before transferring to RKS. RKS steps the independent variable and computes the dependent variables using these storage locations.

Item (f), the method of evaluating the derivatives of all the dependent variables, is the heart of the integration system. Within RKS, calculated values are placed in the storage locations assigned to the dependent and independent variables. Control is then transferred from RKS to the Function Evaluation Block (F Block). The F Block must replace each dependent variable by the calculated value of the derivative of that variable as specified by the equation(s) being solved. After computing all the derivatives, control must be transferred to location 003, which allows RKS to continue with the computation.

The actual link between a PAT program and RKS is*.

| | | |
|---|---|---|
| L: | 100 8 L+2 3ff 35f, | |
| L+1: | 000 5 000 000 001, | |
| L+2: | 0pp y L+5 -F- -N-, | |
| L+3: | 000 0 000 000 00n, | |
| L+4: | Increment | |
| L+5: | $x_o$ | Independent variable |
| L+6: | $y_{10}$ $y_1$ | Dependent variable |
| L+n+5: | $y_{no}$ $y_n$ | |

where L+2 is the address of the third link word

pp is the number of integration steps between listings [Item (d) above]. Recommended values are 5, 10, or 20; given as $05_{16}$, $0a_{16}$, $14_{16}$.

y is the number of dependent variables to be punched out at each listing [Item (e) above].

L+5 is the address of the fifth link word (independent variable)

-F- is the address of the first order in the F Block

-N- is the address of the first order in the master control block to be executed following each integration step (often called the "return" )

and n is the number of equations being solved [Item (a) above]

The link given above is commonly used only for the initial transfer to RKS because it causes RKS to set up the desired integration step size, determine certain constants, reset the counter used for determining when to list results, and carry out the integration for one step. Referring to the flow chart on page 16, it can be seen that there are a total of three entrances to RKS. The above link transfers to the first entrance (001) which is the start of the set up block of RKS.

- - - - - - - - - -

* A tape is available which may be fed through the Flexowriter to produce the fixed portion of the link. This tape will stop at each place that the user must supply data which is unique to his program.

(marked in green above)

If the increment, number of points per listing, etc. are not to be changed in the course of the integration, it is more efficient to bypass the set up block in RKS in subsequent linkages. This is possible by transferring to the second entrance (002) from the master control block following each integration step, which will cause RKS to integrate ahead one step.

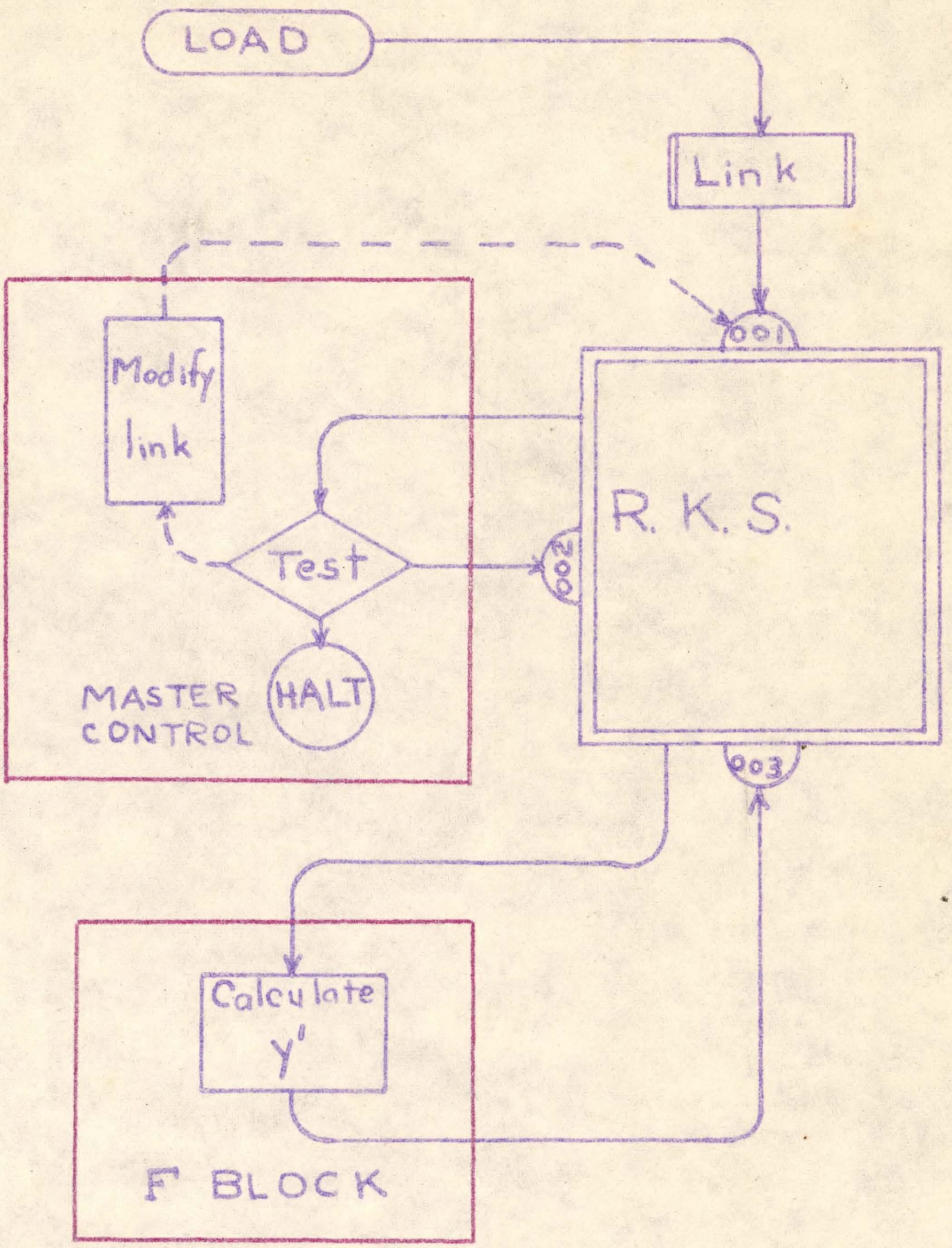
The third entrance (003) is used in conjunction with the F Block as described above.

There are also two exits from RKS: one exit is used to transfer to the F Block in the course of each integration step; the other exit is used to transfer to the master control block following each step.

Although the 6+n words in the link must be stored in consecutive locations, two different loading instructions must be used. The first four link words must be loaded by a RPT instruction; hence, all addresses contained in these words must be given in the hexadecimal equivalents of the decimal addresses used. The remaining n+2 words (the increment and the initial values of the variables) must be loaded by an IN instruction.

RKS uses the entire band of locations from 001 through 150 for its program, and in addition, it uses a band of 2n locations (where n is the number of equations being solved) following location 150 for temporary storage of intermediate results. Furthermore, of course, 5+n consecutive locations are needed for the link itself. Care should be exercised that the PAT program using RKS does not place any instructions or data in memory locations below 150 + 2n .

Special loading procedures (Breakpoint Switches, etc.) must be used to allow PAT to process a program containing a RPT instruction and to insure that RKS is available. The person in charge of the computer will be able to assist the user in these matters. It is further suggested that a member of the computing staff be consulted whenever a system of more than nine equations is being programmed.




Flow Chart for a PAT Program employing R.K.S.

PAT CODING FOR RKS Example

BY Charles W. McClure DATE ______ PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | N | X | T | A | B | R |
|---|---|---|---|---|---|---|---|---|
| | | (RKS must be loaded) | | | | | | |
| | | | | | | | | |
| | LOAD | Load the PAT program; set F.C. = 2 | | 000 | 0 | 250 | 259 | 250 |
| | | | | | | | | |
| | RPT | Load the first four words of the link | 250 | 000 | 0 | 270 | 273 | 251 |
| | IN | Load the remaining words of the link | 251 | 001 | 4 | 274 | 000 | 280 |
| | TRA | Go to the link | 252 | 000 | 5 | 000 | 000 | 270 |
| | | | | | | | | |
| -N- | CHK | Check the computer | 253 | 000 | 4 | 000 | 000 | 000 |
| | TNE | Is $x$ less than $x_f$ ? | 254 | 000 | e | 275 | 278 | 002 |
| | HTR | All through! | 255 | 000 | 6 | 000 | 000 | 255 |
| | | | | | | | | |
| -F- | MPY | Form $6y_1$ | 256 | 000 | 2 | 279 | 276 | 290 |
| | ADD | Form $y_1' = y_2$ | 257 | 000 | 8 | 277 | 280 | 276 |
| | ADD | Form $y_2' = y_2 + 6y_1$ | 258 | 000 | 8 | 277 | 290 | 277 |
| | TRA | Go back to RKS | 259 | 000 | 5 | 000 | 000 | 003 |
| | | | | end. | | | | |
| | | Link to RKS | 270 | 100 | 8 | 110* | 3ff | 35f |
| | | | 271 | 000 | 5 | 000 | 000 | 001 |
| | | | 272 | 005 | 1 | 113* | 100* | 0fd* |
| | | | 273 | 000 | 0 | 000 | 000 | 002 |
| | | Increment | 274 | 102 | 2 | 000 | 000 | 000 |
| | | $x$ | 275 | 000 | 0 | 000 | 000 | 000 |
| | | $y_1$ | 276 | 000 | 2 | 000 | 000 | 000 |
| | | $y_2$ | 277 | 000 | 1 | 000 | 000 | 000 |
| | | | 278 | 000 | 1 | 000 | 000 | 000 |
| | | Constants | 279 | 000 | 6 | 000 | 000 | 000 |
| | | | 280 | 000 | 0 | 000 | 000 | 000 |

*$110_{16} = 272_{10}$; $113_{16} = 275_{10}$

$100_{16} = 256_{10}$; $0fd_{16} = 253_{10}$

SIMULTANEOUS EQUATIONS ROUTINE

Solving a system of linear algebraic equations is a rather straightforward, although laborious, procedure involving considerable repetitious calculation.* It is therefore natural to use a digital computer to obtain the desired results with less labor and fewer chances for human error.

SER is a subroutine that solves a system of linear algebraic equations with real coefficients by the Crout method. The user of SER does not need to understand the method employed if he can prepare the equations for numerical solution and write the link which will cause SER to solve the system.

It is necessary to write the equations in the standard matrix form:

$$a_{11} x_1 + a_{12} x_2 + \ldots\ldots + a_{1n} x_n = y_1$$

$$a_{21} x_1 + a_{22} x_2 + \ldots\ldots + a_{2n} x_n = y_2$$

$$\ldots\ldots\ldots\ldots$$

$$a_{n1} x_1 + a_{n2} x_2 + \ldots\ldots + a_{nn} x_n = y_n$$

The terms of the augmented matrix, defined as the matrix of the coefficients of the system with the column of constants affixed to the right side, must be stored in the computer sequentially by rows before control is transferred to SER. That is, the following terms must be loaded by an IN instruction into a band of consecutive locations:

$$a_{11}, a_{12}, \ldots\ldots, a_{1n}, y_1, a_{21}, a_{22}, \ldots\ldots, a_{2n}, y_2,$$
$$\ldots\ldots, a_{n1}, a_{n2}, \ldots\ldots, a_{nn}, y_n.$$

SER will solve the system of equations described by the augmented matrix and store the solutions obtained in a band of $n + 1$ sequential locations. The value of $x_1$ will be placed in the first location in this band, to be followed by $x_2, x_3, \ldots x_n$. The determinant of the system will also be computed and stored in the location following $x_n$.

The bands of $n(n + 1)$ sequential locations for the terms of the augmented matrix and $(n + 1)$ sequential locations for the results must be chosen with care because of their size. Since SER uses the entire band of locations from 001 through 160, care must be exercised that neither the

\- - - - - - - - - -

* Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, 1956, pp. 424-433.

Kunz, K. S., Numerical Analysis, McGraw-Hill, New York, 1957, pp. 215-243.

Nielson, K. L., Methods in Numerical Analysis, Macmillan Co., New York, 1956, pp. 178-198.

PAT program nor any data is ever placed in locations below 160. Of course the two bands chosen should not overlap each other, nor should they over-write any of the PAT program. It should be further noted that SER destroys the augmented matrix in the process of solution.

The accuracy of the results obtained from the Crout Method cannot be guaranteed because of the limitations of finite accuracy within the computer, round-off, etc. The best results are obtained when the equations are arranged so that $a_{11}$ is the largest of the $x_1$ coefficients.

An approximate analysis could be carried out to obtain a rough figure for the accuracy of the results, but the analysis is as involved as the actual solution. A fairly sensitive indicator of the accuracy of the results is the magnitude of the determinant of the system: if the magnitude of the determinant is much smaller (say a few orders of magnitude) than the solutions, very little accuracy may be anticipated. However, this is not always a reliable guide since the solution of a problem of a limited number of equations may be quite accurate even when the determinant is small. The final check on the accuracy of the solutions is to substitute the solutions back into the original equations.

If poor results are obtained, check with a member of the computing staff. An improved version of SER is available which will produce better results in some cases by the process of back-substitution and iteration. However, even this method will not always give useable results because some matrices are "ill-conditioned" for numerical solution.

The link between SER and the PAT program is:*

```
  L:   100 8 L+2 3ff 35f,
L+1:   000 5 000 000 001,
L+2:   000 n -A- -X- -N-,
```

where L+2 is the address of the third link word

n is the number of equations being solved

-A- is the address of the first location of the band of $n(n + 1)$ sequential locations used for the terms of the augmented matrix.

-X- is the address of the first location of the band of $(n + 1)$ sequential locations to be used for the solutions

and -N- is the address of the first instruction to be executed after all the solutions have been obtained (often called the "return")

Because the link must be loaded by a RPT instruction, all addresses must be converted to the hexadecimal equivalents of the decimal addresses used. In addition, certain procedures (such as Breakpoint Switches, etc.) are required to allow PAT to process a program containing an RPT instruction, and to insure that SER is available. The person in charge of the computer can assist the user in these matters. Furthermore, it is suggested that a member

---

* A tape is available which may be fed through the Flexowriter to produce the fixed portion of the link. This tape will stop at each place that the user must supply data which is unique to his program.

of the computing staff be consulted when a system of more than nine is being programmed.

## LEAST SQUARES N

It is possible to construct many functions to represent emprical data which is given in terms of data points $(x,y)$. Of the simplest functions, the polynomials, many expressions could be found that would represent the data in some "best" manner.* A polynomial of degree $(m - 1)$ can be constructed so that it will pass exactly through $m$ data points, and serve as a smooth approximating function for points in between. If a polynomial of degree less than $(m - 1)$ is used to represent the data, it will not be possible, in general, to have the polynomial pass through all the data points.

A number of criteria may be used to determine what constitutes the "best" representation of the data. One polynomial may pass through the two end points and the mid-point. Another may minimize the difference between the polynomial and the data points. One of the most useful is the polynomial that minimizes the sum of squares of the differences between the polynomial and the data points.

LSN is a subroutine that will determine the coefficients of an $n^{th}$ degree polynomial that will minimize the sum of the squares of the differences between the polynomial and $m$ data points. This polynomial is then known as the least squares $n^{th}$ degree polynomial approximation to the given data. It is not necessary that the user understand the method used to generate the coefficients of this polynomial as long as he is able to call it into use.

In many ways, LSN is the easiest of the subroutines for the user to employ. It is necessary for him to specify to LSN only the degree of the polynomial desired and the number of data points available. LSN will input the data as required, compute the coefficients of the desired polynomial, and punch out the $(n + 1)$ coefficients in order of ascending powers of $x$. After completing this, LSN will transfer back to the beginning of PAT and stop with the PAT display showing; at this time the computer is ready to load a new program.

LSN will produce a meaningful result only if the number of data points is at least one greater than the degree of the desired polynomial, and if all the data points are distinct. However, it is not necessary to supply the data points in any particular order.

Because of the time required for input and handling of the data by LSN, only the minimum number of data points consistent with the required accuracy should be used. Of course it is undesirable to specify a polynomial of degree higher than required for much the same reason. It is useful to note that if the data shows a marked symmetry about some point, a translation of the axis coupled with supplying data from only one side of the new origin will cause the resulting polynomial to represent the entire data. If even symmetry is shown in the data a polynomial of even degree should be specified; if odd symmetry is shown, a polynomial of odd degree.

- - - - - - - - - -

* Hildebrand, F. B., Introduction to Numerical Analysis, McGraw-Hill, New York, 1956, pp. 258-311, 368-423.

  Nielsen, K. L., Methods in Numerical Analysis, Macmillan Co., New York, 1956, pp. 260-265.

PAT CODING FOR SER Example

BY Charles W. McClure DATE ______ PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | N | X | T | A | B | R |
|---|---|---|---|---|---|---|---|---|
| | | (Have SER loaded) | | | | | | |
| | LOAD | Load the PAT program; Set F.C. = 3 | | — | 0 | 160 | 164 | 160 |
| | RPT | Load the link | 160 | — | 0 | 165 | 167 | 161 |
| | IN | Load the data | 161 | 001 | 4 | 180 | — | 191 |
| | TRA | Go to the link | 162 | — | 5 | — | — | 165 |
| | OUT | List the answers | 163 | 002 | 4 | 200 | — | 203 |
| | HTR | Stop | 164 | — | 6 | — | — | 164 |
| | end. | | | end. | | | | |
| | | | | | | | | |
| | | Link to PRS | 165 | 100 | 8 | 0a7 | 3ff | 35f |
| | | | 166 | — | 5 | — | — | 001 |
| | | | 167 | — | 3 | 0b4 | 0c8 | 0a3 |
| | | | | | | | | |
| | | $a_{11}$ | 180 | | | | | |
| | | $a_{12}$ | 181 | | | | | |
| | | $a_{13}$ | 182 | | | | | |
| | | $y_1$ | 183 | | | | | |
| | | $a_{21}$ | 184 | | | | | |
| | | Data $a_{22}$ | 185 | | | | | |
| | | $a_{23}$ | 186 | | | | | |
| | | $y_2$ | 187 | | | | | |
| | | $a_{31}$ | 188 | | | | | |
| | | $a_{32}$ | 189 | | | | | |
| | | $a_{33}$ | 190 | | | | | |
| | | $y_3$ | 191 | | | | | |
| | | | | | | | | |
| | | $X_1$ | 200 | | | | | |
| | | Answers $X_2$ | 201 | | | | | |
| | | $X_3$ | 202 | | | | | |
| | | | 203 | | | | | |

The user will have to write only the link to call LSN into play. The data points must not be loaded by the user's program because LSN will call for the data from the input tape reader when needed. This data must be listed in (x,y) pairs with the independent variable first.

The link between LSN and the PAT program is:

```
  L:   100 8 L+2 3ff 35f,
L+1:   000 5 000 000 001,
L+2:   000 0 00n --m 000,
```

where L+2 is the address of the third link word

n is the degree of the polynomial desired

and --m is the number of data points available.

Because the link must be loaded with an RPT instruction, all data must be given in the hexadecimal equivalents of the decimal numbers used. Since the addresses are already filled in on the standard link to LSN provided in the laboratory, it will be necessary to convert only m from decimal to hexadecimal.

In addition, certain procedures (such as Breakpoint Switches, etc.) are required to allow PAT to process a program containing an RPT instruction and to insure that LSN is available. The person in charge of the computer can assist the user in these matters. Furthermore, it is suggested that a member of the computing staff be consulted when a polynomial of degree higher than nine is being programmed.

# OPERATION TIMES

## PAT and PATRICK programs

When writing long programs it is desirable to be able to estimate the running time that will be required. The following table of operating times can be used as a guide, with the understanding that it will be only approximate because the time required for a special function is somewhat dependent on the value of the function being computed.

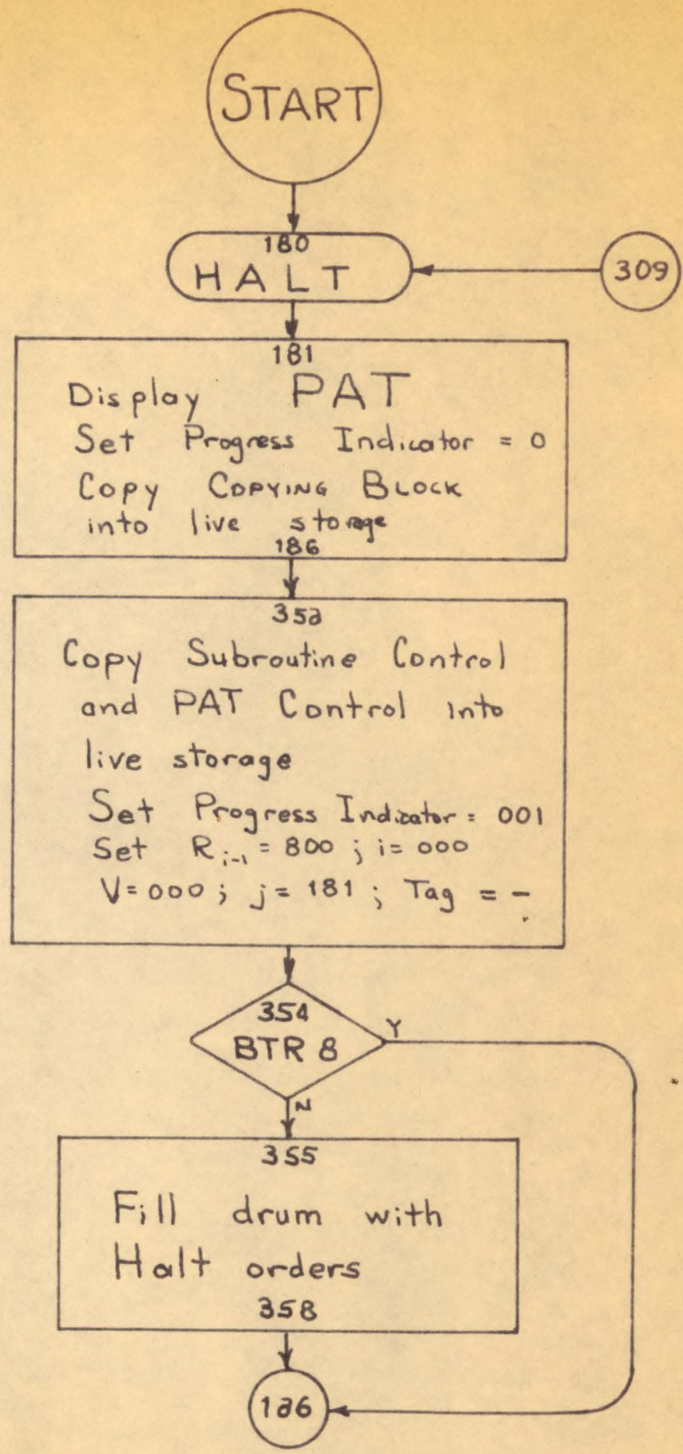
| | |
|---|---|
| LOAD | 39 + 2.7T + 2.1S seconds, where T is the total number of special function and counting instructions, and S is the total number of other instructions. |
| ARITHMETIC | 0.016 seconds. |
| TRANSFER | 0.032 seconds. |
| COMPARISON | 0.032 seconds if transfer does not take place; 0.048 seconds if transfer does take place. |
| IN | 0.6 + 2.1N + 0.9Z seconds where Z is the number of zero, and N is the number of non-zero, numbers being loaded. |
| OUT | 0.6 + 2.7N + 0.9Z seconds where Z is the number of zero, and N is the number of non-zero, numbers being punched out. |
| SQR | 1.2 seconds. |
| SINR, COSR SIND, COSD | 1.8 seconds. |
| ARCR, ARCD | 1.0 seconds. |
| EXPE, EXPT | 1.7 seconds. |
| LOGE, LOGT | 1.6 seconds. |
| DOK | 1.2 seconds. |
| MOVE | 1.2 seconds. |
| RST | 0.9 seconds. |
| CHK | 1.2 seconds. |

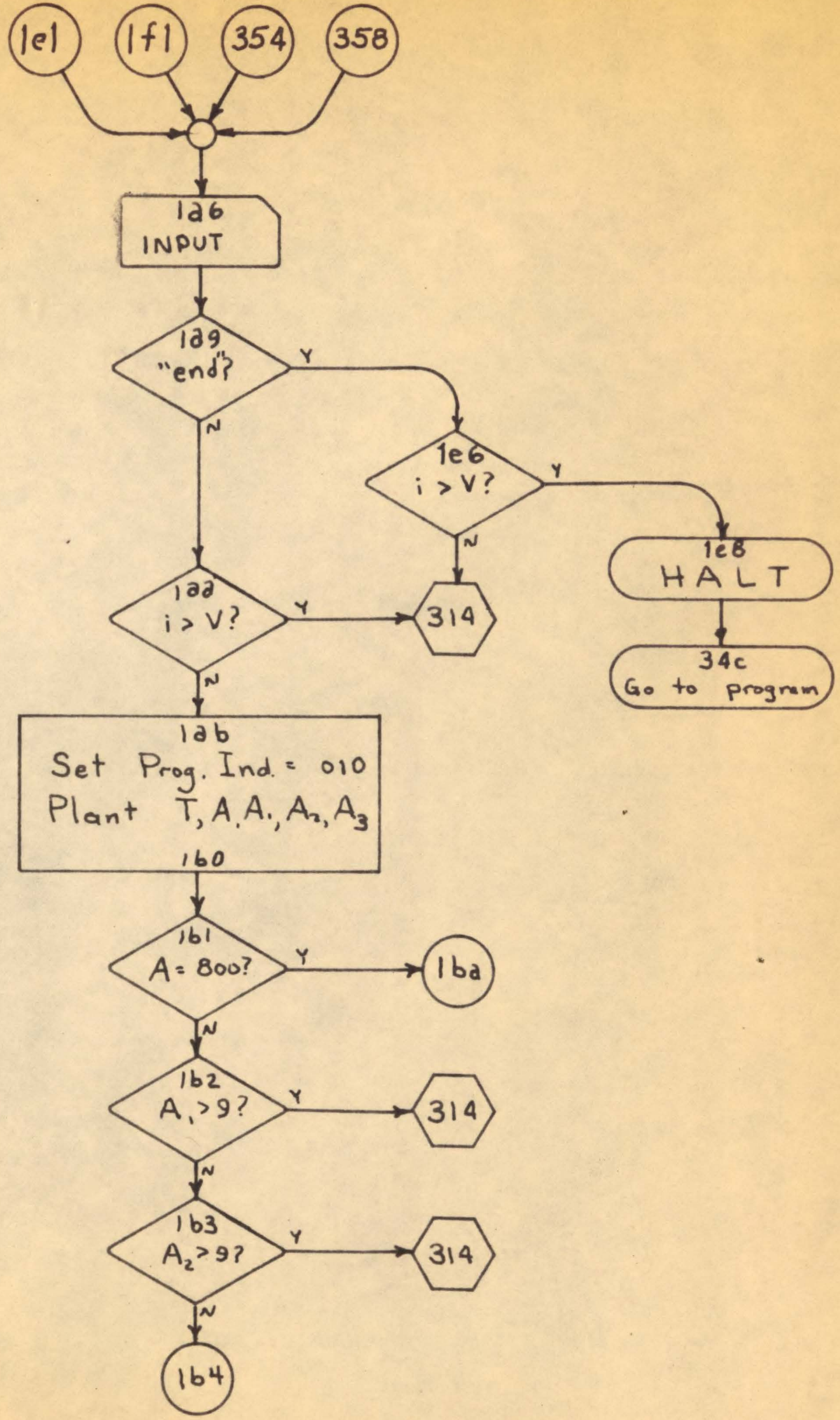
DECIMAL to HEXADECIMAL

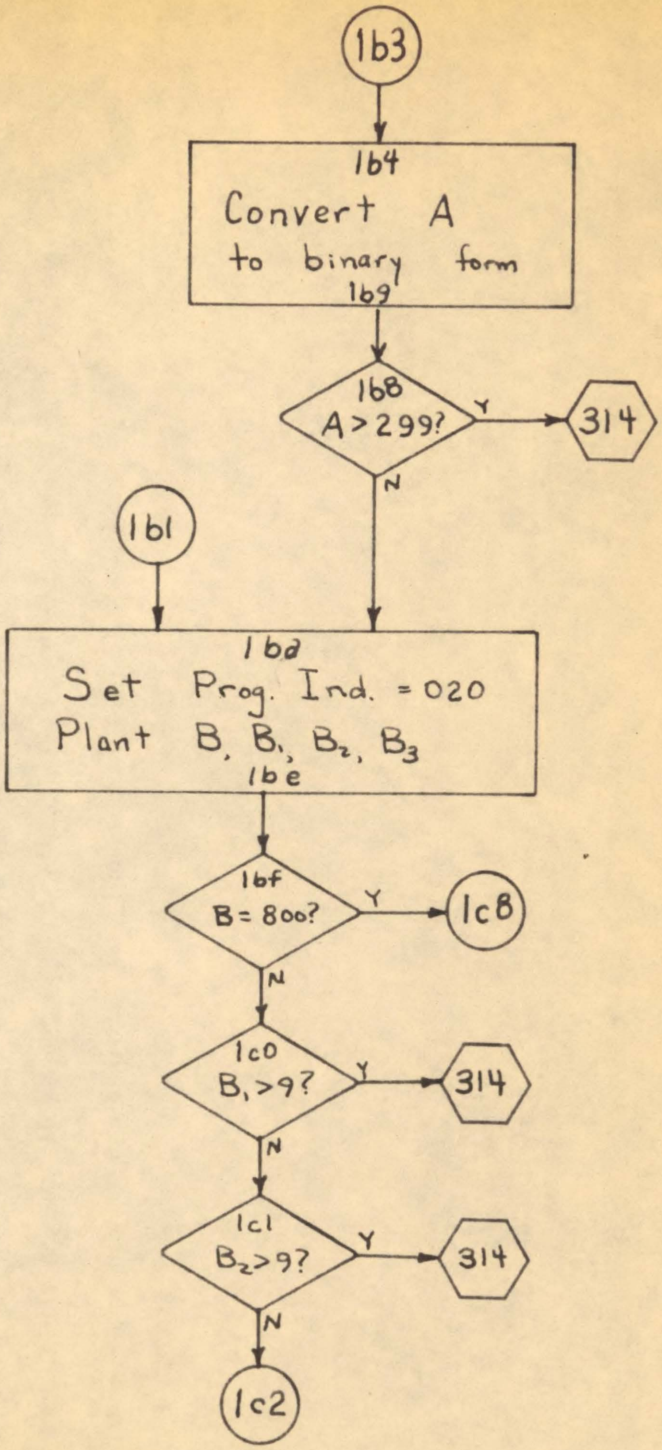
CONVERSION TABLE

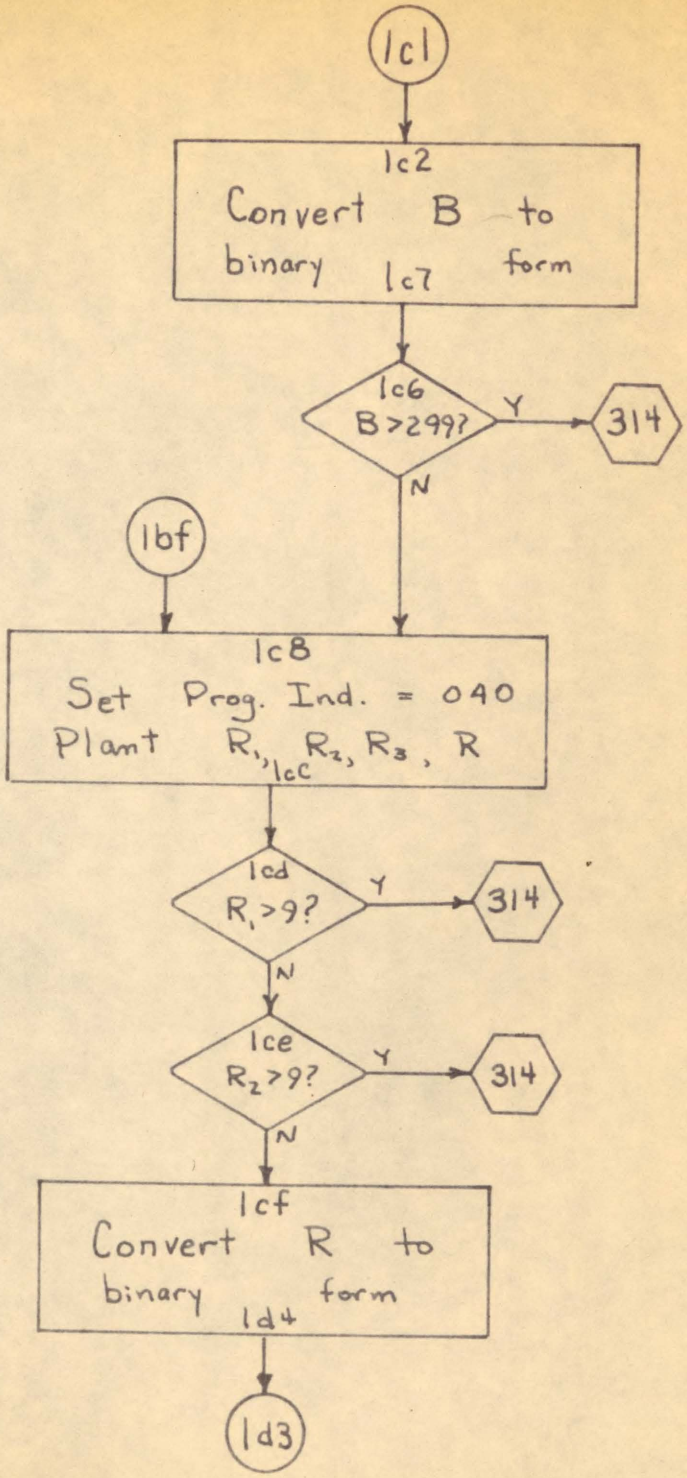
0000-0400

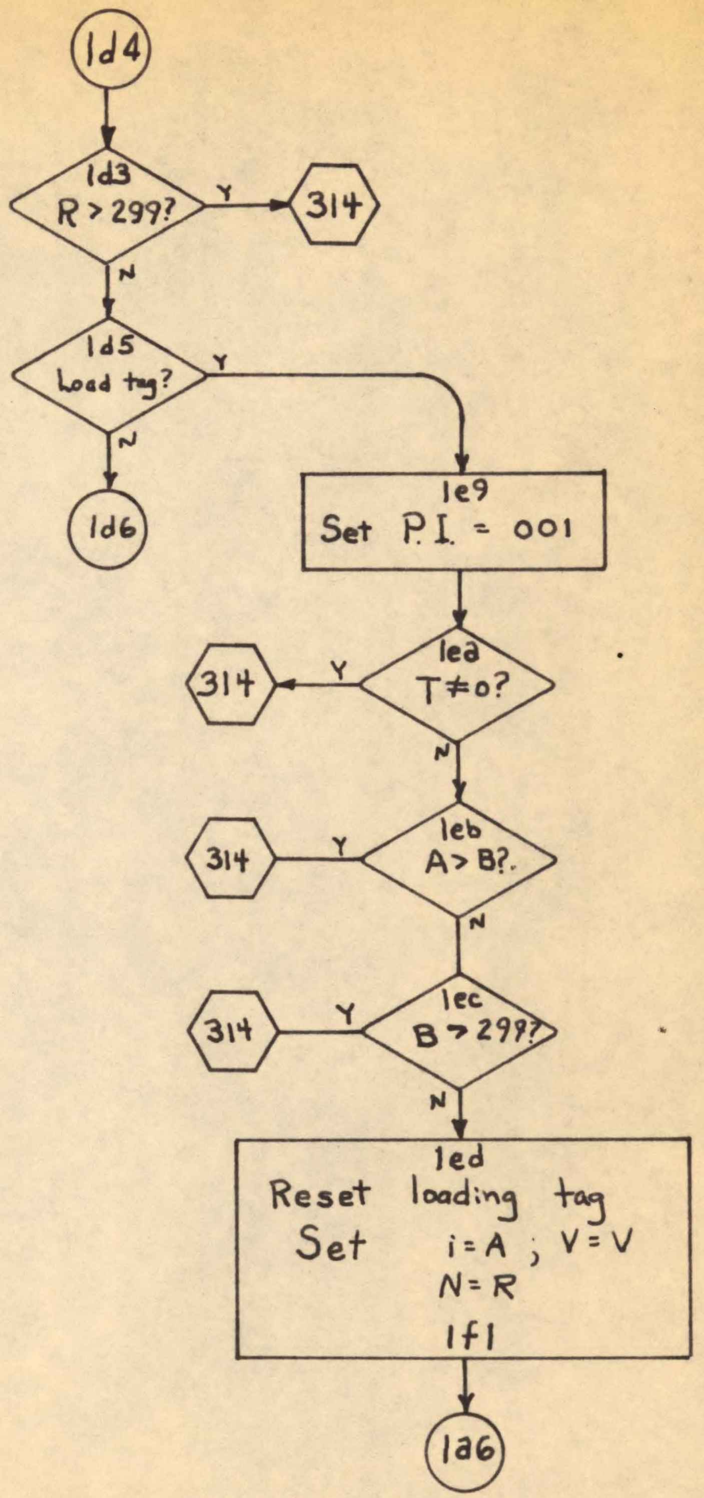
| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 000 | 001 | 002 | 003 | 004 | 005 | 006 | 007 | 008 | 009 |
| 1 | 00a | 00b | 00c | 00d | 00e | 00f | 010 | 011 | 012 | 013 |
| 2 | 014 | 015 | 016 | 017 | 018 | 019 | 01a | 01b | 01c | 01d |
| 3 | 01e | 01f | 020 | 021 | 022 | 023 | 024 | 025 | 026 | 027 |
| 4 | 028 | 029 | 02a | 02b | 02c | 02d | 02e | 02f | 030 | 031 |
| 5 | 032 | 033 | 034 | 035 | 036 | 037 | 038 | 039 | 03a | 03b |
| 6 | 03c | 03d | 03e | 03f | 040 | 041 | 042 | 043 | 044 | 045 |
| 7 | 046 | 047 | 048 | 049 | 04a | 04b | 04c | 04d | 04e | 04f |
| 8 | 050 | 051 | 052 | 053 | 054 | 055 | 056 | 057 | 058 | 059 |
| 9 | 05a | 05b | 05c | 05d | 05e | 05f | 060 | 061 | 062 | 063 |
| 10 | 064 | 065 | 066 | 067 | 068 | 069 | 06a | 06b | 06c | 06d |
| 11 | 06e | 06f | 070 | 071 | 072 | 073 | 074 | 075 | 076 | 077 |
| 12 | 078 | 079 | 07a | 07b | 07c | 07d | 07e | 07f | 080 | 081 |
| 13 | 082 | 083 | 084 | 085 | 086 | 087 | 088 | 089 | 08a | 08b |
| 14 | 08c | 08d | 08e | 08f | 090 | 091 | 092 | 093 | 094 | 095 |
| 15 | 096 | 097 | 098 | 099 | 09a | 09b | 09c | 09d | 09e | 09f |
| 16 | 0a0 | 0a1 | 0a2 | 0a3 | 0a4 | 0a5 | 0a6 | 0a7 | 0a8 | 0a9 |
| 17 | 0aa | 0ab | 0ac | 0ad | 0ae | 0af | 0b0 | 0b1 | 0b2 | 0b3 |
| 18 | 0b4 | 0b5 | 0b6 | 0b7 | 0b8 | 0b9 | 0ba | 0bb | 0bc | 0bd |
| 19 | 0be | 0bf | 0c0 | 0c1 | 0c2 | 0c3 | 0c4 | 0c5 | 0c6 | 0c7 |
| 20 | 0c8 | 0c9 | 0ca | 0cb | 0cc | 0cd | 0ce | 0cf | 0d0 | 0d1 |
| 21 | 0d2 | 0d3 | 0d4 | 0d5 | 0d6 | 0d7 | 0d8 | 0d9 | 0da | 0db |
| 22 | 0dc | 0dd | 0de | 0df | 0e0 | 0e1 | 0e2 | 0e3 | 0e4 | 0e5 |
| 23 | 0e6 | 0e7 | 0e8 | 0e9 | 0ea | 0eb | 0ec | 0ed | 0ee | 0ef |
| 24 | 0f0 | 0f1 | 0f2 | 0f3 | 0f4 | 0f5 | 0f6 | 0f7 | 0f8 | 0f9 |
| 25 | 0fa | 0fb | 0fc | 0fd | 0fe | 0ff | 100 | 101 | 102 | 103 |
| 26 | 104 | 105 | 106 | 107 | 108 | 109 | 10a | 10b | 10c | 10d |
| 27 | 10e | 10f | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 |
| 28 | 118 | 119 | 11a | 11b | 11c | 11d | 11e | 11f | 120 | 121 |
| 29 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 12a | 12b |
| 30 | 12c | 12d | 12e | 12f | 130 | 131 | 132 | 133 | 134 | 135 |
| 31 | 136 | 137 | 138 | 139 | 13a | 13b | 13c | 13d | 13e | 13f |
| 32 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 |
| 33 | 14a | 14b | 14c | 14d | 14e | 14f | 150 | 151 | 152 | 153 |
| 34 | 154 | 155 | 156 | 157 | 158 | 159 | 15a | 15b | 15c | 15d |
| 35 | 15e | 15f | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 |
| 36 | 168 | 169 | 16a | 16b | 16c | 16d | 16e | 16f | 170 | 171 |
| 37 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 17a | 17b |
| 38 | 17c | 17d | 17e | 17f | 180 | 181 | 182 | 183 | 184 | 185 |
| 39 | 186 | 187 | 188 | 189 | 18a | 18b | 18c | 18d | 18e | 18f |

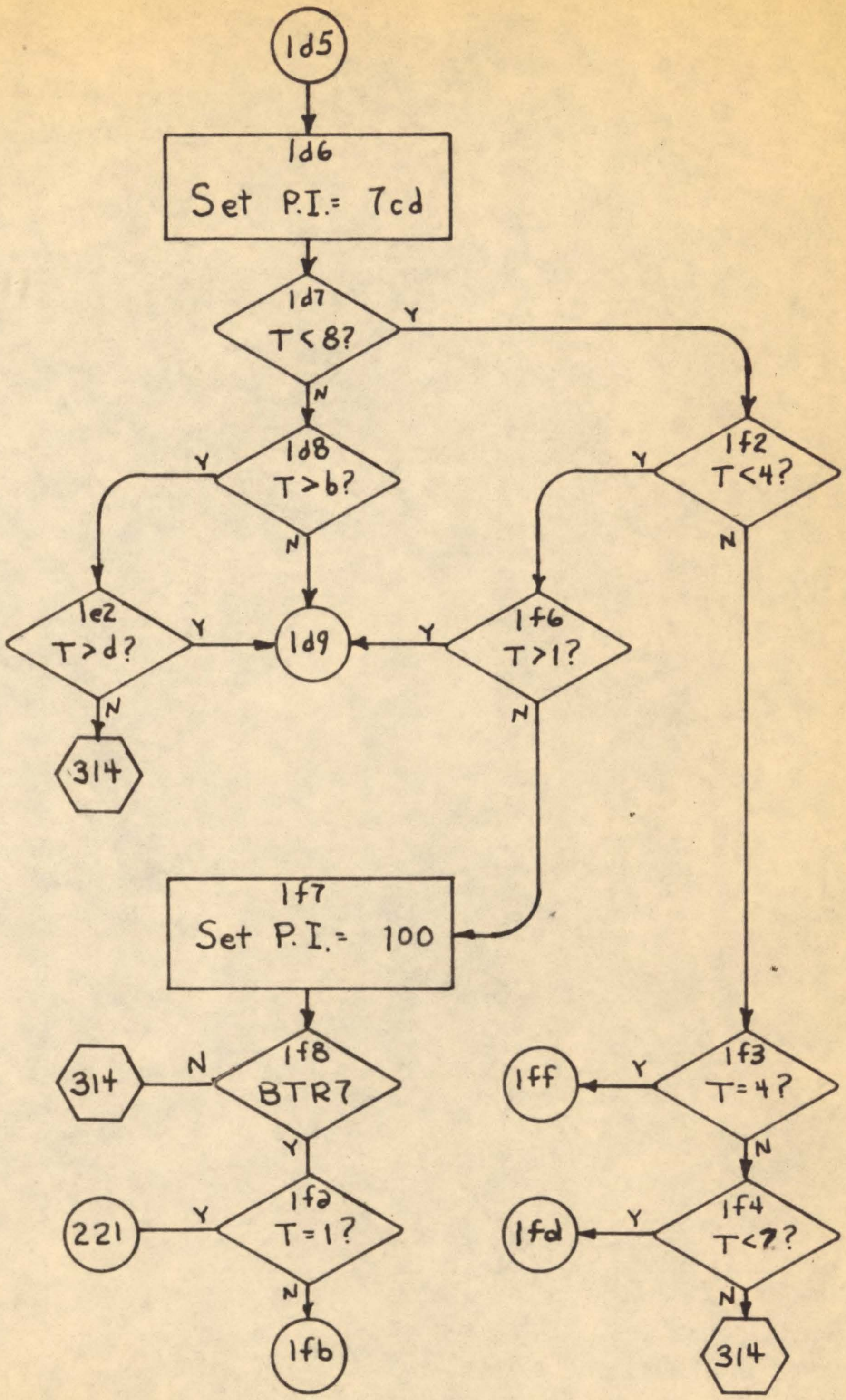

START
180
HALT
309
181
Display PAT
Set Progress Indicator = 0
Copy COPYING BLOCK into live storage
186
352
Copy Subroutine Control and PAT Control into live storage
Set Progress Indicator = 001
Set R i-1 = 800 ; i = 000
V = 000 ; j = 181 ; Tag = -
354
BTR 8
Y
N
355
Fill drum with Halt orders
358
126

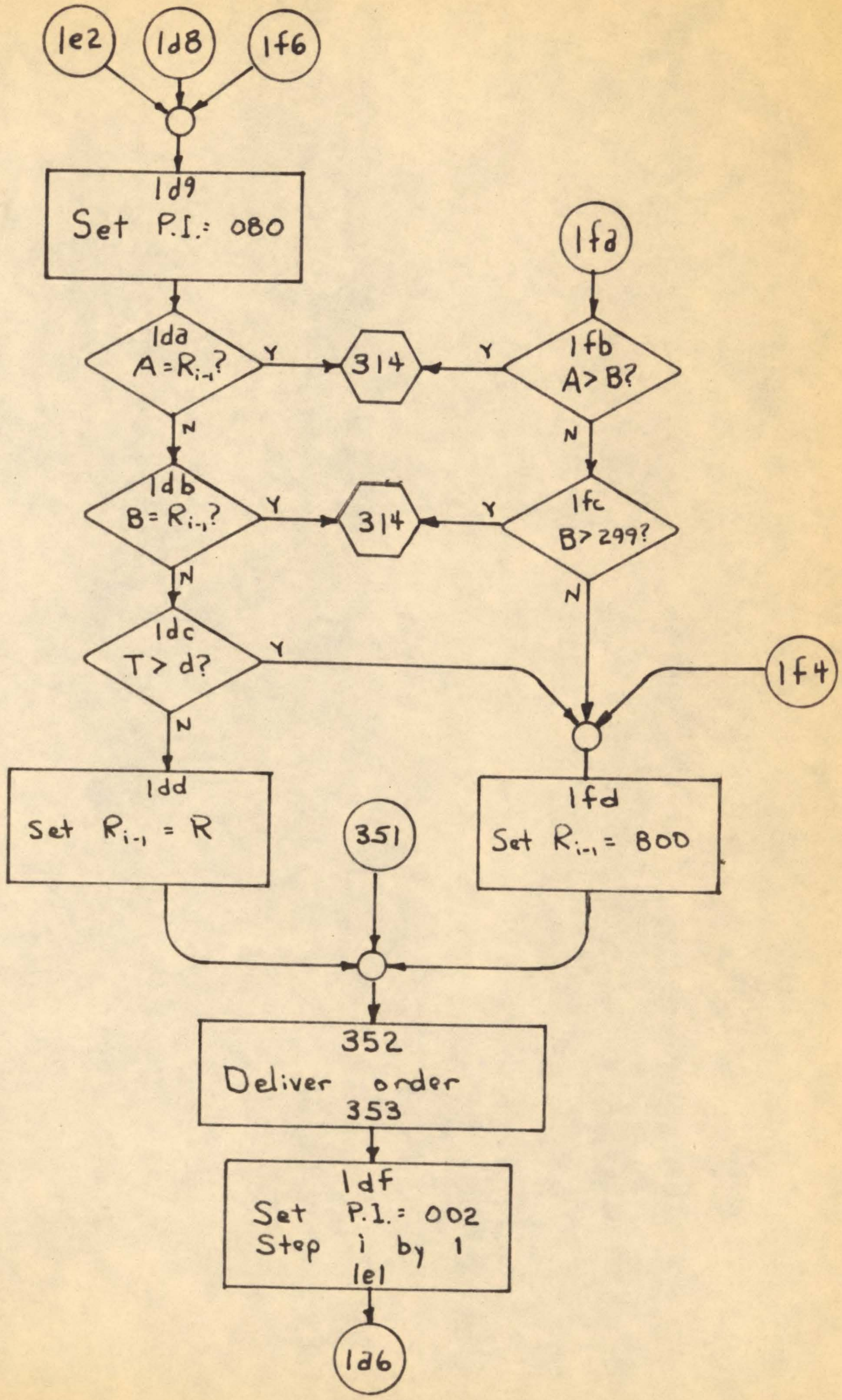

1e1
1f1
354
358
1a6
INPUT
1a9
"end"?
Y
N
1e6
i > V?
Y
N
1e8
HALT
34c
Go to program
1a2
i > V?
Y
N
314
1ab
Set Prog. Ind. = 010
Plant T, A, A., A2, A3
160
161
A = 800?
Y
N
1ba
162
A1 > 9?
Y
N
314
163
A2 > 9?
Y
N
314
164

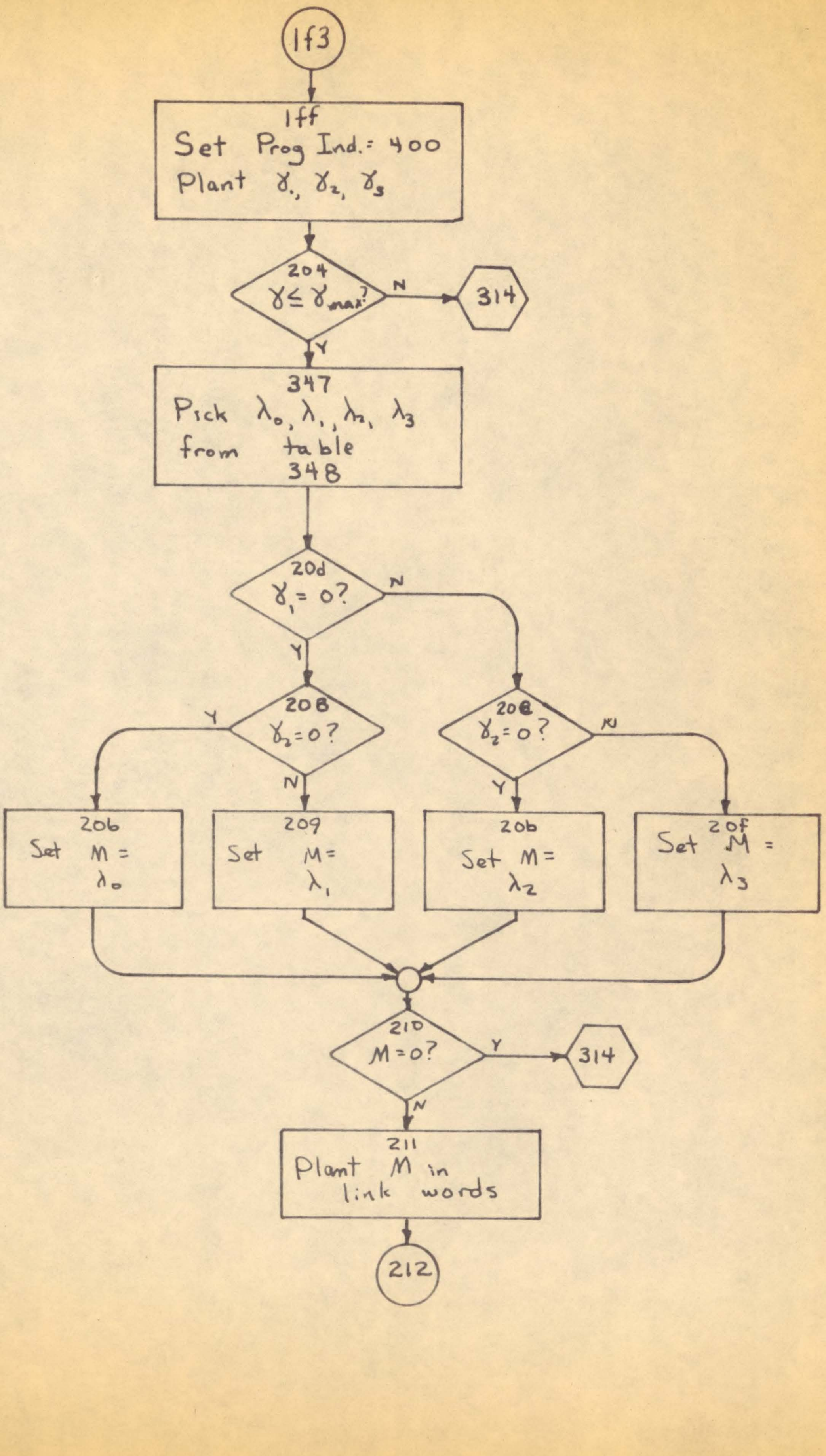

1b3
1b4
Convert A
to binary form
1b9
1b8
A>299?
Y
314
N
1b1
1bd
Set Prog. Ind. = 020
Plant B, B1, B2, B3
1be
1bf
B = 800?
Y
1c8
N
1c0
B1>9?
Y
314
N
1c1
B2>9?
Y
314
N
1c2

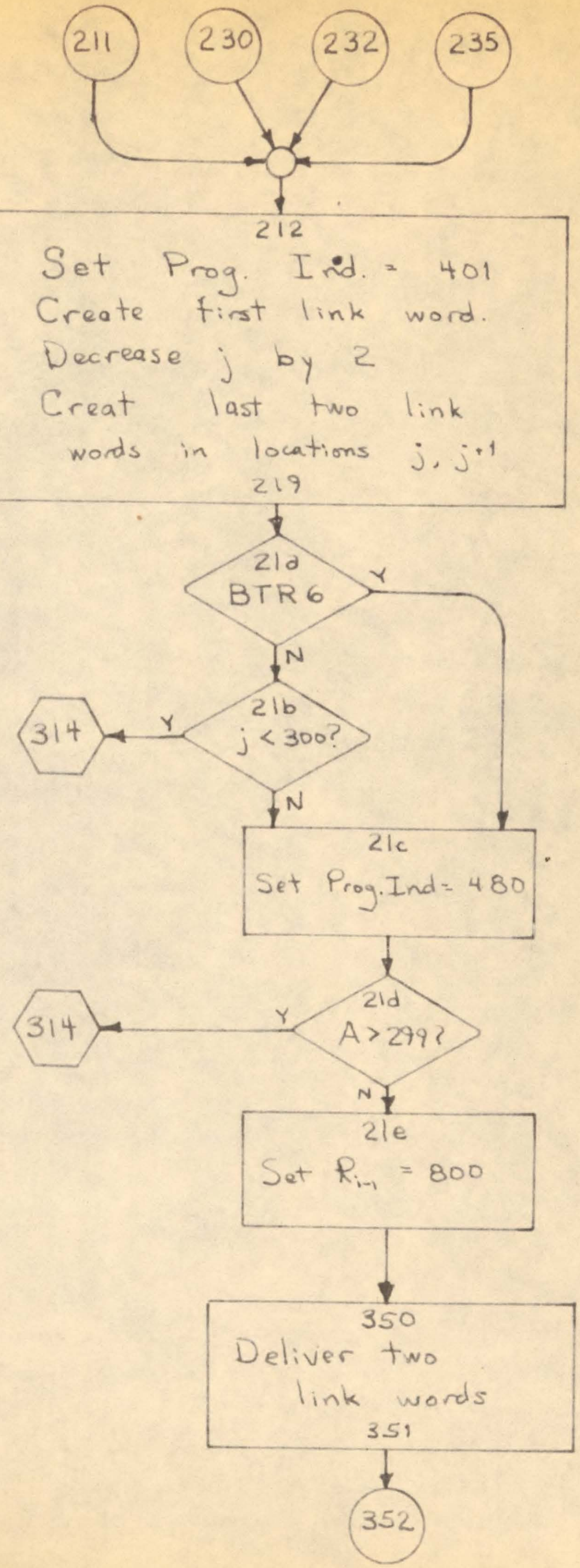

1c1
1c2
Convert B to binary form
1c7
1c6
B>299?
Y
N
314
1bf
1c8
Set Prog. Ind. = 040
Plant R1, R2, R3, R
1cC
1cd
R1>9?
Y
N
314
1ce
R2>9?
Y
N
314
1cf
Convert R to binary form
1d4
1d3

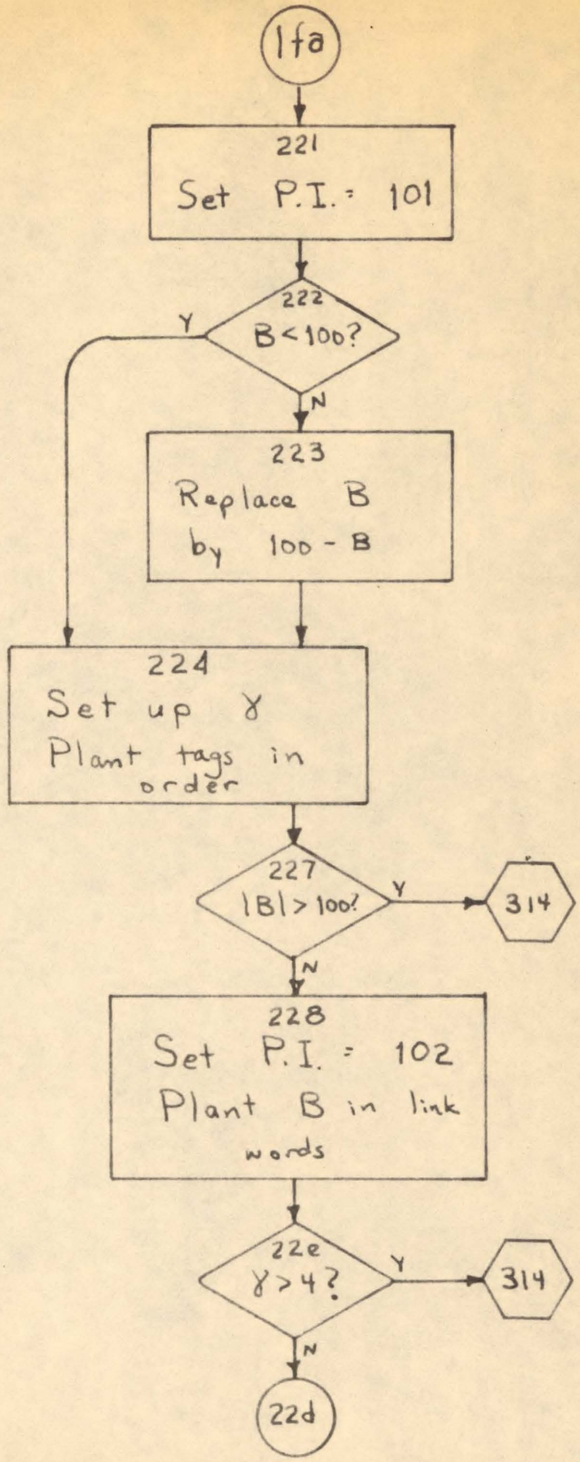

1d4
1d3
R > 299?
Y
314
N
1d5
Load tag?
Y
N
1d6
1e9
Set P.I. = 001
1ea
T≠0?
Y
314
N
1eb
A > B?.
Y
314
N
1ec
B > 299?
Y
314
N
1ed
Reset loading tag
Set i = A ; V = V
N = R
1f1
1a6

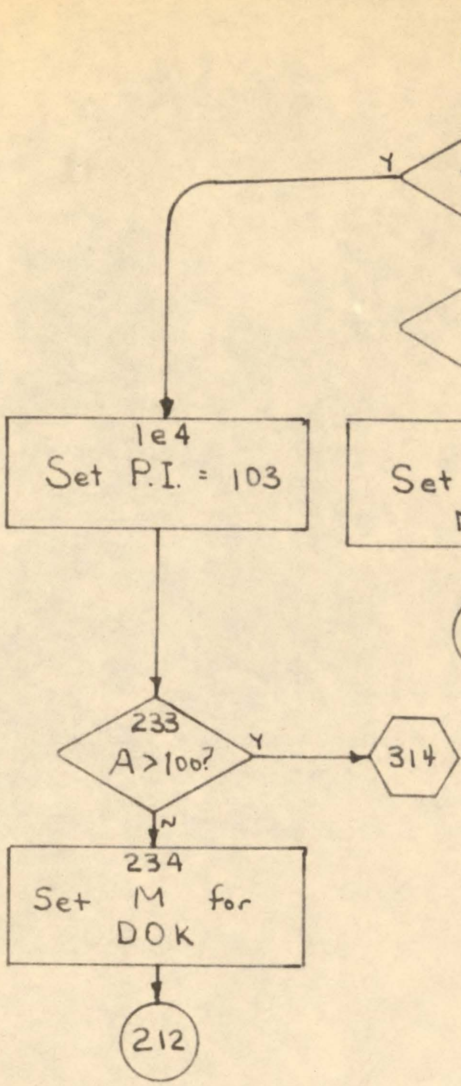

1d5
1d6
Set P.I.= 7cd
1d7
T<8?
1d8
T>b?
1e2
T>d?
1d9
1f6
T>1?
1f2
T<4?
314
1f7
Set P.I.= 100
1f8
BTR7
314
1fa
T=1?
221
1fb
1f3
T=4?
1ff
1f4
T<7?
1fd
314

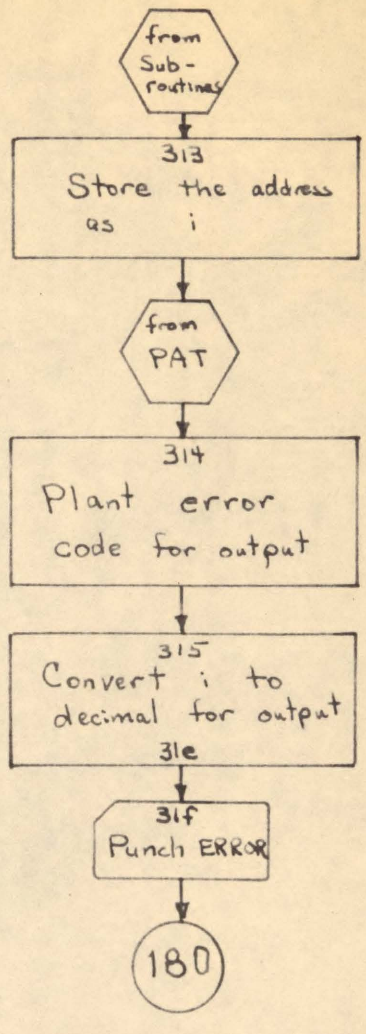

1e2
1d8
1f6
1d9
Set P.I.: 080
1fa
1da
A = R_{i-1}?
Y
314
Y
1fb
A > B?
N
N
1db
B = R_{i-1}?
Y
314
Y
1fc
B > 299?
N
N
1dc
T > d?
Y
1f4
N
1dd
Set R_{i-1} = R
351
1fd
Set R_{i-1} = 800
352
Deliver order
353
1df
Set P.I. = 002
Step i by 1
1e1
1d6


1f3
1ff
Set Prog Ind. = 400
Plant γ1, γ2, γ3
204
γ ≤ γmax?
N
314
Y
347
Pick λ0, λ1, λ2, λ3
from table
348
20d
γ1 = 0?
N
Y
208
γ2 = 0?
Y
N
20e
γ2 = 0?
N
Y
206
Set M = λ0
209
Set M = λ1
20b
Set M = λ2
20f
Set M = λ3
210
M = 0?
Y
314
N
211
Plant M in link words
212


211
230
232
235
212
Set Prog. Ind. = 401
Create first link word.
Decrease j by 2
Creat last two link words in locations j, j+1
219
21a
BTR6
Y
N
21b
j < 300?
Y
314
N
21c
Set Prog. Ind = 480
21d
A > 299?
Y
314
N
21e
Set $R_{i-1}$ = 800
350
Deliver two link words
351
352


1fa
221
Set P.I. = 101
222
B<100?
Y
N
223
Replace B by 100 - B
224
Set up γ
Plant tags in order
227
|B|>100?
Y
314
N
228
Set P.I. = 102
Plant B in link words
22e
γ>4?
Y
314
N
22d


22e
22d
γ=0?
Y
N
22e
γ=4?
Y
N
1e4
Set P.I. = 103
22f
Set M for MOVE
231
Set M for RESET
212
212
233
A>100?
Y
314
N
234
Set M for DOK
212


from
Sub-
routines
313
Store the address
as i
from
PAT
314
Plant error
code for output
315
Convert i to
decimal for output
31e
31f
Punch ERROR
180

| | | | |
|---|---|---|---|
| 000018031f3e8, | | | |
| 0006000000181, | 0044fa47a79f8, | 10083ff3ff35e, | 10081873ff35a, |
| 10001883ff35b, | 10081803ff35c, | 000500000035a, | 100818a3ff340, |
| [illegible] | 0000001000001, | 019135d19c344, | 100835d3fb800, |
| 0191800c1c34c, | 00d135d01c348, | 10080003ff35f, | 019180019a347, |
| 00d135f01a34a, | 10082383ff35b, | 000500000020d, | 100835f3ff000, |
| 100835c3ff000, | 100001000034c, | 0005000000000, | 101135dc10fff, |
| 10080003ff35d, | 2005000000340, | 100834e3ff180, | 10c834f3ff181, |
| 101135dc10000, | 00050000001df, | 10050800001a6, | 100a3f9355000, |
| [illegible] | 100e8001a4355, | 00050000001a6, | 0000000800000, |
| [illegible] | 0005000000354, | 000035f35f1a7, | 100835f3ff35d, |
| [illegible] | [illegible] | 100e34d352314, | 10083eb3ff35e, |
| [illegible] | [illegible] | 119135f0d435a, | 11d135f12435b, |
| [illegible] | 100c3561a31ba, | 100e23635a314, | 100e23735b314, |
| 100235c196800, | 100880035b800, | 1002800196800, | 100880035a356, |
| 100e237800314, | [illegible] | 100835e35e35e, | 10d135f0dc357, |
| 10d135f0d435a, | 111135f124356, | 119135f17435c, | 100c3571a31c8, |
| [illegible] | 100e23735b314, | 100235c196800, | 100880035b800, |
| [illegible] | [illegible] | 100e237800314, | 00d13570dc35d, |
| [illegible] | 101135f0dc358, | 101135f0d435a, | 105135f12435b, |
| [illegible] | 100e23635a314, | 100e23735b314, | 100235c196800, |
| [illegible] | 1002800196800, | 100880035a358, | 100e237800314, |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | 100c350359314, | [illegible] |
| [illegible] | [illegible] | [illegible] | 10083fb3fb35e, |
| [illegible] | 00050000001a6, | 100e1db3591d9, | 0005000000314, |
| [illegible] | 0005000000235, | 100e34d3521e8, | 0005000000314, |
| [illegible] | [illegible] | 100e1a6355314, | 100e357356314, |
| [illegible] | [illegible] | 00d135501c352, | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | 100e1f03551d7, | 10083fa3fa35e, |
| [illegible] | 0005000000314, | 100c3fa355221, | [illegible] |
| [illegible] | [illegible] | [illegible] | 10023fa3f935e, |
| [illegible] | [illegible] | 12a135f01135a, | [illegible] |
| [illegible] | 0005000000314, | 125135b0dc35b, | [illegible] |
| [illegible] | [illegible] | [illegible] | 10d135b0dc35b, |
| [illegible] | [illegible] | 100c35c3ff20b, | [illegible] |
| [illegible] | 00d135b0dc34f, | 10d121a19c35e, | [illegible] |
| [illegible] | [illegible] | 100a3503f3350, | [illegible] |
| [illegible] | [illegible] | 100502040121c, | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |
| [illegible] | [illegible] | [illegible] | [illegible] |

| | | | |
|---|---|---|---|
| 00083583ea358, | 0003359800800, | 0002357800357, | 0008356800356, |
| 0002800312800, | 000f80035724f, | 00023563ea356, | 000235530e800, |
| 000835680035c, | 100e3ff34034a, | 000335c30d35c, | 10093403ff340, |
| 000500000034a, | 10083ff35b35b, | 0005000000249, | 119126019d35e, |
| 0005083000313, | 00113ee321340, | 000935b3ff800, | 000a8003fe355, |
| 00088003ea800, | 0003355800355, | 0002800800356, | 000280027d800, |
| 000880027e800, | 0002800356800, | 000880027f800, | 0002800356800, |
| 0008800280800, | 0002800356800, | 0008800281800, | 0002800356800, |
| 0008800282800, | 0002800356800, | 0008800283800, | 0002800356800, |
| 0008800284800, | 0002800355800, | 000880027c35c, | 032135b32135c, |
| 100e3ff34034a, | 000335c30c35c, | 10093403ff340, | 000500000034a, |
| 000c90fdaa222, | 30784d7e74355, | 105b31651d7e0, | 304e501c3c528, |
| 103c577df40b3, | 3028e6c62d66c, | 102cc40a5b5d1, | 301aaa61be24c, |
| 000fffff4ced2, | 000235b30c800, | 0005000000289, | 000235b30c800, |
| 000a30b800800, | 00038003f2355, | 000f80031028d, | 119128c19d35e, |
| 0005081000313, | 00083fe3ff356, | 000e3ff355291, | 000a3ff355355, |
| 000a3ff356356, | 0008310355800, | 000a800310800, | 000a355800355, |
| 0005000000297, | 000a3ff356356, | 000a3553fe355, | 000e3f8800295, |
| 00023553f2357, | 0002800800800, | 000a3ff80035a, | 00083fe3ff359, |
| 00083573ff358, | 00083593ea359, | 000a8003fe800, | 0003358800800, |
| 000280035a800, | 0003800359358, | 0008357800357, | 0002800312800, |
| 000f80035829d, | 000235735635c, | 000500000034a, | 000235b30d35b, |
| 000380030e355, | 000f30f8002bd, | 00083f83ff356, | 0009355311357, |
| 0011800298356, | 000a357311800, | 000280030e358, | 000b35b800357, |
| 00083ff3ff358, | 00083fe3ff35c, | 00083fe3ff359, | 00083583fe358, |
| 0003357800800, | 0002800359359, | 000835c80035c, | 0002800312800, |
| 000f8003592b3, | 000235635c35c, | 000e3ff35b34a, | 00033fe35c35c, |
| 000500000034a, | 000e3ff35b2c0, | 00083ff3ff35c, | 000500000034a, |
| 11912c119d35e, | 0005082000313, | 000935b3ff35e, | 000a35c35c35c, |
| 000c35b3ff34a, | 00083ee3ff35c, | 02a135b29735c, | 031135b31135c, |
| 000335e35c800, | 000a80035c800, | 00028003f835a, | 000880035c35c, |
| 0002800312800, | 000f80035a2c8, | 032135b32135c, | 000500000034a, |
| 100b3ff35f35f, | 119135f0dc355, | 10d135f0dc356, | 00d12d501c35f, |
| 100e3563552d6, | 000500034b360, | 11912d719d35e, | 0005001000313, |
| 10082d93ff35f, | 000500034b3d3, | 100835f3ff35c, | 019135d01c34a, |
| 129135f0d8355, | 119135f0d8356, | 10083553fc355, | 10088003fc800, |
| 100e3568002e4, | 00d135529835c, | 00d135f01c34c, | 000500000034a, |
| 10083ff35f35c, | 000500000034a, | 100835b3ff35c, | 019135d01c349, |
| 129135f0d8355, | 129135d0d7356, | 030135d321356, | 119135f0da357, |
| 10d135f0da358, | 100e2403552fa, | 10083553fc800, | 00d180029835f, |
| 100e2373572fa, | 100e2373582fa, | 123135f011357, | 117135f011358, |
| 100c3573ff2f7, | 10033563eb800, | 100835f80035f, | 100c3583ff349, |
| 100835f35635f, | 0005000000349, | 10d12fb19d35e, | 0005001004313, |
| 10080003ff35c, | 019134701c349, | 129135b1b8355, | 10083ff35b35f, |
| 10088003fd800, | 001180019c347, | 00d130301c348, | 0005000304347, |
| 129135b1b7356, | 030135b321356, | 1002355356800, | 100a3ff800356, |
| 00050000002f2, | 2005000000180, | 0000000000000, | 001c90fdaa222, |
| 1058efa351296, | 002935d8dddab, | 000b17217f7d2, | 008feffffffff, |
| 0298000000000, | 0288000000001, | 1108000000000, | 019134401c352, |
| 119135e25d35e, | 101135209a354, | 1003354240800, | 10088003fc354, |
| 025180009435e, | 00113ff254354, | 1002354196354, | 020180005435e, |
| 00113ff600354, | 1002354196800, | 01b180001435e, | 000735e35f309, |

# USING THE FLEXOWRITER

The Flexowriter electric typewriter is used to prepare program tapes for the computer and to list the answers and error punchouts produced by the computer. The following important points should be kept in mind when using the Flexowriter.

(1) Always use three sheets of paper in the Flexowriter to protect the roller.

(2) Every key on the keyboard produces a unique combination of holes on the tape when the PUNCH switch is on.

(3) All orders and numbers must consist of 13 characters (ignoring spaces and a decimal point) AND a comma. It is desirable to insert spaces in an instruction to separate the parts of the order. It is also desirable to insert the decimal point (period) in a number, and to insert spaces to make numbers the same length as orders.

(4) Errors in typing may be corrected either by over-punching (if this is possible) or by removing the incorrect character with the CODE DELETE switch and typing the correct character. It is not good practice to try to glue holes back onto the tape!

(5) Place a Stop Code at the end of the tape using the STOP CODE switch. Advance the tape using the TAPE FEED switch. Place some identification (including the name of the typist) on the tape as unlabeled tapes WILL BE DESTROYED AT ONCE!

(6) Prepare a listing of the tape using the reading feature of the Flexowriter. Always proof-read the listing so produced to guarantee that the tape correctly represents the written program.

(7) It is possible to reproduce a tape using the combined reading and punching feature of the Flexowriter. The START READ switch will prevent reading while it is depressed; the STOP READ switch will terminate reading when it is depressed.

# PAT OPERATING INSTRUCTIONS

The console of the computer is divided into three areas:

(1) The computer console is the center third of the console and is used to control the computer proper.

(2) The input console is the left third of the console and is used to control the input tape reader, which is located near the computer console.

(3) The output console is the right third of the console and is used to control the output tape punch, which is located at the right side of the console.

All switches will be referred to in this manual by stating the name of the switch in capital letters within quotation marks separated by double spaces from the remainder of the text; the same name appears on the console by the switch.

Only those switches mentioned by name in this manual are to be used by PAT users. While a PAT user may open the input and output console doors to check the state of certain neon displays, no switches nor tubes located within these consoles are to be touched.

Every user should have four items ready when he arrives to use the computer:

(1) a program that the user feels is correct,

(2) a tape which is a faithful copy of the program,

(3) a listing of the tape, and

(4) some sort of a check for the computer results.

In order to load a PAT or PATRICK program into the computer the following steps must be performed in the order listed. Extreme care must be exercised to ensure that all steps are executed in the exact manner described. Notify the engineer in charge if the proper execution of any step does not result in the indicated action.

## STARTING

1. Sign in on the computer log.

2. Stop the computer by pushing the "STOP" button on the computer console.

3. Clear the computer by pushing the "CLEAR" button on the computer console at least three times.

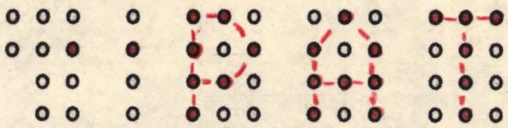
4. Place a "Start PAT" tape into the input tape reader, being careful to have the tape under the guides and the drive wheel engaging the sprocket holes in the tape. (The "Start PAT" label will be readable from the front of the console when the tape is properly inserted into the reader.)

5. Push the "CLEAR EMPTY" button on the input console.

6. Turn on the tape reader motor with the toggle switch located to the left of the tape reader.

7. Start the tape reader by pushing the "START" button on the input console; the tape reader will stop automatically when it reaches the end of the tape.

8. Start the computer by pushing the "RUN" button on the computer console. The computer will stop at once with the PAT display shown below displayed in the neon bulbs located on the computer console.



9. Remove the "Start PAT" tape from the input tape reader.

10a. If no subroutines nor counting instructions are to be used in the program to be loaded, throw all "BREAKPOINT SWITCHES" on the computer console down; skip to step 26.

10b. If counting instructions but no subroutines are to be used in the program to be loaded, throw "BREAKPOINT SWITCH # 7" on the computer console up and all other "BREAKPOINT SWITCHES" down; skip to step 26.

10c. If it is known that the subroutine to be used with the program to be loaded is available,[1] throw "BREAKPOINT SWITCHES # 7 and # 8" on the computer console up and all other "BREAKPOINT SWITCHES" down; skip to step 26.

10d. If the subroutine to be used with the program to be loaded is not available,[2]* throw all "BREAKPOINT SWITCHES" on the computer console down; proceed to step 11.

1 i.e., ~~in memory already~~ already stored in memory

*2 i.e., not already in memory ~~not~~

# LOADING A SUBROUTINE

11. Start the computer by pushing the "RUN" button on the computer console. This will cause PAT to erase all traces of previous programs and subroutines.

12. Locate the desired subroutine tape on the top of the console. All subroutines are on heavy red tape and are identified by a label giving the name of the subroutine on the leader.

13. Place the subroutine tape into the input tape reader, being careful to have the tape under the guides and the drive wheel engaging the sprocket holes in the tape. (The label on the tape will be readable from the front of the console when the tape is properly inserted into the reader.)

14. Wait until the computer completes erasing the previous program. This will be indicated by the computer ceasing computation while the RUN light neat the top of the console remains on.

15. Stop the computer by pushing the "STOP" button on the computer console.

16. Clear the computer by pushing the "CLEAR" button on the computer console at least three times.

17. Start the input tape reader by pushing the "START" button on the input console. The tape reader will stop automatically when it reaches the end of the subroutine tape.[3*]

18. Start the computer by pushing the "RUN" button on the computer console. This causes the computer to load the subroutine for future use.

19. The computer will stop automatically when the input tape reader stops. The STOP light located near the top of the console will come on. At this time remove the subroutine tape from the input tape reader and rewind it.

- - - - - - -

[3*] It is convenient to wind long subroutine tapes unto the rewind spool while the tape is being read by the reader.

20. Clear the computer by pushing the "CLEAR" button on the computer console at least three times.

21. Place a "Start PAT" tape into the input tape reader, being careful to have the tape under the guides and the drive wheel engaging the sprocket holes in the tape. (The "Start PAT" label will be readable from the front of the console when the tape is properly inserted into the reader.)

22. Start the input tape reader by pushing the "START" button on the input console; the tape reader will stop automatically when it reaches the end of the tape.

23. Start the computer by pushing the "RUN" button on the computer console. The computer will stop at once with the PAT display shown in step 8 displayed in the neon bulbs located on the computer console.

24. Remove the "Start PAT" tape from the input tape reader.

25. Throw "BREAKPOINT SWITCHES # 7 and # 8" on the computer console up and check that all other "BREAKPOINT SWITCHES" are down.

## LOADING A PROGRAM

26. Start the computer by pushing the "RUN" button on the computer console. This will cause PAT to prepare to examine the program to follow.

27. Set the Format Control (identified as "WORDS/LINE" ) on the output console to the desired number of columns of results.

28. Push the "CLEAR EMPTY" button on the output console. This button must be depressed quite firmly for proper operation. Call the engineer in charge to check the proper operation of this button if in doubt.

29. If any punched tape has been left in the output punch, advance the tape by depressing the "TAPE FEED" button on the output console. Tear off the unwanted tape and discard it.

30. Place the program tape into the input tape reader, being careful to have the tape under the guides and the drive wheel engaging the sprocket holes in the tape. (The line of printing on the tape will be visible from the front of the console when the tape is properly inserted into the reader.)

31. Start the tape reader by pushing the "START" button on the input console. The tape reader will stop automatically only when it reaches a stop code on the tape. PAT will now check the program for logical errors and store the program in the memory of the computer.

32a. The computer will stop without producing any output if no logical errors are found in the program.

Reset the timer located above the "BREAKPOINT SWITCHES" if it is desired to obtain the running time of the program. This timer is controlled by the computer and runs whenever the computer is running.

B-6

space 25 Start the computer by pushing the "RUN" button on the computer console. This causes the computer to start executing the program.

When all the program tape has been read by the input tape reader, turn off the tape reader motor with the toggle switch to the left of the tape reader. Remove the program tape from the reader.

Follow steps 33-39 when it is desired to remove an answer tape from the output punch.

32b. The computer will stop with the PAT display showing in the neon bulbs on the computer console ONLY if a logical error has been found.

Stop the input tape reader by pushing the "STOP" button on the input console. Turn off the tape reader motor with the toggle switch to the left of the tape reader. Remove the program tape from the reader.

Follow steps 34-39 to remove the error tape from the output punch.

Identify the error using the error codes listed in another part of this manual. (It is convenient to be able to read an error tape without listing it via the Flexowriter. ~~See the section on reading tapes for further information.)~~ Check with the engineer in charge for instruction on reading tapes.)

Correct the error in the program tape and arrange to use the computer again.

## REMOVING A TAPE

33. Stop the computer by pushing the "STOP" button on the computer console whenever a tape is to be removed from the output punch.

34. Wait until the punch stops punching if it is still punching information onto the tape.

35. Place a stop code on the tape by depressing and releasing the "STOP CODE" button on the output console.

36. Advance the tape by depressing the "TAPE FEED" button on the output console until all the information punched on the tape is beyond the end of the punch.

37. Tear off the tape and put some identification (including both the user's name and the date) on it at once. The engineer is charge will destroy all unlabeled tapes immediately and will destroy all properly labeled tapes after 48 hours.

38. If it is desired to continue with the computation, restart the computer by pushing the "RUN" button on the computer console.

39. Relinguish the computer and sign out on the computer log as soon as you are through with the computer. This means both when you finish a problem and when you leave the computer to correct an error!

# USE of BREAKPOINT SWITCHES

All switches are normally down for PAT.

BP 8 If this switch is up the drum will not be filled with the halt orders to catch any transfers outside the program. This feature will have to be used whenever an additional subroutine (RKS, PRS, etc.) is used because that subroutine will have to be loaded before control is given to PAT.

It is suggested that "Start PAT" be fed in which will cause the drum to be filled with halt orders; Then the desired subroutine be loaded; Then "Start PAT" be fed in again this time with BP 8 up; followed by the program.

BP 7 If this switch is up PAT will allow all order type 0 and 1 orders to be processed. This feature distinguishes between PAT programs and PATRICK programs.

*This switch should be down for PAT p[rogram]s and up for PATRICK.*

BP 6 If this switch is up PAT will allow an unlimited number of subroutine links. Normally only 42 type 4 and type 1 orders may be employed because the links for these orders (two link words per order) are stored in the band from 19c to 17f. Further subroutine links will be placed in PAT space starting at the top of the space and working down.

BP 5 If this switch is up the computer will halt after completing each subroutine. The result of the subroutine will have been delivered when the computer halts, and the order in the order register will be the transfer instruction to return to the PAT program.

All other breakpoint switches may be used in other subroutines (such as RKS, PRS, etc.) to control iteration, accuracy limits, form of solutions, etc.

# IDENTIFYING AN ERROR

~~PROCESSING A PROGRAM~~

After loading a program as described on the preceding pages, PAT will examine the program for logical errors. It is, of course, impossible to detect all errors because PAT can not know what problem is being solved, but it is possible to detect all violations of the basic rules of PAT programming.

If no errors are found in the program, the computer will stop with a transfer instruction displayed in the neons on the computer console. The N address of this transfer instruction will be the address specified by the LOAD instruction as the first order of the program to be executed. This address will be specified as a hexadecimal number, and may be converted to a decimal number by the table in the section on HEXADECIMAL NUMBERS.

If an error is found in the program, PAT will prepare an error punchout describing the type of error, the address of the instruction containing the error, and a copy of the incorrect instruction. After producing the error punchout the computer will stop with the PAT display showing in the neons on the computer console. PAT is now ready to examine another program.

As indicated above, an error punchout will consist of two words containing an error key describing the error found. ~~Discard the first three characters of the first word (which will always be zeros);~~ The ~~next~~ first four characters are the error key; discard the next ~~three~~ six characters (which will always be zeros); the last three characters give the address of the incorrect instruction. The second word will be a copy of the incorrect order.

The following pages list all the error keys produced by PAT.

It is possible to have the program correct, but to ask for an impossible special function (such as the logarithm of a negative number). Such errors will produce error punchouts during running. The meaning of these error keys are also listed. It should be noted that the order calling for this impossible operation will be punched out with the error key, but it will be in hexadecimal number form.

## ERROR KEYS

~~Check to see that all Debugging Switches are~~ OK

0000 Program started with special order "end"

0001 Loading instruction was incorrect

- (a) Order type was not 0
- (b) -A- was greater than -B-
- (c) -B- was greater than 299

0002 Loading instruction does not agree with length of tape

- (a) If the address in the error punchout is less than or equal to -V- in the loading instruction, there were not enough words between the loading instruction and "end"
- (b) If the address in the error punchout is greater than -V- in the loading instruction, there were too many words between the loading instruction and "end"

0010 Order has an impossible A address

- (a) Some character was greater than 9
- (b) Total address was greater than 299 and not short memory

0020 Order has an impossible B address

- (a) Some character was greater than 9
- (b) Total address was greater than 299 and not short memory

0040 Order has an impossible R address

- (a) Some character was greater than 9
- (b) Total address was greater than 299

0080 Order has an incorrect use of short memory in a normal order

- (a) Should have used 800 to pick up preceding result
- (b) Should not have used 800 to pick up preceding result

0100 Order has an incorrect use of order type 0 or 1

- (a) Breakpoint Switch 7 was not up
- (b) Order type 0 with -A- greater than -B-
- (c) Order type 0 with -B- greater than 299

0101 Order called for a MOVE order with a step of greater than 100.

0102 Order called for a non-existent type of counting instruction

0103 Order called for a DOK order with K greater than 100

Can all this be condensed to one page if single spaced with narrower margins?

ERROR KEYS (Cont.)

0400 Order called for a non-existent type of special function

0401 Order would have called for more than 42 subroutine links. Every order type 4 and order type 1 generates a subroutine link. Breakpoint switch 6 allows more links at the expense of program space; consult with a staff member for further details.

0480 Order called for preceding result with short memory in a special function or counting instruction.

07cd Order used illegal order type 7, c, or d.

1001 An IN or an OUT order specified -F- greater than -L-

1004 The MOVE instruction has been used more than 100 times without being reset; or the MOVE instruction used an -A- or -R- address outside the range of PAT (000-299)

1081 The angle being specified for the sine or cosine special functions was outside the range of computability

1082 The number being specified for the exponential special functions was outside the range of computability

1083 The number specified for the logarithm special functions was less than or equal to zero.

If the computer ever halts during the running of a program, check the following error stops which do not produce error punchouts.

A machine error detected by the CHECK order. The computer will stop with 3d4 in the order counter and 100 9 3e5 3ff 35e in the order register. Call the engineer in charge.

An unnormalized number has been delivered to DECON. Computer will stop with 3b9 in the order counter and 200 6 000 000 3b8 in the order register. Must restart PAT. Check the range of all OUT orders.

## HALTS

If the computer ever halts during the running of a program, check the following error stops, which do not produce error punchouts, and follow the suggested procedure.

A machine error detected by the CHECK instruction.

The computer will stop with 3d4 in the order counter and 100 9 3e5 3ff 35e in the order register. Call the engineer in charge.

An unnormalized number has been delivered to DECON.

The computer will stop with 3b9 in the order counter and 200 6 000 000 3b8 in the order register. "Start PAT" must be fed in, causing PAT to load a new program as described in STARTING PAT. Examine the program to find an OUT instruction that caused a number to be delivered to DECON that had not been generated by the program.

A transfer outside of the program.

The computer will stop with 200 6 3f9 355 -i- in the order register, where -i- is the address to which transfer has taken place. "Start PAT" must be fed in, causing PAT to load a new program as described in STARTING PAT. Examine the program to find a transfer outside the range of the program. PAT fills all unused locations with halt orders to catch any transfers beyond the program.

An incompleted IN order.

If an IN order is being executed and there are not enough words supplied on the tape, the computer will "hang up" with the RUN light on and 000 0 000 35e 35d in the order register.

## HEXADECIMAL NUMBERS

The computer actually works only with binary numbers, and all decimal numbers supplied to it are converted to binary notation. However, to facilitate human understanding, the binary bits are grouped into sets of four bits each, and the hexadecimal number system is used for reading the binary numbers.

The orders displayed in the order register neons on the computer console are grouped into sets of four bits vertically, and the entire 50 bit (13 hexadecimal characters) word can be read from the neons. The light patterns for the sixteen combinations and the hexadecimal equivalents are shown on ~~the next~~ this page. Once an address has been read from the neons (as the R address of the transfer order will causes PAT to transfer to the program after loading), the hexadecimal-decimal conversion tables can be consulted to find the corresponding decimal address. The ~~second~~ page following is such a conversion table.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| o | o | o | o | o | o | o | o |
| o | o | o | o | ● | ● | ● | ● |
| o | o | ● | ● | o | o | ● | ● |
| o | ● | o | ● | o | ● | o | ● |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| ● | ● | ● | ● | ● | ● | ● | ● |
| o | o | o | o | ● | ● | ● | ● |
| o | o | ● | ● | o | o | ● | ● |
| o | ● | o | ● | o | ● | o | ● |
| 8 | 9 | a | b | c | d | e | f |

HEXADECIMAL to DECIMAL

CONVERSION TABLE

Charles W. McClure

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | a | b | c | d | e | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 01 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 02 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 |
| 03 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 |
| 04 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 |
| 05 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 | 91 | 92 | 93 | 94 | 95 |
| 06 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 |
| 07 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 |
| 08 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 |
| 09 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 |
| 0a | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 |
| 0b | 176 | 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 |
| 0c | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 |
| 0d | 208 | 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 |
| 0e | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 |
| 0f | 240 | 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 |
| 10 | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 |
| 11 | 272 | 273 | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 |
| 12 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 |
| 13 | 304 | 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 |
| 14 | 320 | 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | 335 |
| 15 | 336 | 337 | 338 | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 |
| 16 | 352 | 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 | 364 | 365 | 366 | 367 |
| 17 | 368 | 369 | 370 | 371 | 372 | 373 | 374 | 375 | 376 | 377 | 378 | 379 | 380 | 381 | 382 | 383 |
| 18 | 384 | 385 | 386 | 387 | 388 | 389 | 390 | 391 | 392 | 393 | 394 | 395 | 396 | 397 | 398 | 399 |
| 19 | 400 | 401 | 402 | 403 | 404 | 405 | 406 | 407 | 408 | 409 | 410 | 411 | 412 | 413 | 414 | 415 |
| 1a | 416 | 417 | 418 | 419 | 420 | 421 | 422 | 423 | 424 | 425 | 426 | 427 | 428 | 429 | 430 | 431 |
| 1b | 432 | 433 | 434 | 435 | 436 | 437 | 438 | 439 | 440 | 441 | 442 | 443 | 444 | 445 | 446 | 447 |
| 1c | 448 | 449 | 450 | 451 | 452 | 453 | 454 | 455 | 456 | 457 | 458 | 459 | 460 | 461 | 462 | 463 |
| 1d | 464 | 465 | 466 | 467 | 468 | 469 | 470 | 471 | 472 | 473 | 474 | 475 | 476 | 477 | 478 | 479 |
| 1e | 480 | 481 | 482 | 483 | 484 | 485 | 486 | 487 | 488 | 489 | 490 | 491 | 492 | 493 | 494 | 495 |
| 1f | 496 | 497 | 498 | 499 | 500 | 501 | 502 | 503 | 504 | 505 | 506 | 507 | 508 | 509 | 510 | 511 |

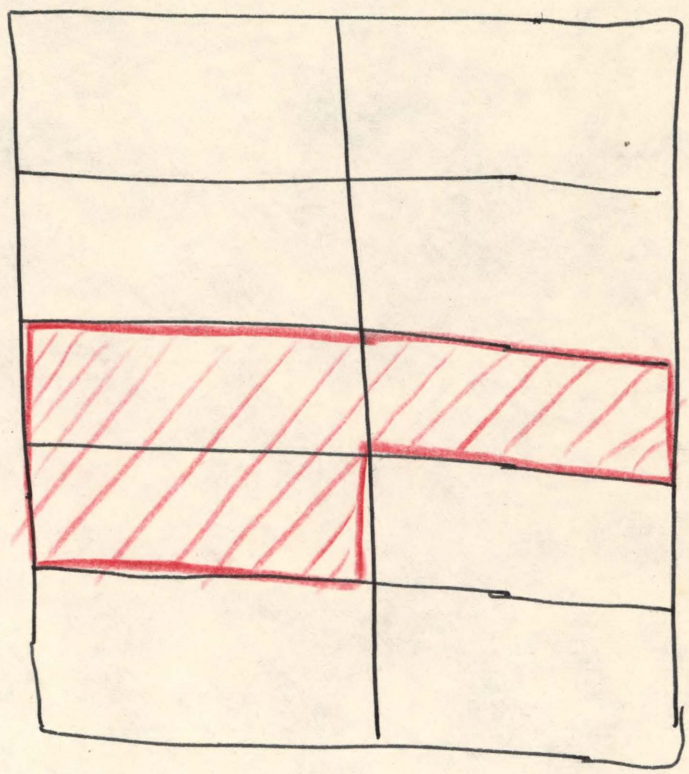
000-1ff

## CLEARING INPUT and OUTPUT

The "CLEAR EMPTY" push buttons on both the input and the output consoles must be depressed quite firmly to cause proper operation.

It is suggested that a check be made on the proper operation of the "CLEAR EMPTY" push button on the output console (the procedure is the same for the input console) whenever it is used.

Open the console door and check that all the neons on the chassis indicated in the sketch below are out. If any neon is on, depress "CLEAR EMPTY" again. If it is impossible to clear all the neons, call the engineer in charge.



$$e_o = E\left(1 - e^{-t/RC}\right)$$

t = .1 (×2) 10

RC = .1, .25, 1, 2.5, 10

PAT CODING FOR EE 183 Exp 1 Eq 5

BY ______ DATE 9/24/60 PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | | | | N | X | T | A | B | R |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | DECIMAL | | | | |
| 0.01 | LOAD | Program | | | | | — | 0 | 001 | 013 | 001 |
| 1.01 | IN | Constants, data | | | | 1 | 001 | 4 | 021 | — | 026 |
| 2.01 | IN | RC | | | | 2 | 001 | 4 | 032 | — | 032 |
| .02 | ADD | to | + 0 | → t | | 3 | — | 8 | 024 | 021 | 031 |
| .03 | SUB | 0 | −RC | → −RC | | 4 | — | a | 021 | 032 | 034 |
| .04 | CHK | | | | | 5 | — | 4 | — | — | — |
| 3.01 | DIV | t | /−RC | | | 6 | — | 3 | 031 | 034 | 035 |
| .02 | EXP E | " | | | | 7 | 008 | 4 | 035 | — | 035 |
| .03 | SUB | 1 | − " | | | 8 | — | a | 022 | 035 | 035 |
| .04 | MPY | E | x " | → $e_o$ | | 9 | — | 2 | 023 | 800 | 033 |
| (set FC = 3) 4.01 | OUT | t, RC, $e_o$ | | | | 10 | 002 | 4 | 031 | — | 033 |
| .02 | MPY | $t_i$ | x $R_t$ | $t_{i+1}$ | | 11 | — | 2 | 031 | 025 | 031 |
| 5.01 | TNE | t (") | −$t_f$ | : | 3.01 | 12 | — | e | 800 | 026 | 006 |
| .02 | TRA | | | : | 2.01 | 13 | — | 5 | — | — | 002 |
| | end, | | | | | | end, | | | | |
| | | 0 | | | | 21 | — | 0 | — | — | — |
| | | 1 | | | | 22 | — | 1 | — | — | — |
| | | E = 7 | | | | 23 | — | 7 | — | — | — |
| | | $t_o$ = .1 | | | | 24 | 101 | 1 | — | — | — |
| | | $R_t$ = 2 | | | | 25 | — | 2 | — | — | — |
| | | $t_f$ = 10 | | | | 26 | 001 | 1 | — | — | — |
| | | ___t | | | | 31 | | | | | |
| | | ___RC | | | | | 101 | 1 | — | — | — |
| | | | | | | | 101 | 2 | 500 | — | — |
| | | | | | | 32 | — | 1 | — | — | — |
| | | | | | | | — | 2 | 500 | — | — |
| | | | | | | | 001 | 1 | — | — | — |
| | | ___$e_o$ | | | | 33 | | | | | |
| | | ___OPSTO : −RC | | | | 34 | | | | | |
| | | ___OPSTO : t/−RC ; $e^{exp}$ ; $1 - e^{exp}$ | | | | 35 | | | | | |

EE 183 Exp 1 Eq (5)

Answers

| t | RC | $e_o$ |
|---|---|---|
| 1001000000000, | 1011000000000, | 0004424843811, |
| 1012000000000, | 1011000000000, | 0006052652743, |
| 1014000000000, | 1011000000000, | 0006871790514, |
| 1018000000000, | 1011000000000, | 0006997651761, |
| 0001600000000, | 1011000000000, | 0006999999212, |
| 0003200000000, | 1011000000000, | 0007000000000, |
| 0006400000000, | 1011000000000, | 0007000000000, |
| 1011000000000, | 1012500000000, | 0002307758601, |
| 1012000000000, | 1012500000000, | 0003854696917, |
| 1014000000000, | 1012500000000, | 0005586724218, |
| 1018000000000, | 1012500000000, | 0006714664470, |
| 0001600000000, | 1012500000000, | 0006988369099, |
| 0003200000000, | 1012500000000, | 0006999980674, |
| 0006400000000, | 1012500000000, | 0007000000000, |
| 1011000000000, | 0001000000000, | 1016661375880, |
| 1012000000000, | 0001000000000, | 0001268884299, |
| 1014000000000, | 0001000000000, | 0002307758601, |
| 1018000000000, | 0001000000000, | 0003854696917, |
| 0001600000000, | 0001000000000, | 0005586724218, |
| 0003200000000, | 0001000000000, | 0006714664470, |
| 0006400000000, | 0001000000000, | 0006988369099, |
| 1011000000000, | 0002500000000, | 1012744739203, |
| 1012000000000, | 0002500000000, | 1015381854102, |
| 1014000000000, | 0002500000000, | 0001034993356, |
| 1018000000000, | 0002500000000, | 0001916956478, |
| 0001600000000, | 0002500000000, | 0003308951569, |
| 0003200000000, | 0002500000000, | 0005053738493, |
| 0006400000000, | 0002500000000, | 0006458866792, |
| 1011000000000, | 0011000000000, | 1026965116087, |
| 1012000000000, | 0011000000000, | 1011386092418, |
| 1014000000000, | 0011000000000, | 1012744739203, |
| 1018000000000, | 0011000000000, | 1015381854102, |
| 0001600000000, | 0011000000000, | 0001034993356, |
| 0003200000000, | 0011000000000, | 0001916956478, |
| 0006400000000, | 0011000000000, | 0003308951569, |

183-1-5b
(WRONG)

PAT CODING FOR EE 183 - Exp 1 - Eq 5

BY CHD DATE 9/13/60 PAGE 1 OF 1

| FLOW # | TYPE | COMMENTS | N | X | T | A | B | R |
|---|---|---|---|---|---|---|---|---|
| | LOAD | Program | | — | 0 | 001 | 014 | 001 |
| | IN | Constants, Data | 001 | 001 | 4 | 195 | — | 200 |
| | IN | RC | 002 | 001 | 4 | 201 | — | 201 |
| | OUT | RC | 003 | 002 | 4 | 201 | — | 201 |
| | ADD | $t_0$ +0 → t | 004 | — | 8 | 198 | 195 | 202 |
| | SUB | 0 −RC → −RC | 005 | — | 9 | 195 | 201 | 201 |
| | HTR | *+1 | 006 | — | 6 | — | — | 007 |
| push "CLEAR EMPTY" on output, then "RUN" | CHK | | 007 | — | 4 | — | — | — |
| | DIV | t /−RC | 008 | — | 3 | 202 | 201 | 000 |
| | EXP E | " | 009 | 008 | 4 | 000 | — | 000 |
| | SUB | 1 − " | 010 | — | 9 | 196 | 800 | 000 |
| | MPY | E * " → $e_0$ | 011 | — | 2 | 197 | 000 | 203 |
| FC=2 | OUT | t, $e_0$ | 012 | 002 | 4 | 202 | — | 203 |
| | MPY | $t_i$ * $R_t$ → $t_{i+1}$ | 013 | — | 2 | 202 | 199 | 202 |
| | TNE | " − $t_f$ : | 014 | — | e | 800 | 200 | 007 |
| | TRA | | 015 | — | 5 | — | — | 002 |
| | | | | end | | | | |
| | | 0 | 195 | — | 0 | — | — | — |
| | | 1 | 196 | — | 1 | — | — | — |
| | | E = 7 | 197 | — | 7 | — | — | — |
| | | $t_0$ = .1 | 198 | 101 | 1 | — | — | — |
| | | $R_t$ = 2 | 199 | — | 2 | — | — | — |
| | | $t_f$ = 10 | 200 | 001 | 1 | — | — | — |
| | | RC { .1 | 201 { | 101 | 1 | — | — | — |
| | | .25 | | 101 | 2 | 500 | — | — |
| | | 1 | | — | 1 | — | — | — |
| | | 2.5 | | — | 2 | 500 | — | — |
| | | 10 | | 001 | 1 | — | — | — |
| | | ___ t | 202 | | | | | |
| | | ___ $e_0$ | 203 | | | | | |
| | | OPSTO: exp; $e^{-exp}$; $1-e^{-exp}$ | 000 | | | | | |
| | | | | | | | | |
| | | | | | | | | |

EE 183 Exp 1 Eq (5)

Second Version with Error Correction

```
0000001014001,
0014195000200,
0014201000201,
0024201000201,
0008198195202,
000a195201201,
0006000000007,
0004000000000,
0003202201000,
0084000000000,
000a196800000,
0002197000203,
0024202000203,
0002202199202,
000e800200007,
0005000000002,
end,
0000000000000,
0001000000000,
0007000000000,
1011000000000,
0002000000000,
0011000000000,
1011000000000,
1012500000000,
0001000000000,
0002500000000,
0011000000000,
```

```
0000001014001,
0014195000200,
0014201000201,
0024201000201,
0008198195202,
000a195201201,
0006000000007,
0004000000000,
0003202201000,
0084000000000,
000a196000000,
0002197000203,
0024202000203,
0002202199202,
000e800200007,
0005000000002,
end,
0000000000000,
0001000000000,
0007000000000,
1011000000000,
0002000000000,
0011000000000,
1011000000000,
1012500000000,
0001000000000,
0002500000000,
0011000000000,
```

```
0000001014001,
0014195000200,
0014201000201,
0024201000201,
0008198195202,
000a195201201,
0006000000007,
0004000000000,
0003202201000,
0084000000000,
000a196000000,
0002197800203,
0024202000203,
0002202199202,
000e800200007,
0005000000002,
end,
0000000000000,
0001000000000,
0007000000000,
1011000000000,
0002000000000,
0011000000000,
1011000000000,
1012500000000,
0001000000000,
0002500000000,
0011000000000,
```

```
0000001015001,
0014195000200,
0014201000201,
0024201000201,
0008198195202,
000a195201201,
0006000000007,
0004000000000,
0003202201000,
0084000000000,
000a196000000,
0002197800203,
0024202000203,
0002202199202,
000e800200007,
0005000000002,
end,
0000000000000,
0001000000000,
0007000000000,
1011000000000,
0002000000000,
0011000000000,
1011000000000,
1012500000000,
0001000000000,
0002500000000,
0011000000000,
```

First error punchout:

000008000000a, (10) 000a196800000,

Second error

0000080000011, 0002197000203,

Third

0000002000015, 0005000000002,

EE 183 Exp 1 Eq (5) b

**Answers**

9/17/60

| | |
|---|---|
| 1011000000000, | |
| 1011000000000, | 0004424843811, |
| 1012000000000, | 0006052652743, |
| 1014000000000, | 0006871790514, |
| 1018000000000, | 0006997651761, |
| 0001600000000, | 0006999999212, |
| 0003200000000, | 0007000000000, |
| 0006400000000, | 0007000000000, |
| 1012500000000, | |
| 1011000000000, | 0002307758601, |
| 1012000000000, | 0003854696917, |
| 1014000000000, | 0005586724218, |
| 1018000000000, | 0006714664470, |
| 0001600000000, | 0006988369099, |
| 0003200000000, | 0006999980674, |
| 0006400000000, | 0007000000000, |
| 0001000000000, | |
| 1011000000000, | 1016661375880, |
| 1012000000000, | 0001268884299, |
| 1014000000000, | 0002307758601, |
| 1018000000000, | 0003854696917, |
| 0001600000000, | 0005586724218, |
| 0003200000000, | 0006714664470, |
| 0006400000000, | 0006988369099, |
| 0002500000000, | |
| 1011000000000, | 1012744739203, |
| 1012000000000, | 1015381854102, |
| 1014000000000, | 0001034993356, |
| 1018000000000, | 0001916956478, |
| 0001600000000, | 0003308951569, |
| 0003200000000, | 0005053738493, |
| 0006400000000, | 0006458866792, |
| 0011000000000, | |
| 1011000000000, | 1026965116087, |
| 1012000000000, | 1011386092418, |
| 1014000000000, | 1012744739203, |
| 1018000000000, | 1015381854102, |
| 0001600000000, | 0001034993356, |
| 0003200000000, | 0001916956478, |
| 0006400000000, | 0003308951569, |

PAT including the new ERROR format CWMc 10-12-60

000018031f3e8,

| | | | |
|---|---|---|---|
| 0006000000181, | 0044fa47a78f8, | 10083ff3ff35e, | 10081873ff35a, |
| 10081883ff35b, | 10081863ff35c, | 000500000035a, | 100818a3ff340, |
| 100835a18935a, | 0000001000001, | 019135d19c344, | 100835d3fb800, |
| 019180001c34c, | 00d135d01c348, | 10080003ff35f, | 019180019a347, |
| 00d135f01a34a, | 10082383ff35b, | 000500000020d, | 100835f3ff000, |
| 100835c3ff000, | 100601000034c, | 0005000000000, | 101135dc10fff, |
| 10080003ff35d, | 2005000000340, | 100834e3ff180, | 12c834f3ff181, |
| 101135dc10000, | 00050000001df, | 10050800001a6, | 100a3f9355000, |
| 10083553fd355, | 100e8001a4355, | 00050000001a6, | 0000000800000, |
| 100a3f9355180, | 0005000000354, | 000035f35f1a7, | 100835f3ff35d, |
| 02d135f296350, | 000e3503511e6, | 100e34d352314, | 10083eb3ff35e, |
| 125135f254355, | 119135f0dc356, | 119135f0d435a, | 11d135f12435b, |
| 121135f17435c, | 100c3561a31ba, | 100e23635a314, | 100e23735b314, |
| 100235c196800, | 100880035b800, | 1002800196800, | 100880035a356, |
| 100e237800314, | 00d135619c35d, | 100835e35e35e, | 10d135f0dc357, |
| 10d135f0d435a, | 111135f12435b, | 115135f17435c, | 100c3571a31c8, |
| 100e23635a314, | 100e23735b314, | 100235c196800, | 100880035b800, |
| 1002800196800, | 100880035a357, | 100e237800314, | 00d13570dc35d, |
| 100835e35e35e, | 101135f0dc358, | 101135f0d435a, | 105135f12435b, |
| 109135f17435c, | 100e23635a314, | 100e23735b314, | 100235c196800, |
| 100880035b800, | 1002800196800, | 100880035a358, | 100e237800314, |
| 00d135801c35d, | 100e34f3ff1e9, | 11911de19c35e, | 100e35531e1f2, |
| 100e1dd3551e2, | 10113ee20135e, | 100c356359314, | 100d357359314, |
| 100e1db3551fd, | 100b3583ff359, | 00057cd000352, | 10083fb3fb35e, |
| 10083523fd352, | 00050000001a6, | 100e1db3551d9, | 0005000000314, |
| 100835e3fb35e, | 0005000000233, | 100e34d3521e8, | 0005000000314, |
| 000600000034c, | 10083fb3ff35e, | 100e1a6355314, | 100e357356314, |
| 100e237357314, | 100934f3ff34f, | 00d135601c352, | 00d135701c34d, |
| 00d135801c34c, | 00050000001a6, | 100e3553f91f6, | 100c3553f91ff, |
| 100e3551e81fd, | 0005000000314, | 100e1f03551d9, | 10023fa3fa35e, |
| 10050400001fa, | 0005000000314, | 100c3fa355221, | 100e357356314, |
| 100e237357314, | 10081a33ff359, | 0005000000352, | 10023fa3f935e, |
| 129135f07835b, | 02b135f193347, | 12a135f01135a, | 129135f01135c, |
| 100e35b3fc347, | 0005000000314, | 125135b0dc35b, | 0005000000210, |
| 100c35c3ff206, | 119135b0dc35b, | 0005000000210, | 10d135b0dc35b, |
| 0005000000210, | 100c35a3ff208, | 100c35c3ff20b, | 101135b0dc35b, |
| 100c35b3ff314, | 00d135b0dc34f, | 10d121a19c35e, | 001135d0da35d, |
| 001135219c34f, | 025121a25435d, | 100a3503f3350, | 100a3513f3351, |
| 001180019c34e, | 001135001c35d, | 100502040121c, | 100e350220314, |
| 119121f19c35e, | 100e237356314, | 10081a33ff359, | 0005480000350, |
| 100834e3ff12b, | 100835e3fb35e, | 100e357240224, | 100a240357357, |
| 129135f0b835b, | 029135f23135d, | 02a135f17135d, | 100f240357314, |
| 00d13572973 4f, | 032135730134f, | 10083ff35d35d, | 100835e3fb35e, |
| 100e3fc35b314, | 100c3ff35b1e4, | 100c3fc35b231, | 00d12300dc34f, |
| 00050002e6212, | 00d12320dc34f, | 00050002fc212, | 100e240356314, |
| 00d12350dc34f, | 00050002da212, | 0000000009000, | 000000012b000, |
| 02d82d12d0000, | 0000289288242, | 02a8262000000, | 0000000000000, |
| 02c2000000000, | 0000285287241, | 02a7261000000, | 0000000000001, |
| 0000000064000, | 00113ee321340, | 100e35b3fd25f, | 10113ee2c1355, |
| 029135b218355, | 031135b321355, | 00083553ff355, | 000f3ea80025d, |
| 000a355355355, | 000a35b3fe356, | 00088003ea800, | 0003356800356, |
| 00088003ff357, | 00083fe3ff358, | 0002356356359, | 0002357358357, |

| | | | |
|---|---|---|---|
| 00083583ea358, | 0003359800800, | 0002357800357, | 0008356800356, |
| 0002800312800, | 000f80035724f, | 00023563ea356, | 000235530e800, |
| 000835680035c, | 100e3ff34034a, | 000335c30d35c, | 10093403ff340, |
| 00050000034a, | 10083ff35b35b, | 0005000000249, | 119126019d35e, |
| 0005083000313, | 00113ee321340, | 000935b3ff800, | 000a8003fe355, |
| 00088003ea800, | 0003355800355, | 0002800800356, | 000280027d800, |
| 000880027e800, | 0002800356800, | 000880027f800, | 0002800356800, |
| 0008800280800, | 0002800356800, | 0008800281800, | 0002800356800, |
| 0008800282800, | 0002800356800, | 0008800283800, | 0002800356800, |
| 0008800284800, | 0002800355800, | 000880027c35c, | 032135b32135c, |
| 100e3ff34034a, | 000335c30c35c, | 10093403ff340, | 00050000034a, |
| 000c90fdaa222, | 30784d7e74355, | 105b31651d7e0, | 304e501c3c528, |
| 103c577df40b3, | 3028e6c62d66c, | 102cc40a5b5d1, | 301aaa61be24c, |
| 000ffff4ced2, | 000235b30c800, | 0005000000289, | 000235b30c800, |
| 000a30b800800, | 00038003f2355, | 000f80031028d, | 119128c19d35e, |
| 0005081000313, | 00083fe3ff356, | 000e3ff355291, | 000a3ff355355, |
| 000a3ff356356, | 0008310355800, | 000a800310800, | 000a355800355, |
| 0005000000297, | 000a3ff356356, | 000a3553fe355, | 000e3f8800295, |
| 00023553f2357, | 0002800800800, | 000a3ff80035a, | 00083fe3ff359, |
| 00083573ff358, | 00083593ea359, | 000a8003fe800, | 0003358800800, |
| 000280035a800, | 0003800359358, | 0008357800357, | 0002800312800, |
| 000f80035829d, | 0002357356 35c, | 00050000034a, | 000235b30d35b, |
| 000380030e355, | 000f30f8002bd, | 00083f83ff356, | 0009355311357, |
| 0011800298356, | 000a357311800, | 000280030e358, | 000b35b800357, |
| 00083ff3ff358, | 00083fe3ff35c, | 00083fe3ff359, | 00083583fe358, |
| 0003357800800, | 0002800359359, | 000835c80035c, | 0002800312800, |
| 000f8003592b3, | 000235635c35c, | 000e3ff35b34a, | 00033fe35c35c, |
| 00050000034a, | 000e3ff35b2c0, | 00083ff3ff35c, | 00050000034a, |
| 11912c119d35e, | 0005082000313, | 000935b3ff35e, | 000a35c35c35c, |
| 000c35b3ff34a, | 00083ee3ff35c, | 02a135b29735c, | 031135b31135c, |
| 000335e35c800, | 000a80035c800, | 00028003f835a, | 000880035c35c, |
| 0002800312800, | 000f80035a2c8, | 032135b32135c, | 00050000034a, |
| 100b3ff35f35f, | 119135f0dc355, | 10d135f0dc356, | 00d12d501c35f, |
| 100e3563552d6, | 000500034b360, | 11912d719d35e, | 0005001000313, |
| 10082d93ff35f, | 000500034b3d3, | 100835f3ff35c, | 019135d01c34a, |
| 129135f0d8355, | 119135f0d8356, | 10083553fc355, | 10088003fc800, |
| 100e3568002e4, | 00d135529835c, | 00d135f01c34c, | 00050000034a, |
| 10083ff35f35c, | 00050000034a, | 100835b3ff35c, | 019135d01c349, |
| 129135f0d8355, | 129135d0d7356, | 030135d321356, | 119135f0da357, |
| 10d135f0da358, | 100e2403552fa, | 10083553fc800, | 00d180029835f, |
| 100e2373572fa, | 100e2373582fa, | 123135f011357, | 117135f011358, |
| 100c3573ff2f7, | 10033563eb800, | 100835f80035f, | 100c3583ff349, |
| 100835f35635f, | 0005000000349, | 10d12fb19d35e, | 0005001004313, |
| 10080003ff35c, | 019134701c349, | 129135b1b8355, | 10083ff35b35f, |
| 10088003fd800, | 001180019c347, | 00d130301c348, | 0005000304347, |
| 129135b1b7356, | 030135b321356, | 1002355356800, | 100a3ff800356, |
| 00050000002f2, | 2005000000180, | 0000000000000, | 001c90fdaa222, |
| 1058efa351296, | 002935d8dddab, | 000b17217f7d2, | 008feffffffff, |
| 0298000000000, | 0288000000001, | 1108000000000, | 019134401c352, |
| 119135e25d35e, | 101135209a354, | 1003354240800, | 10088003fc354, |
| 025180009435e, | 00113ff254354, | 1002354196354, | 020180005435e, |
| 00113ff600354, | 1002354196800, | 01b180001435e, | 000735e35f309, |

D. Libraries .................................................. 60
1. Use
2. Form
3. Adaptation
a. Types of Addresses
(1) Non-essential
(2) Fixed
(3) Essential
b. How to Accomplish Adaptation
(1) Manual
(2) Automatic
(3) Semi-automatic
c. The Adaptation Routine

III. TESTING

A. Preventive Maintenance ........................................ 77

B. Test Routine .................................................. 77
1. What is Testing
2. Automatic Testing
3. Nature of a Test Routine
a. Complications
b. Simplifications
(1) Arithmetic Control Voltages
(2) Transfer
(3) Short Memory Addresses
c. Accomplishments
d. Inadequacies
(1) The Halt Order
(2) Memory
(3) Numbers Used
(a) Generation of Zero
(b) Too Small an Exponent
(c) Determination of the Larger Exponent
(d) Too Great a Difference in Exponents
e. Value

C. Trouble Shooting Routine ...................................... 83

D. One Cycle Operation ........................................... 85

E. Checking ...................................................... 85

IV. SUMMARY

A. Standard Operating Procedure .................................. 87

B. Conclusions ................................................... 89
1. Programming
2. Maintenance
3. Applications
4. Operating Rules

APPENDIX 1: Mechanized Computation--How the WISC Works ...... 92
APPENDIX 2: Binary Arithmetic ................................ 96

REFERENCES ....................................................... 103