MSUCP-4

MICHIGAN STATE UNIVERSITY

CYCLOTRON PROJECT*

Computer Program for Tracking Of Linear Cyclotron Orbits

D. A. Johnson and H. G. Blosser

October 1960

Department of Physics

East Lansing, Michigan

*Research Supported in part by U. S. Atomic Energy Commission Contract AT(11-1)-872

ABSTRACT

This report describes a computer routine for the fast tracking of orbits in the intermediate radius region of sector focused cyclotrons. Input to the routine consists of a table of equilibrium orbit properties including linear oscillation results. The routine interpolates in the table to obtain linear transformation matrices at arbitrary energies and tracks the particle motion by successively applying the appropriate transformations. The routine is enormously faster than direct integration routines, the exact factor varying with the amount of output requested. Accuracy with respect to spatial coordinates directly reflects the degree of linearity of the fields; 1 in 10^4 agreement with orbit integrations is common in the intermediate radius region. The computation of the particle phase is usually less accurate but quite adequate in most circumstances.

CODE DESCRIPTION

Particle orbits are traced using simplified equations of motion based on the assumption that radial betatron oscillation is described by a linear, second order differential equation of the form:

$$\frac{d^2x}{d\theta^2} + (\nu^2 + f(\theta))x = 0 \qquad (1)$$

In addition, the routine assumes the orbital frequency of the particle to be independent of betatron amplitude.

Notation used is defined as follows: (1) Let a subscript n on any quantity refer to the nth fractional turn, where a fractional turn is the portion of a turn extending from one gap crossing to the next and including crossing of the accelerating gap at the end of the turn. (2) Let M represent the number of accelerating gaps. (3) Let a superscript k on any quantity indicate that the quantity is evaluated at the azimuth of the kth accelerating gap. Since a fractional turn includes only one accelerating gap, n equals k(mod M). (4) Let r_n^K and Pr_n^K respectively represent the radius value at which the particle is located and the radial component of its momentum at the end of the nth fractional turn. (5) Let E_n and ϕ_n denote respectively the energy of the particle after the nth fractional turn and the phase of the particle at the gap with respect to the r.f. accelerating voltage. (6) Let r_e^K, Pr_e^K, ϕ_e^K, a^K, b^K, c^K, & d^K all denote functions of E, the particular values of the functions being determined from equilibrium orbit code output in a manner to be described subsequently. (7) Define x_n^K and Px_n^K as follows:

$$x_n^K = r_n^K - r_e^K(E_n) \qquad Px_n^K = Pr_n^K - Pr_e^K(E_n)$$

With the above notations, initial conditions for a particle are specified as E_0, x_0^0, Px_0^0, and $\frac{1}{4}\phi_0^0$. The particle motion is mapped through successive fractional turns by these equations:

$$\frac{1}{4}\phi_{n+1}^{k+1} = \frac{1}{4}\phi_n^k + \frac{\pi}{2}\left(\phi_e^{k+1} - \phi_e^k\right) \qquad (2)$$

$$E_{n+1}^{k+1} = E_n^k + \frac{\Delta E}{M}\cos\phi_{n+1}^{k+1} \qquad (3)$$

$$x_{n+1}^{k+1} = a^k(E_n)\cdot x_n^k + b^k(E_n)\cdot px_n^k + r_e^{k+1}(E_n) - r_e^{k+1}(E_{n+1}) \qquad (4)$$

$$px_{n+1}^{k+1} = c^k(E_n)\cdot x_n^k + d^k(E_n)\cdot px_n^k + pr_e^{k+1}(E_n) - pr_e^{k+1}(E_{n+1}) \qquad (5)$$

The process continues until the energy exceeds a preset maximum, E_{max} which is an input parameter. Output from the code consists of an identification word (also an input parameter) followed by lines of information giving the particle conditions on the $0th$ and every Lth subsequent fractional turn, arranged in this pattern:

I.D. word

0	E_0	x_0^0	px_0^0	r_0^0	pr_0^0	$\frac{1}{4}\phi_0^0$
L	E_L	x_L^k	px_L^k	r_L^k	pr_L^k	$\frac{1}{4}\phi_L^k$
2L	E_{2L} . . .					
⋮						

Stop Symbol

The 0, L, 2L, . . . are output as three digit integers; all other quantities are output as a sign and six decimal digits.

Input parameters for the program are as follows:

E_0 - initial particle energy

x_0^0 - initial particle displacement from the equilibrium orbit

px_0^0 " " " " " " "

$\frac{1}{4}\phi_0^0$ - initial particle phase with respect to the r.f. voltage

M - number of accelerating gaps

L - number of fractional turns between print-outs

ΔE - energy gain per turn if $\cos\phi$ equals <u>one</u> at all gaps

E_{max} stopping point for code

J - number of energy values at which equilibrium orbits are given

When the " $r\,P_r$ overwrite tape" is used with the code, r_0^o and Pr_0^o are used as input parameters instead of x_0^o and Px_0^o.

Input data for the routine consists of output from the revised equilibrium orbit code, with information in this format:

10 digit I.D. plus 37 non-fifth-hole characters (headings)

$$
\begin{array}{llllllll}
E_j & & & & & & & \\
\theta_0 & r_e^0(E_j) & Pr_e^0(E_j) & x_1^0(E_j) & Px_1^0(E_j) & x_2^0(E_j) & Px_2^0(E_j) & \phi_e^0(E_j) \\
\theta_1 & r_e^1(E_j) & Pr_e^1(E_j) & x_1^1(E_j) & \cdots & & & \\
\vdots & & & & & & & \\
\theta_M. & r_e^M(E_j) & Pr_e^M(E_j) & x_1^M(E_j) & \cdots & & &
\end{array}
$$

[Subscripts 1 & 2 on x & Px values are for identification only, and do not represent values of n.]

All numbers are given as signed decimal fractions except θ, which is given as two sexadecimal digits. θ represents the number of the angular step for which the set of values is given, the size of the steps being $\frac{2\pi}{48}$. $\theta_M - \theta_0$ is always equal to forty-eight, meaning that the set of data represents one full revolution. Since the equilibrium orbits close, it follows that the r and P_r values at θ_M are equal to the corresponding values at θ_0. The $x_{1,2}$ and $Px_{1,2}$ values, however, represent sample displaced orbits which do not necessarily close, and thus are not necessarily the same at θ_M and θ_0.

The transformation matrix $A^k(E_j)$ is determined such as to map the two linearly independent vectors (x_1^k, Px_1^k) and (x_2^k, Px_2^k) into (x_1^{k+1}, Px_1^{k+1}) and (x_2^{k+1}, Px_2^{k+1}), i.e.:

$$
A^k(E_j) = \begin{pmatrix} a^k(E_j) & b^k(E_j) \\ c^k(E_j) & c^k(E_j) \end{pmatrix} \text{ must give } \begin{pmatrix} x_{1,2}^{k+1} \\ Px_{1,2}^{k+1} \end{pmatrix} = (A^k)\begin{pmatrix} x_{1,2}^{k} \\ Px_{1,2}^{k} \end{pmatrix}
$$

Which, solved for a, b, c, & d, yields the following:

$$
a^k(E_j) = \frac{x_1^{k+1}\cdot Px_2^k - x_2^{k+1}\cdot Px_1^k}{x_1^k\cdot Px_2^k - x_2^k\cdot Px_1^k} \qquad b^k(E_j) = \frac{x_1^{k+1}\cdot x_2^k - x_1^k\cdot x_2^{k+1}}{x_2^k\cdot Px_1^k - x_1^k\cdot Px_2^k}
$$

$$
c^k(E_j) = \frac{Px_1^{k+1}\cdot Px_2^k - Px_1^k\cdot Px_2^{k+1}}{x_1^k\cdot Px_2^k - x_2^k\cdot Px_1^k} \qquad d^k(E_j) = \frac{x_2^k\cdot Px_1^{k+1} - x_1^k\cdot Px_2^{k+1}}{x_2^k\cdot Px_1^k - x_1^k\cdot Px_2^k}
$$

The quantities $a^K(E_j)$, $b^K(E_j)$, . . . are computed and stored during interludes in the data input routine, as are the $(\phi_e^{K+1} - \phi_e^K)$ values. The data as stored is then in this form:

$$E_j\ ,\ r_e^0(E_j),\ Pr_e^0(E_j),\ a_e^0(E_j),\ b^0(E_j),\ c^0(E_j),\ d^0(E_j),\ (\phi_e^1 - \phi_e^0),$$

$$\vdots$$

$$r_e^{M-1}(E_j),\ Pr_e^{M-1}(E_j),\ a^{M-1}(E_j),\ b^{M-1}(E_j),\ c^{M-1}(E_j),\ d^{M-1}(E_j),\ (\phi_e^M - \phi_e^{M-1});$$

$$E_{j+1},\ r_e^0(E_{j+1})\ .\ .\ .$$

To obtain the values of these functions of E at arbitrary values of E, the routine uses four-point interpolation in the stored table of numbers. Since the E_j values are equally spaced in the equilibrium orbit code output, equal interval polynomial interpolation is suitable for the process.

The Linear Motion Code is written in two parts, which are run separately. The first part is the data input routine; it is loaded into the memory with a bootstrap input and includes the DOI on the front of the tape. This part of the code reads in the equilibrium orbit data, calculates $a^K(E_j)$, $b^K(E_j)$, . . . during interludes, and stores the data sets in the sequence outlined above. The identification word from the equilibrium orbit data is also stored. Three parameters are needed for Part 1: the number of the first memory location for data storage, the number of accelerating gaps used, and the number of energy values for which information is given. Because the stored data must meet certain requirements for the interpolation subroutine used in Part 2, Part 1 tests each value and stops if the requirements are not met. The stop order appearing on the MISTIC control board identifies the difficulty:

KKK01 - $|r_e^K|$, $|Pr_e^K|$, $|x_1^K|$, $|Px_1^K|$, $|x_2^K|$, $|Px_2^K|$ or $|\phi_e^K| \geq \frac{1}{2}$

KKK02 - $|a^K|$, $|b^K|$, $|c^K|$ or $|d^K| \geq \frac{1}{2}$

KKK03 - $|E_j| \geq \frac{1}{2}$

KKK04 - $E_{j+1} - E_j \neq E_1 - E_0$

KKK05 - $|\phi_e^{K+1} - \phi_e^{K}| \geq \frac{1}{2}$

Part 2 is read in with the DOI, overwriting Part 1 and occupying memory locations 4 through 310. At this point, it is often desirable to use the X-16S sexadecimal dump routine to give a single program tape for later runs with the particular fields used. The DOI must be added to the end of the hex tape, as it is needed for parameter input. Whether the single hex tape or the separate DOI format tapes are used, after the equilibrium orbit data and Part 2 are in the memory, the only further step is to supply the remaining parameters and transfer control to the program. The parameters for Part 2 consist of: initial particle conditions; the maximum energy desired; the number of fractional turns between print-outs; a ten-digit identification word; and finally, the maximum energy gain per revolution. Part 2 tracks the particle until one of a set of tests fails, then prints out an indication of the reason for the stop:

E MAX - particle energy exceeds

PHASE - particle is more than 90° out of phase with the r.f. voltage

X PX - the value of $[(x)^2 + (p_x)^2]$ exceeds 2^{-8}

E OUT - equilibrium orbit data not available at particle energy

After printing one of the above, a sum check is carried out on the main program and the stored data. If these change after the first run due to memory failure, the computer will stop on an SF order. Otherwise, control is transferred to the DOI with a black switch stop. Thus, if several parameter tapes are made in one piece, and the black switch is set to "ignore" when they are started, a series of Linear Motion Code runs can be made with no further attention.

Normal running procedure:

1) Bootstrap LMC Tape 1
2) Black switch 1st parameter tape
3) Black switch E-O data tape
4) Black switch LMC Tape 2
5) Black switch 2nd parameter tape
   Code runs, stops on 243F7

Making hex tape:

1) Bootstrap LMC Tape 1
2) Black switch 1st parameter tape
3) Black switch E-O data tape
4) Black switch LMC Tape 2
5) Bootstrap X-16S subroutine, white switch past stop in tape
6) Put console DOI on end of resulting hex tape

Running with hex tape:

1) Bootstrap LMC hex tape
2) Black switch 2nd parameter tape
   Code runs, stops on 243F7

Running with r Pr overwrite:

Black switch r Pr overwrite after LMC Tape 2 or LMC hex tape

Running with constant acceleration:

Add the following order pair and directive to the 2nd parameter tape:

00 198K
L5 10S9 10 1F

1st parameter tape (DOI format)

00 16K
E1 (integer)
M (integer) } chosen such that
J (integer) } $J(1+7M) \leq 688$
24 44N

2nd parameter tape (DOI format).

00 19K
E_0 (fraction, scaled by 10^{-3})
x_0 (fraction)
Px_0 (fraction)
$\frac{1}{4}\phi_0$ (fraction)
E_{max} (fraction, scaled by 10^{-3})
L (integer)
I.D. (10 digit integer)
ΔE (fraction, scaled by 10^{-3})
26 151N

2nd parameter tape for use with r Pr overwrite:

00 19K
E_0
00 22K
$\frac{1}{4}\phi_0$
E_{max}
L
I.D.
ΔE
r_0 (fraction)
Pr_0 (fraction)
26 151N

Parameters:

E1 - Number of 1st memory location for E-O data storage (Space required = $J(1+7M)$)
M - Number of accelerating gaps used
J - Number of energy values for which E-O data is given
E_0 - Initial particle energy
x_0 - Initial particle displacement from the equilibrium orbit
Px_0 - " " " " " " "
$\frac{1}{4}\phi_0$ - Initial particle phase with respect to the r.f. voltage
E_{max} - Maximum energy desired
L - Number of fractional turns between prints
I.D. - Ten digit identification integer
ΔE - Maximum energy gain per revolution
r_0 - Initial particle radius location
Pr_0 - Initial particle radial momentum

LOCATIONS OF PROGRAM PARTS

Part 1

16 - 28	Parameters (Parts 1 & 2)
29 - 33	Storage (Parts 1 & 2)
34 - 43	Storage & constants, etc. (Part 1)
44 - 148	LMC main program
149 - 170	N8 subroutine

Part 2

16 - 28	Parameters
29 - 33	Storage (some carried over from Part 1)
34 - 40	ISR subroutine storage
41 - 76	Interpolation Subroutine
77 - 135	Stop indications and sum check
136 - 150	Storage (Part 2) constants, etc.
151 - 257	LMC Part 2 main program
258 - 285	P1 subroutine
286 - 306	T5 subroutine
307 - 310	r Pr overwrite (if used)

Linear Motion Code

Part 1

```
00 4+
00 F 00 34F
00 F 00 41F
00 F 00 149F
00 F 00 29F
00 F 00 44F
00 F 00 16F
00 F 00 136F
00 F 00 77F
00 F 00 258F
00 F 00 151F
00 F 00 286F
```

} S-box locations

```
00 36+
40 2-7 L5 8-7
26 12-8 00 F
22 14-8 00 F
00 F 00 F
00 F 00 6F
00 F 00 1F
00 F 00 3F
00 F 00 7F
```

} 7S7 - 14S7

```
00 44+
```

Main routine (S8-104S8)

```
L5 -9 40 5-7
L5 7-7 40 10L
81 40F 40 3F
80 48F 80 48F
```

} [1]

```
80 52F L5 5-7
42 9L 50 5L
26 -6 -L F
36 8L ++ 2563F
F5 5-7 40 5-7
-5 F 40 F
```

} [2]

```
40 2-7 L5 8-7
40 10L 22 21L
```

} [3]

```
L0 2-7 40 4-7
L5 9-7 40 10L
-5 F 40 F
```

} [4]

```
L5 2-7 40 3-7
L5 F 40 2-7
L0 3-7 L0 4-7
40 F L7 F
10 4F 40 F
L3 F 32 21L
```

} [5]

3 - I.D. storage

S7 - 3S7 temporary storage

4S7 - $E_1 - E_0$

5S7 - counter A

6S7 - counter C

10S7 - counter for [8]

S9 - E1
1S9 - M
2S9 - J
} parameters

```
++ 2564F 41 6-7        [5] - [6]
80 8F L5 5-7           - [7]
42 26L 50 23L
26 -6 -L F
36 26L ++ 2561F        [8]
-5 F 40 F
F5 5-7 40 5-7
F5 10-7 40 10-7
L5 11-7 L0 10-7
32 22L 41 10-7
F5 6-7 40 6-7
L5 12-7 L0 6-7         [9]
36 22L L5 5-7
L0 12-7 42 92L
L0 12-7 42 66L
42 83L L0 12-7
42 57L 42 75L
L0 12-7 42 67L
42 85L L0 12-7
42 59L 42 76L          set addresses for
L0 13-7 42 93L         [10], [11], & [12]
42 96L L0 12-7
42 55L 42 60L
42 68L 42 89L
L0 12-7 42 52L
42 77L 42 84L
42 91L L0 12-7
42 53L 42 58L
42 65L 42 81L
L0 12-7 42 54L
42 74L 42 82L
42 90L 50 F
N0 F 7J F
40 -7 50 F
N0 F 7J F
L0 -7 40 -7
N0 F 50 F
N0 F 7J F
40 1-7 50 F
N0 F 7J F
L0 1-7 10 4F
66 -7 -L F             [10]
36 64L ++ 2562F
-5 F 40 F
N0 F 50 F
N0 F 7J F
40 1-7 50 F
N0 F 7J F
L0 1-7 10 4F
66 -7 -L F
```

```
[10]
36 72L ++ 2562F
-5 F 40 1F
L1 -7 40 -7
NØ F 50 F
NØ F 7J F
40 1-7 50 F
NØ F 7J F
LØ 1-7 10 4F
66 -7 -L F
36 81L ++ 2562F
-5 F 40 F                [11]
NØ F 50 F
NØ F 7J F
40 1-7 50 F
NØ F 7J F
LØ 1-7 10 4F
66 -7 -L F
36 89L ++ 2562F
-5 F 40 F
L5 F 40 F
L5 1F 40 F
NØ F L5 F
NØ F LØ F                [12]
40 F LL F
36 96L ++ 2565F
L5 F 40 F
L5 1-9 LØ 6-7            [13]
36 22L L5 5-7
LØ 14-7 40 5-7
50 14-7 75 1-9
75 2-9 -5 F              [14]
L4 2-9 L4 -9
LØ 5-7 LØ 12-7
32 4L 24 999F

ØØ 149+
```

N8 subroutine

```
24 999N
```

Linear Motion Code
Part 2

Interpolation Subroutine
(55-3555, 54-654)

entry:

to interpolate in $f(E_1)$, $f(E_2)$, $f(E_3)$, $f(E_4)$ given $\frac{E-E_2}{E_2-E_1}$ in 157 and location of E_3 equal to F;

p) 50 P 50 p
p+1) 26 55 –

```
    ØØ 41+
0)  41 F +5 F            - Link
1)  42 34L 1Ø 2ØF        } store P
2)  42 F 5Ø 6-+          }
3)  75 3-+ -5 F          }
4)  L4 F L4 2-+          }
5)  LØ 1-+ 42 25L        } set addresses
6)  L4 -+ 42 22L         }
7)  L4 -+ 42 2ØL         }
8)  L4 -+ 42 27L         }
9)  L5 1-7 4Ø -4
10) Ø9 1F L4 -4
11) 4Ø 2-4 1Ø 1F
12) 4Ø 3-4 49 4-4
13) L5 -4 1Ø 1F
14) 4Ø 5-4 L4 4-4
15) 4Ø 6-4 L5 3-4
16) LØ 4-4 5Ø -4
17) 4Ø 4-4 7J 2-4
18) 4Ø -4 5Ø -4
19) 7J 35L 4Ø -4
20) NØ F 5Ø F
21) 79 5-4 5Ø 3-4
22) 4Ø 3-4 7J F
23) L4 3-4 4Ø 3-4
24) 5Ø 6-4 7J 4-4
25) 4Ø 5-4 5Ø F
26) 7J 4-4 1Ø 1F
27) 4Ø 4-4 5Ø F
28) 7J 6-4 1Ø 1F
29) LØ 4-4 4Ø 6-4
30) 5Ø 3-4 7J 5-4
31) ØØ 2F 4Ø 4-4
32) 5Ø 6-4 7J -4
33) L4 4-4 4Ø 4-4
34) L5 4-4 22 F
35) 4Ø F ØØ 1666 6666 6667 J
```

$P = 0$ for r_e^K	K = counter H
$P = 1$ for Pr_e^K	location of $f(E_1) = F - 14M - 1 + 7H + P$
$P = 2$ for a^K	" " $f(E_2) = F - 7M + 7H + P$
$P = 3$ for b^K	" " $f(E_3) = F + 1 + 7H + P$
$P = 4$ for c^K	" " $f(E_4) = F + 7M + 2 + 7H + P$
$P = 5$ for d^K	
$P = 6$ for $(\phi_e^{K+1} - \phi_e^K)$	

Stop indication & sum check
(55 - 58 55)

```
    ØØ 77+
0)  92 131F 92 259F      }
1)  92 194F 92 963F      }
2)  92 643F 92 387F      } prints E MAX
3)  92 451F 92 7Ø7F      }
4)  92 147F 26 19L       }
5)  92 131F 92 259F      }
6)  92 2F 92 771F        }
```

7)	92 387F 92 706F	prints PHASE
8)	92 194F 92 707F	
9)	92 147F 26 19L	
10)	92 131F 92 259F	prints XPX
11)	92 451F 92 963F	
12)	92 2F 92 451F	
13)	92 707F 92 147F	
14)	26 19L 92 131F	
15)	92 259F 92 194F	prints EOUT
16)	92 963F 92 578F	
17)	92 450F 92 322F	
18)	92 707F 92 147F	
19)	41 1F L5 10F	set counter
20)	L4 58L 40 F	
21)	L5 F 42 23L	form sum
22)	L5 1F L4 F	
23)	40 1F F5 F	increase counter
24)	40 F L0 13F	
25)	L0 51L 40 2F	skip LMC 2 changing orders & addresses
26)	L3 2F 32 23L	
27)	L5 F L0 13F	
28)	L0 52L 40 2F	
29)	L3 2F 32 23L	
30)	L5 F L0 13F	
31)	L0 53L 40 2F	
32)	L3 2F 32 23L	
33)	L5 F L0 13F	
34)	L0 54L 40 2F	
35)	L3 2F 32 23L	
36)	L5 F L0 12F	skip P1 changing address
37)	L0 55L 40 2F	
38)	L3 2F 32 23L	
39)	L5 F L0 14F	skip T5 changing address
40)	L0 56L 40 2F	
41)	L3 2F 32 23L	
42)	L5 57L L0 F	test for end
43)	35 21L L5 1F	store sum (1st time)
44)	40 49L L5 50L	order change (1st time)
45)	40 43L L5 1F	test sum (2nd time on)
46)	L0 49L 40 1F	
47)	L3 1F 32 48L	
48)	F F 24 999F	stop, transfer to DOI if no SF
49)	00 F 00 F	
50)	36 21L 22 45L	
51)	00 F 00 17F	Constants, addresses of changing orders
52)	00 F 00 23F	
53)	00 F 00 30F	
54)	00 F 00 36F	
55)	00 F 00 25F	
56)	00 F 00 12F	

```
57) 00 F 00 998F
58) 00 F 00 6F

    00 142+

    00 F 00 7F                          }
    00 F 00 1F                          }
    40 1-7 N0 F                         }
    40 12-9 L5 10-+                     }
    40 1-7 26 51-J                      }  6SK - 14SK
    40 12-9 26 43-J                     }
    00 F 00 3926 9908 1700 J            }
    00 F 00 0039 0625 0000 J            }
    00 F 00 3183 0988 6180 J            }

    00 151+

0)  50 1-9 75 6-+       }
1)  -5 F 40 F           }  [0]
2)  L4 7-+ 40 -+        }
3)  L4 F 40 1-+         }
4)  L4 7-+ L4 -9        }
5)  40 2-+ 41 3-+       }  [1]
6)  41 4-+ 41 5-+       }
7)  L5 8-+ 40 30L       }  [2]
8)  L5 9-+ 40 36L       }
9)  L5 3-9 L0 7-9       }  [3]
10) 36 -- L7 6-9        }  [4]
11) L0 12-+ 36 5--      }
12) 50 4-9 7J 4-9       }
13) 40 F 50 5-9         }  [5]
14) 7J 5-9 L4 F         }
15) L0 13-+ 36 10--     }
16) L5 2-+ 42 17L       }
17) N0 F L1 F           }  [6]
18) L4 3-9 32 19L       }
19) 22 21L L5 2-+       }  [7]
20) L4 -+ 40 2-+        }
21) 26 16L L5 2-+       }
22) L0 -+ 42 23L        }
23) L5 3-9 L0 F         }  [8]
24) 35 25L 22 14--      }
25) 40 -7 50 2-9        }
26) L1 S9 L4 SK         }  [9]
27) L4 2SK 70 SK        }
28) 2 14-- L5 -7        }
29) 66 4-7 -5 F         }  [10]
```

SK - 1 + 7M
1SK - 1 + 14M
2SK - F
3SK - H
4SK - I
5SK - G

(2SK–5SK: Counters)

```
30) 40  1-7  NØ  F
31) 5Ø  F  5Ø  31L          11
32) 25  -5  L4  4-9
33) 4Ø  11-9  NØ  F
34) 5Ø  1F  5Ø  34L
35) 25  -5  L4  5-9          12
36) 4Ø  12-9  L5  1Ø-+
37) 4Ø  3ØL  92  131F
38) L5  3F  82  4ØF
39) 92  967F  L5  9-9        13
40) JØ  11ØF  5Ø  4ØL
41) 26  -N  L5  11-+
42) 4Ø  36L  22  61L
43) 5Ø  6-9  75  14-+
44) ØØ  2F  4Ø  F
45) LJ  F  5Ø  45L
46) 25  -F  4Ø  F
47) 5Ø  F  75  1Ø-9          14
48) 1Ø  38F  66  1-9
49) -5  F  L4  3-9
50) 4Ø  3-9  26  9L
51) 5Ø  F  5Ø  51L
52) 25  -5  4Ø  F            15
53) L5  11-9  LØ  F
54) 4Ø  4-9  NØ  F
55) 5Ø  1F  5Ø  55L
56) 26  -5  4Ø  F            16
57) L5  12-9  LØ  F
58) 4Ø  5-9  F5  4-+
59) 4Ø  4-+  LØ  8-9         17
60) 36  61L  22  77L
61) 41  4-+  92  131F        18
62) 92  7F  L5  5-+
63) JØ  33F  5Ø  63L
64) 26  -N  L5  3-9
65) 52  66F  5Ø  65L
66) 26  -N  L5  4-9
67) 52  66F  5Ø  67L
68) 26  -N  L5  5-9          19
69) 52  66F  5Ø  69L
70) 26  -N  L5  11-9
71) 52  66F  5Ø  71L
72) 26  -N  L5  12-9
73) 52  66F  5Ø  73L
74) 26  -N  L5  6-9
75) 52  66F  5Ø  75L
76) 26  -N  NØ  F
77) NØ  F  F5  5-+
                             20
```

```
78) 40  5-+  L5  4-9   }
79) 40  2-7  L5  5-9   } [20]
80) 40  3-7  NØ  F     }
81) 50  2F  50  81L    }
82) 26  -5  40  F      }
83) 50  F  7J  2-7     }
84) 40  4-9  NØ  F     }
85) 50  3F  50  85L    } [21]
86) 26  -5  40  F      }
87) 50  F  75  3-7     }
88) L4  4-9  ØØ  4F    }
89) 40  4-9  NØ  F     }
90) 50  4F  50  9ØL    }
91) 26  -5  40  F      }
92) 50  F  7J  2-7     }
93) 40  5-9  NØ  F     }
94) 50  5F  50  94L    } [22]
95) 26  -5  40  F      }
96)  50  F  75  3-7    }
97) L4  5-9  ØØ  4F    }
98) 40  5-9  NØ  F     }
99) 50  6F  50  99L    }
100) 26  -5  40  F     } [23]
101) 50  F  75  12-+   }
102) ØØ  2F  L4  6-9   }
103) 40  6-9  F5  3-+  }
104) 40  3-+  LØ  1-9  } [24]
105) 36  1Ø6L  26  31L }
106) 41  3-+  26  31L  - [25]

ØØ  258+
```

Pl subroutine (SN-275N)

```
ØØ  286+
```

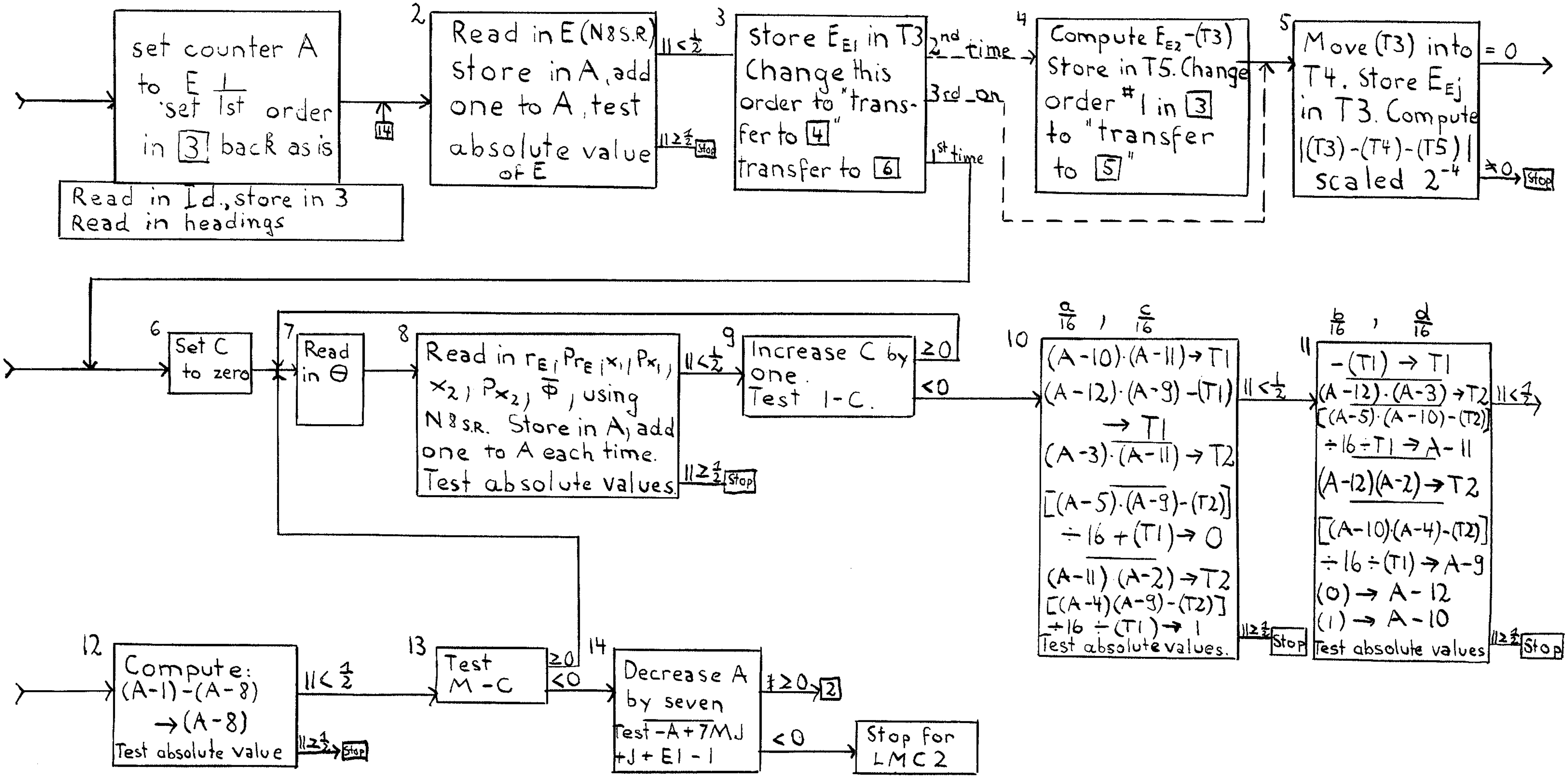
T5 subroutine (SF-20SF

```
24  999N
```

rPr overwrite

(changes order of computation first time through)

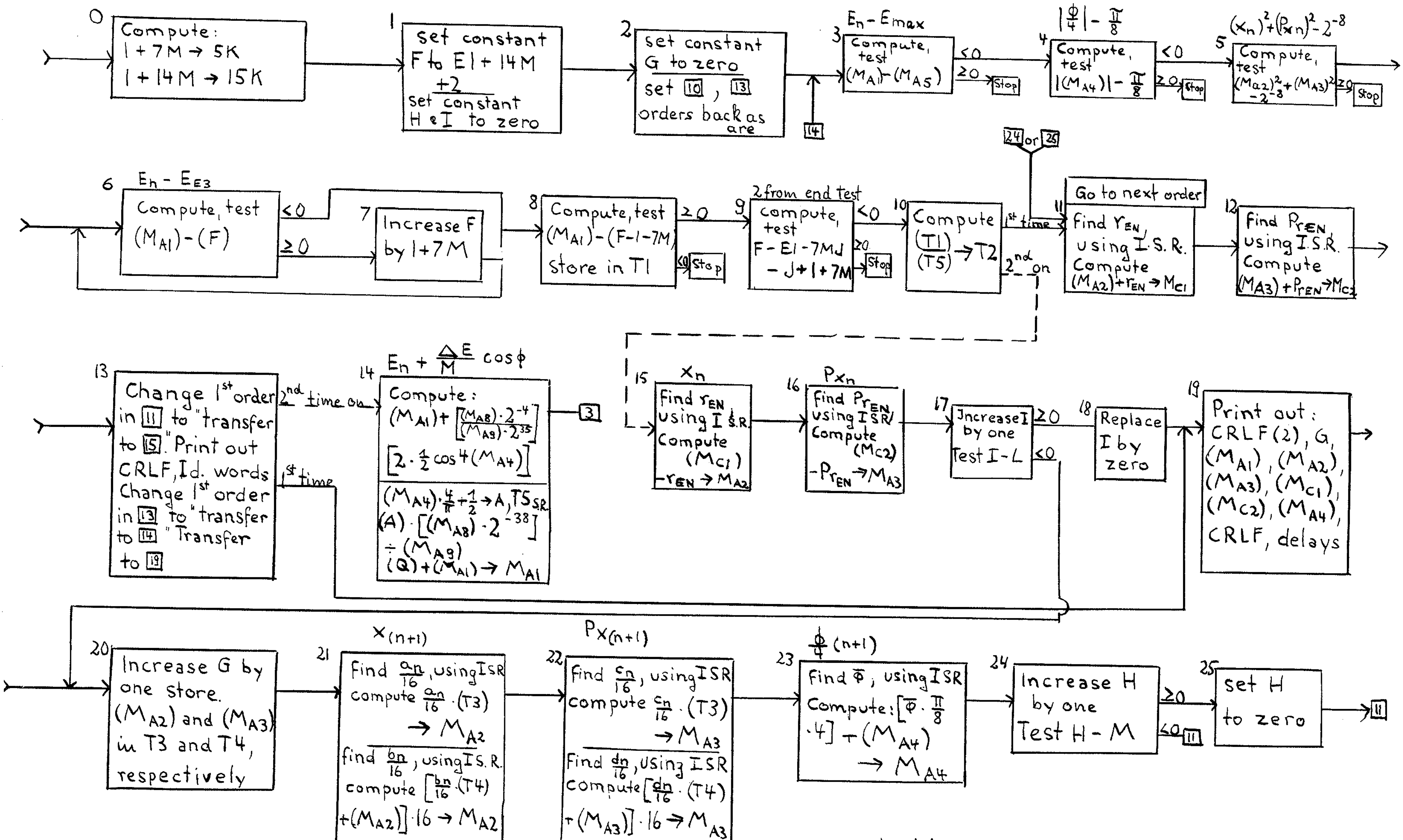
```
    ØØ 158+               } set [2] to change 1st
    L5 3Ø9F 4Ø 58-J       } order in [17]
    ØØ 181+               } set order in [11] to normal
    4Ø 1-7 26 51-J        }
    ØØ 13Ø+               } change skipped address in sum check
    ØØ F ØØ 58F           }
    ØØ 3Ø7+               }
0)  L5 31ØF 4Ø 58-J       } change 1st order in [III] to normal
1)  22 36-J ØØ F          }
2)  4Ø 5-9 26 3Ø7F          starting order in [III]
3)  4Ø 5-9 F5 4-+           normal order in [III]
    24999N
```



LMC Part I

E-O Data input
storage T1-T5, 3
E1 - 1st memory location
for E-O data storage

T1-T5 ≡ 57-457, inc.



M_{A1} - 3S9 (E)
M_{A2} - 4S9 (x)
M_{A3} - 5S9 (P_x)
M_{A4} · 6S9 ($\frac{1}{4}\phi$)

M_{A5} · 7S9 (E_{MAX})
M_{A6} · 8S9 (L)
M_{A7} · 9S9 (Id.)
M_{A8} · 10S9 (ΔE)

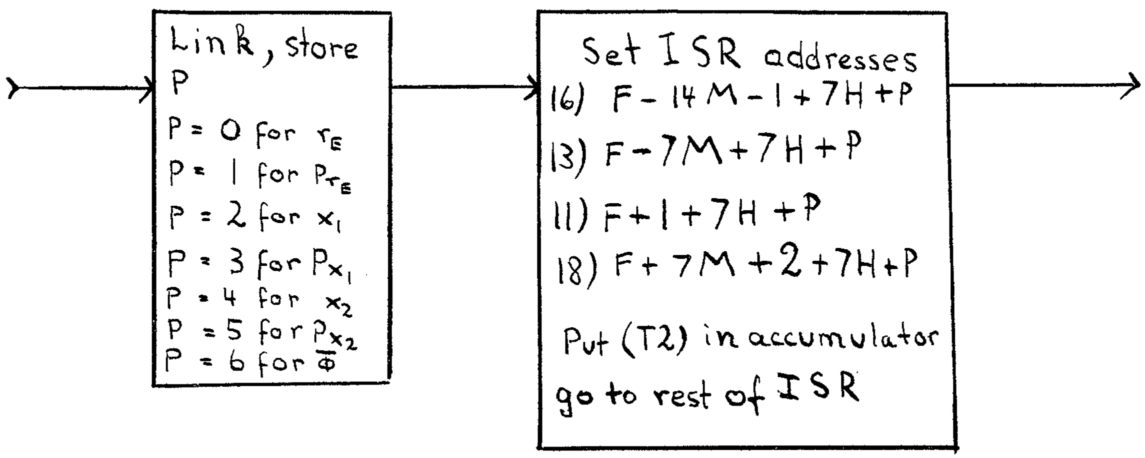
M_{A9} · 1S9 (M)
M_{A10} · 2S9 (J)
M_{A11} · S9 (EI)

M_{c1} · 11S9 (r)
M_{c2} · 12S9 (P_r)

LMC Part 2

storage - M_{C1} ; M_{C2}

LMC Part 2



ISR address setting entry