REPORT R-1777

COMPUTER, GUN DIRECTION, M18 (FADAC) APPLICABILITY IN DEFENSE COMMUNICATION SATELLITE ANTENNA POSITIONER SYSTEM

by

HELEN SMOLEN

AMCMS Code 5331.12.918
DA Project No. 1R322101D253

October 1965

**UNITED STATES ARMY
FRANKFORD ARSENAL
PHILADELPHIA, PA.**

REPORT R-1777

COMPUTER, GUN DIRECTION, M18 (FADAC) APPLICABILITY IN DEFENSE COMMUNICATION SATELLITE ANTENNA POSITIONER SYSTEM

by

HELEN SMOLEN

AMCMS Code 5331.12.918

DA Project No. 1R322101D253

Fire Control Engineering Directorate

FRANKFORD ARSENAL

Philadelphia, Pa. 19137

October 1965

FOREWORD

The study described in this report was performed by the U. S. Army, Frankford Arsenal, Philadelphia, Penna. The work was accomplished under DA Project No. 1R322101D253. Mr. A. Chalfin, of the Diagnostic and Direction Systems Engineering Branch, Fire Direction and Diagnostic Equipment Division (FDDE) was the Frankford Arsenal Project Director.

The study was performed from December 1964 to July 1965 by H. Smolen of the Computer Programming and Analysis Branch, FDDE Division.

The technical assistance of Mr. J. Junier is gratefully acknowledged.

ABSTRACT

The study described in this report was undertaken at the request of the U. S. Army Satellite Communications Agency, Ft. Monmouth, N. J., to determine the suitability of Computer, Gun Direction, M18 (FADAC), as the control unit in an Antenna Position Programmer System designed to acquire and track earth satellites.

Results of the study show that FADAC has the capability of performing the required functions, can interface with the remainder of the system, and, at operator's option, either output control data to the antenna positioner within the real time specified, or generate a punched paper tape.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

INTRODUCTION

At the request of the U. S. Army Satellite Communications Agency, Ft. Monmouth, N. J., a study was undertaken to determine the suitability of Computer, Gun Direction, M18 (FADAC) as the control unit in an Antenna Position Programmer System designed to acquire and track earth satellites. "Suitability" was defined as the ability of FADAC to operate in each of two modes, as follows:

1. MODE I - Real Time Mode

Given input parameters which define start time, the location of the ground link control station and the motion of the satellite, compute required azimuth (AZ) and elevation (EL) angles at which the antenna must be fixed to initiate tracking; accept a time signal from an external source and output the computed parameters in electrical pulse form to the next stage of the system. Update time by 10 seconds, recompute AZ and EL, and compute the change in AZ and EL as a function of change in time (ΔAZ and ΔEL), and rate of change per 1/10 second (ΔAZ/100 and ΔEL/100). On receipt of the next time signal, output the computed parameters in a fixed order. Repeat the compute and output operations at 10-second real-time intervals over a 24-hour period, a total of 8640 times.

2. MODE II - Paper Tape Generator Mode

Given input parameters which define start time, the location of the tracking station and the motion of the satellite, compute the six parameters described in (1) above. Output a properly formated Time Identification Line, followed by the six time-associated parameters, to a 5 - level paper tape punch unit. Update time by 10 seconds, recompute the 6 parameters, output an updated Time Identification Line and the 6 parameters, until 8640 sets of data (1 set for each 10-second interval over a 24-hour period) have been punched on paper tape.

In the Real Time Mode, execution time is critical, since after receipt of the first externally generated time signal, all operations described in (1) above must be completed in less than 10 seconds. Therefore, a major objective of the study was to devise a realistic measure of FADAC execution time in this mode. Toward this end, a program was designed and partially coded in FADAC mnemonics. The program was extended to include the Paper Tape Generator mode, to assure that all compute, interface, control, and tape generation problems were studied.

The software design should be considered a suggested approach to the solution of these problems, and represents only one of many possible approaches which could have been considered, given the tentative specifications which were provided at the time the study was initiated. In future final design of the computer program, system hardware must be fully defined.

DESCRIPTION OF FADAC

General

FADAC is a solid state, general purpose, digital computer designed to perform 12,500 additions or subtractions per second, and 711 multiplications or divisions per second. The computer has a rotating disc memory (1 disc revolution per 10 milliseconds) of 8192 fixed point words, each containing either a sign bit plus 31 magnitude bits, or a 32 bit instruction. Each instruction word contains the addresses of the operand and of the next command (a 1 + 1 system). The high speed operation of FADAC is a function of optimum coding and a unique multiplexing feature. Given minimum access coding, hardware design allows simultaneous, (1) operand read of the preceding instruction, (2) execution of the preceding instruction, (3) instruction read of the current instruction, (4) operand search of the current instruction, and (5) instruction search of the next instruction, all within one word time (78 micro-seconds). (Commands requiring more than one word time for execution delay computation by this additional execution time). The multiplexing characteristic of FADAC operation greatly increases its usefulness in real-time applications. Complete programming specifications are contained in Revision 1, Vol. IV of the Frankford Arsenal Notes on Development Type Materiel, FCDD-361, November 1962.

Input/Output

Input: Input to the computer is from the manual keyboard or the mechanical tape reader located on the control panel. External sources may be used to load the computer, such as:

a. Reproducer, Signal Data, AN/GSQ-64, which is a high speed photoelectric reader, designated the Memory Loading Unit (MLU). (Input at approximately 600 char/sec.)

b. Another FADAC

c. Gunnery Officer's Console

d. Other Fieldata or Teletype Equipment

e. Magnetic Tape Unit

Either Fieldata or Teletype code may be input, with automatic conversion to machine language.

Output: Output may be in the form of a visual display through the Nixie Tubes on the FADAC control panel, or to external output equipment such as:

1. Gunnery Officer's Console

2. Another FADAC

3. Battery Display Unit

4. Printer

5. Paper Tape Punch Unit

6. Magnetic Tape Unit

7. Other Fieldata or Teletype Equipment.

Output may be in Fieldata, 5-level teletype or 2-wire teletype codes, at rates of from 60 to 4000 char./sec., depending on the equipment used.

GENERAL PROGRAM DESCRIPTION

Figure 1 is a general flow diagram of the overall computer program developed to control both the Real Time and Paper Tape Generator Modes of operation. Note that the Initialization and General Computation Routines are common to both operating modes. To start the program, the SET-UP button is pressed, the Initialization Routine is executed and FADAC halts to allow the operator to select the desired operating mode. The COMPUTE button is depressed to select the Real Time Mode (RTM). The SEND button is depressed to select the Paper Tape Generator Mode (PTGM). The mode selected is reflected in the value stored in the programmed branching point (SW2). After SW2 is set, the program will automatically go through the General Computation Routine, test the value of SW2, generate the required output in a fixed order, update time, recompute the time-dependent equations 3 through 11, and repeat the cycle through a real (Mode I) or simulated (Mode II) 24 hour period.

The Initialization Routine is described in the flow diagram of figure 2. In this routine are preset the various indicators and addresses which control the paths through the program and the input of data. Start-time is input through FADAC keyboard and displayed; after which the fourteen input parameters defined in Table I are automatically read-in through the Mechanical Reader on the FADAC control panel. All inputs are converted from BCD to binary and stored.

A detailed flow-diagram of the General Computation Routine is shown in figure 3. The routine, coded in FADAC mnemonics, is contained in Appendix B. The mathematical equations used in the demonstration program were provided by the Satellite Communications Agency, and are shown in figure 3. Symbols used in the General Computation Routine (GCR) are defined in Table I and Appendix D.

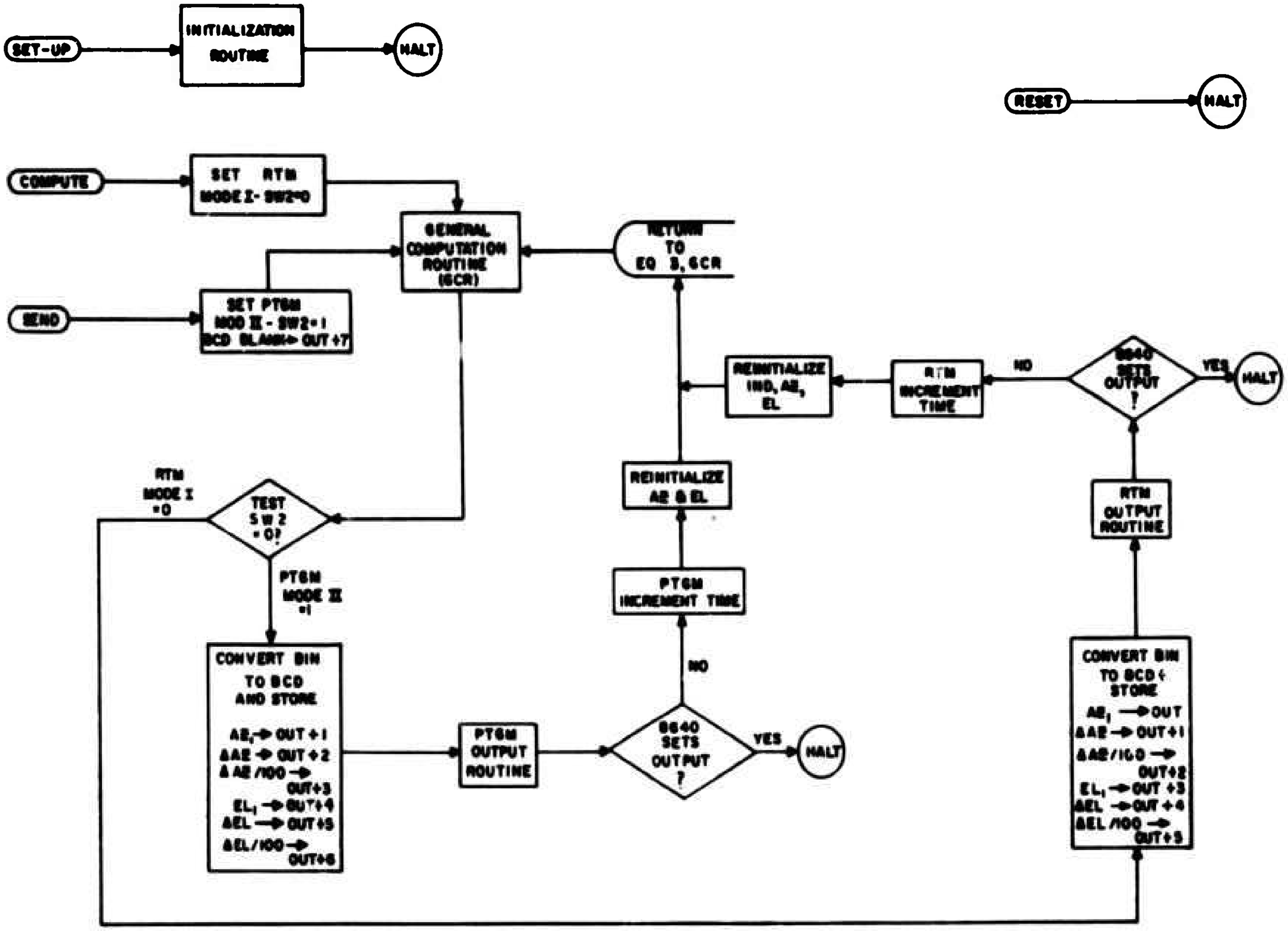

Figure 1. General Flow Diagram-Antenna Position Programmer System

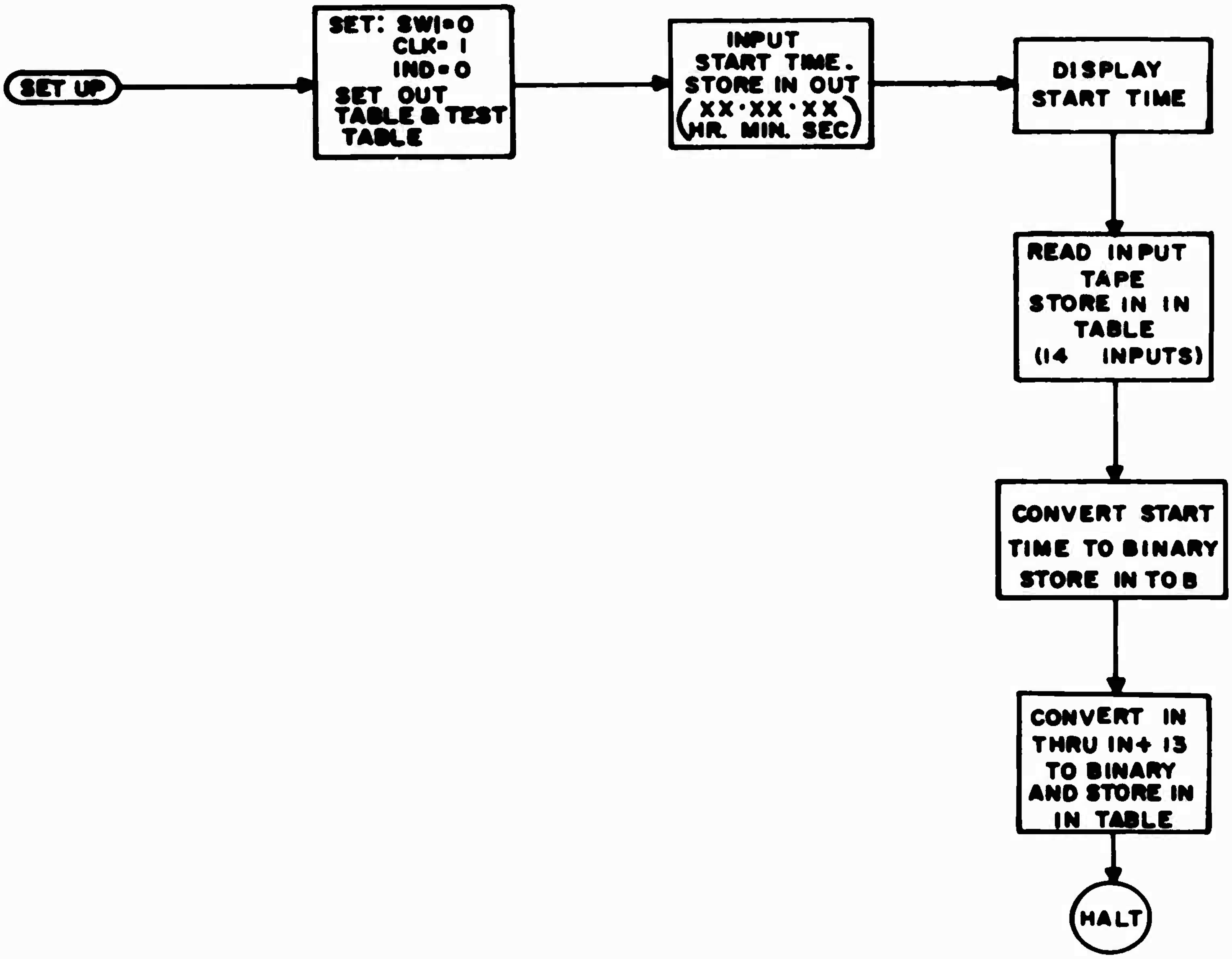

Figure 2. Flow Diagram-Initialization Routine

Table I. DEFINITION OF INPUTS

"In" Table	Symbolic Notation as shown on Program Listing	Symbolic Notation as shown on Flow Diagrams	Definition
In	THDS	θ_{ds}	Geodetic latitude
In + 1	HS	HS	Height of station above sea level
In + 2	REE	Re	Equatorial earth radius
In + 3	LAMS	λ_s	Geodetic longitude
In + 4	OI	OI	Orbital inclination
In + 5	DELO	$\Delta\Omega$	Nodal Motion
In + 6	DELC	$\Delta\Psi$	Perigee Motion
In + 7	PA	PA	Anomolistic Period
In + 8	MO	MO	Mean anomoly at epoch
In + 9	EORB	Eorb	Orbital eccentricity
In + 10	ANOM	ANOM	Semi-major axis normalized to Re
In + 11	CHIO	Ψ_o	Argument of perigee at epoch
In + 12	PHRA	PHRA	Right ascension of ascending node at epoch
In + 13	GHAR	GHAR	Greenwich hour angle of Aries at epoch.

NOTE: In through In + 13 are stored in 14 contiguous locations in FADAC memory.

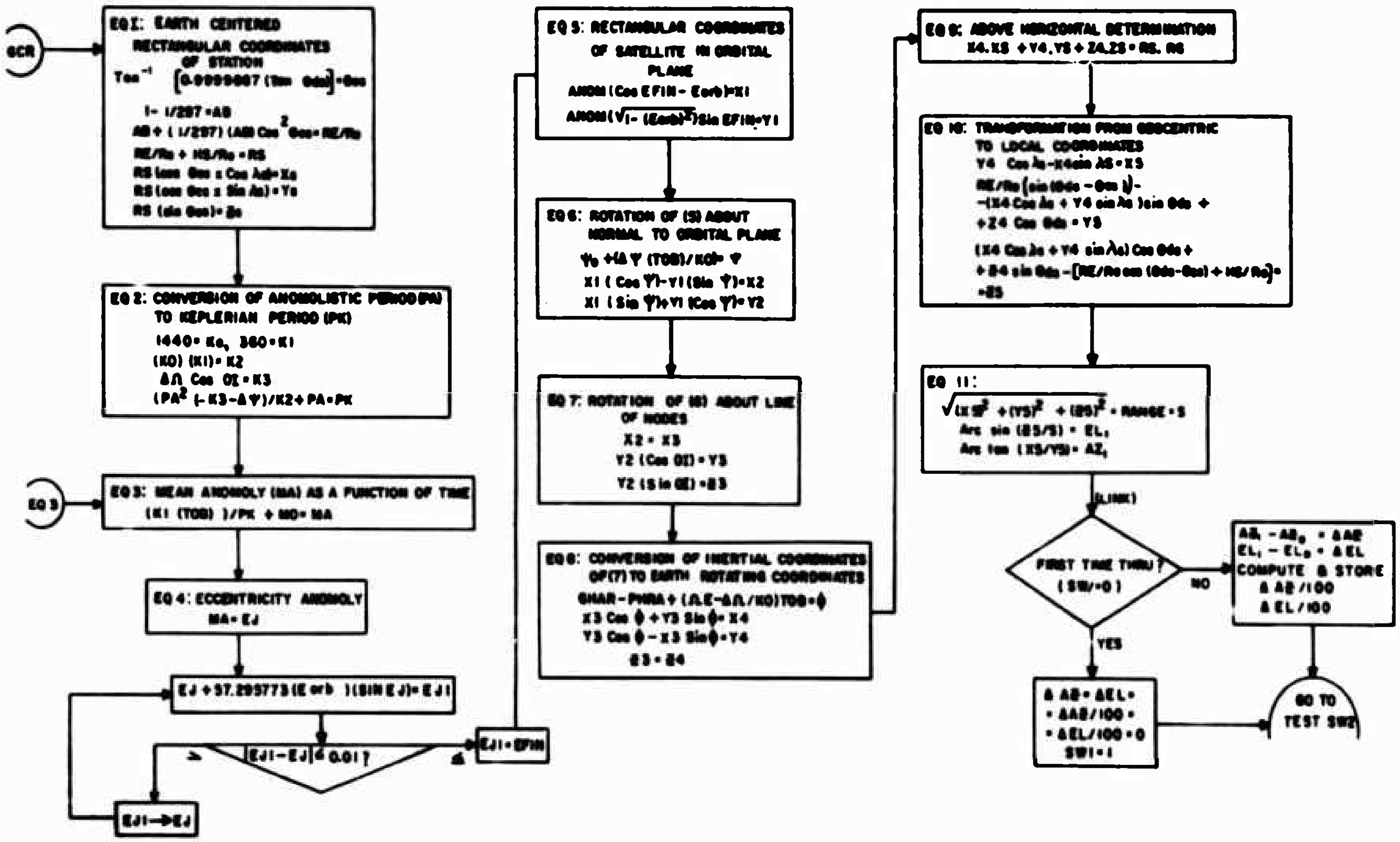

Figure 3. Flow Diagram-General Computation Routine (GCR)

DETAILED DESCRIPTION OF REAL TIME MODE

Hardware Considerations

FADAC operation in the Real Time mode will require the following peripheral equipment:

(1) A high speed Memory Loading Unit (MLU) which is used to load the program into FADAC.

(2) A timer, which will generate a signal every 10 seconds.

(3) An Output Buffer, which is to be designed to accept BCD code generated by FADAC and transmit it to the six antenna position control channels.

(4) A Remote Display Unit (RDU).

FADAC is designed to transmit data information over 5 data lines (for Teletype code) or 8 data lines (for Fieldata code), in parallel by character. In addition, six output control lines (OPL-1 thru OPL-6) are available to set or reset any desired external devices. In the Antenna Positioner System these six lines may be connected through the Output Buffer to each of the six position control channels. Thus, under program control, FADAC will transmit a given data word to a particular channel. However, this method would limit the number of output channels which can be controlled to the six output control lines available. As an alternative, the Output Buffer may contain a 6-position electronic stepping switch (or n-position, if additional control channels are required) designed to interpret a signal on one FADAC output line (OPL-1) as a command to reset the electronic switch to switch position 1, and to interpret a signal on a second FADAC output line (OPL-2) as a command to step through to the next switch position in sequence. The latter output buffer logic is assumed in the model program. (Note that the use of all six output control lines would not significantly affect the results in terms of execution time, or program complexity.)

BCD data will be transmitted from FADAC through the stepping switch to the particular antenna position channel defined by a given switch position, as shown in the following table:

With the Stepping Switch in Position No.	Data is Transmitted to:
1	AZ Channel
2	Δ AZ Channel
3	Δ A Z/100 Channel
4	EL Channel
5	Δ EL Channel
6	Δ EL/100 Channel

The logic assumed in this study requires the use of only two output control lines (OPL-1 and OPL-2)*freeing the remaining four lines for the control of other peripheral equipment.

Given an output command, FADAC hardware requires that a feedback signal be received by the computer after the output of each character. Failure to receive the feedback signal will halt output. In the design of the Output Buffer, either of two feedback methods may be used:

(1) External Feedback: Include a strobe feedback network in the design of the Output Buffer such that a feedback signal will be sent to the computer after each BCD character is received.

(2) Free-running: Wire the FADAC strobe network to accept its self-generated strobe as the required feedback after each character is transmitted. This does not require a change to FADAC hardware, since there is external access to the computer's feedback network through the J-10 output connector.

Figure 4 is a block diagram of the equipment required for FADAC operation in the Real Time Mode. The directed lines indicate the direction of data flow.

---

*A signal on OPL-1 is generated by an OD-1 command, on OPL-2 by an OD-2 command.

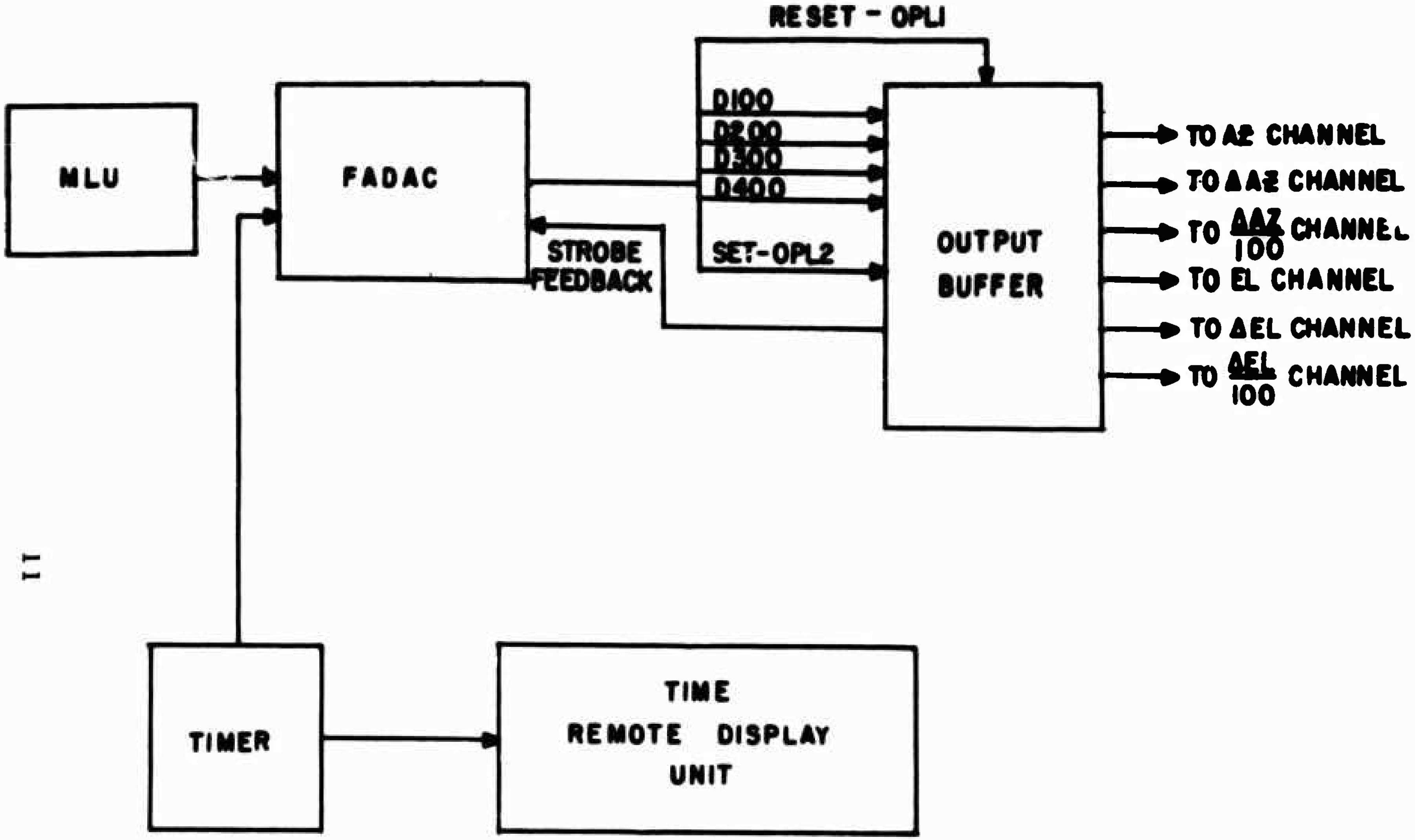

Figure 4. Block-Diagram of Equipment Required - Real Time Mode

Output Routine

A detailed flow diagram of the Real Time Mode Output Routine is contained in figure 5. To expedite output, the so-called "Out" and "Test" tables set by the Initialization Routine are stored in the R and Q loops respectively. Since output is by character rather than by word, the integrity of the contents of the two high speed loops is unaffected by the output commands. A listing of the mnemonic codes for the "Out" and "Test" loops, and the reinitialization of these loops, is contained in Appendix B.

Output Codes

FADAC will output BCD in either Baudot (TT) or Fieldata (FD) code. Study of the FD code showed that the four least significant bits of the eight bit FD code generated for decimal digits 0 through 9, and the decimal point, exactly duplicate the BCD codes for these characters. However, the FD codes for plus and minus are not equivalent to the BCD configuration.

It is assumed in this report that the RTM-Binary to BCD conversion routine contains a test for sign such that, if the sign is plus, the conversion routine is coded to generate four zero bits; if minus, the routine is coded to generate four one bits, rather than the BCD plus code (1010) or minus code (1011). Thus, in the RTM Output Routine, the sign of each parameter is tested with a single TPL command. If plus, an FD "U" is output (the four least significant bits of which are 1010); if minus, an FD "V" is output (the four least significant bits which are 1011). (See figure 5 - RTM Output Routine and Appendix B - Program Listing.) With the FADAC data lines D100 through D400 connected to the Output Buffer, and lines D500 through D800 open, BCD code can be transmitted through the electronic stepping switch circuit to a given antenna position control channel.

If higher output speeds are desired, output can be in the Alpha-4 mode rather than in Fieldata BCD. However, in the ALPHA -4 mode only 4 characters can be stored per FADAC word. It would therefore be necessary to use two FADAC locations for each 8-character word output. (It was assumed that the six output parameters could be adequately defined by 6 decimal digits, a decimal point and a sign, a total of 8 characters per output.) Also a conversion routine from binary to the desired Alpha -4

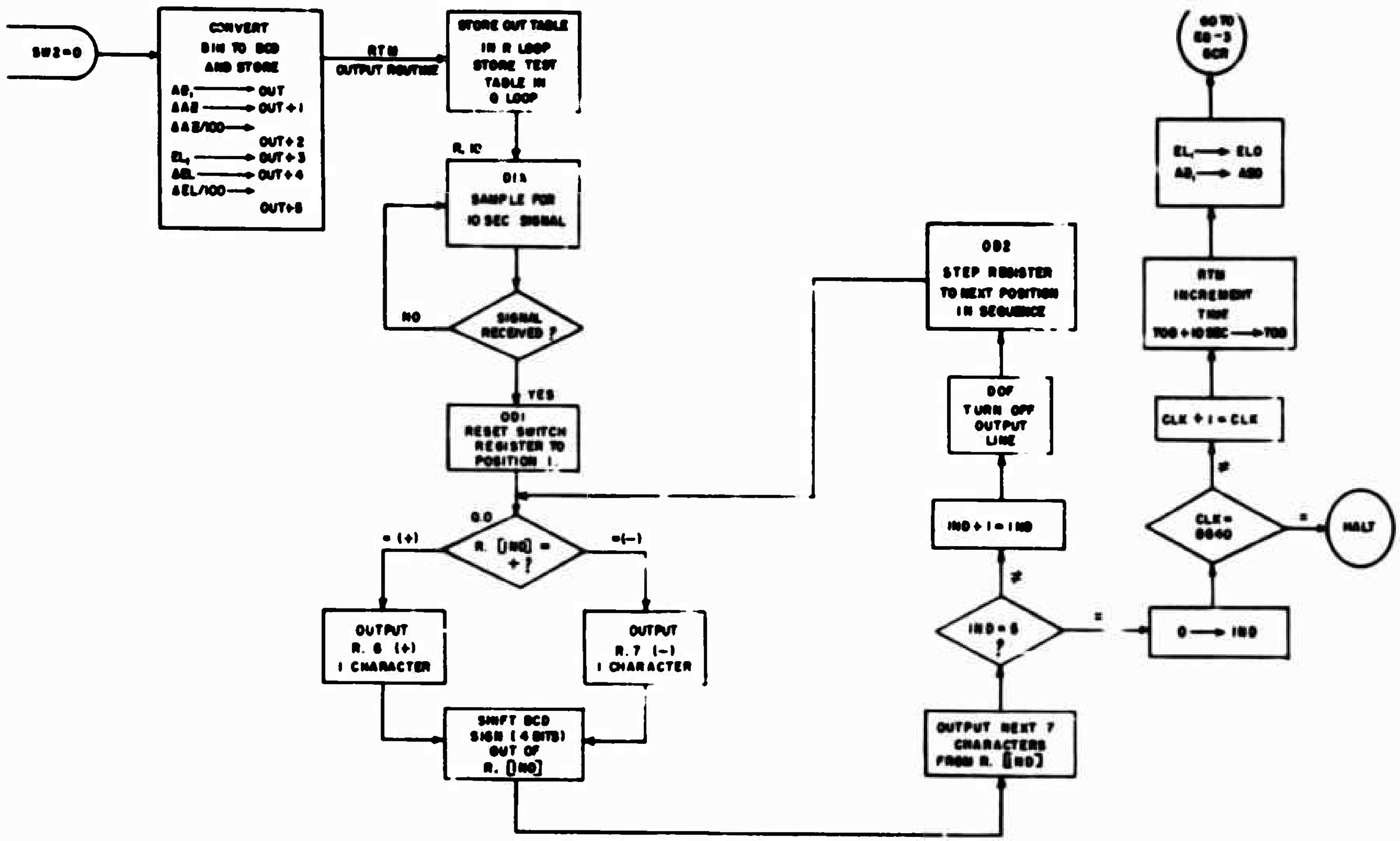

Figure 5. Flow Diagram - Real Time Mode Output Routine (RTM)

configuration would be required. However, this output mode should be considered if it is desired to form a special 8-bit code, 3 bits of which may be used to identify the control channel to which the data word is to be sent, 4 bits to define a BCD coded digit, decimal point or sign; and 1 bit of which may be suppressed by the receiving device. Fieldata BCD output is used in the demonstration program. FADAC output speeds for the two modes are as follows:

(1) BCD is transmitted at the rate of 650 char/sec.

(2) Alpha-4 is transmitted at the rate of 4000 char/sec.

Note that if FADAC is wired to generate its own feedback signal, (free running) the Output Buffer must accept data at a rate greater than the rate at which characters are output by FADAC.

Timing Estimate

To provide a basis for realistically estimating the time required to execute the FADAC program designed for the Real Time Mode (RTM) operation, the following elements of the program were coded in FADAC mnemonics:

a. General Computation Routine (GCR) (fig. 3)

b. RTM Output Routine (fig. 5)

Subroutines for the trigonometric functions, square root, BCD to Binary and Binary to BCD conversions, and special formats were not coded since the execution time of existing versions of similar or identical routines was known, or could be estimated.

A FADAC word time is 78 micro-second; a disc revolution, 10 milliseconds. Most instructions require one word-time for execution; while a STR, MPY, and DIV require 18-word times, and a LDR and LDQ require 16-word times. (See Appendix C for list of FADAC mnemonics.)

Execution time for each element coded was calculated as follows:

Let

X = Number of instructions per routine or equation

$Y = 78 \times 10^{-6}$ sec = 1 word-time

ΣY = Total minimum number of word times per routine or equation, in seconds

Z = Time required to execute each subroutine transferred to within each program element, in seconds

N = Total number of transfers to subroutines within each program element

D = .010 seconds = 1 disc revolution time

Then, Total execution time per program element

$$= \Sigma Y + \Sigma Z + (X/2)(D) + N(\dot{D})$$

Where the factor $(X/2)D$ is a worst case factor which was added to the estimated execution time to compensate for access time and scaling considerations which were not included in the sample coding. The factor $N(D)$ is an additional worst case factor which allows an entire disc revolution lost for each transfer to a subroutine.

To illustrate, the time required to set SW2 and execute Eq 1 was as follows:

$X = 49$

$(X/2)D = 25 \times (.010) =$ .250 sec

Subroutines:

Sin/Cos used twice at	.020 ea	=	.040
Tan used once at	.010 ea	=	.010
Arc tan used once at	.030 ea	=	.030

ΣZ =	.080 sec
$N(D) = 4(.010) =$	.040 sec
$\Sigma Y = 236(78 \times 10^{-6}) =$	.018 sec
Total Eq 1	.388 sec

Note that ΣY was based on 38 one-word time instructions plus 11 eighteen-word time instructions, a total of 236 word times.

Subroutines called by the program were assumed to require the following execution times:

Routine	Time in Seconds
BCD to Bin	.010 sec/char
Bin to BCD	.010 sec/char
Sin/Cos (both solutions)	.020 sec
Arc Sin	.030 sec
Arc Tan	.030 sec
Tan	.010 sec
Square Root	.010 sec

A summary of the estimated worst-case time required to execute all elements of the Real Time operating mode is contained in Table II. Note that execution time of the Initialization Routine is not included, since this routine is completed prior to entry into the Real Time operations.

DETAILED DESCRIPTION OF PAPER TAPE GENERATOR MODE

Hardware Considerations

Peripheral equipment required for FADAC operation in the Paper Tape Generator Mode is illustrated in the block diagram of figure 6. A commercially available 5, 7 or 8 level paper tape punch unit is connected to FADAC data lines through an appropriate interface. (Note that FA personnel have designed such an interface network for the Soroban Model LP-2 Paper Tape Punch Unit.)

Output Codes

The Satellite Communications Agency has indicated that any code may be used for the Tape Generator Mode which is less than 8 bits per

Table II. SUMMARY OF TIME REQUIRED TO EXECUTE FADAC PROGRAM FOR THE REAL-TIME OPERATING MODE

Program Element	Estimated Execution Time (in seconds)
Set SW2 and Execute Eq 1*	0.388
Eq 2	.160
Eq 3	.043
Eq 4	1.037**
Eq 5	.127
Eq 6	.150
Eq 7	.033
Eq 8	.180
Eq 9	.055
Eq 10	.363
Eq 11	.218
Linkage to RTM and Conversion to BCD	.764
RTM Output Routine	.416
Increment Time, Reinitialize AZ0, AZ1, E10, E11, and Return to Eq 3	1.400
Total	5.334

*Equations 1 through 11 constitute the General Computation Routine

**Assuming ten iterations

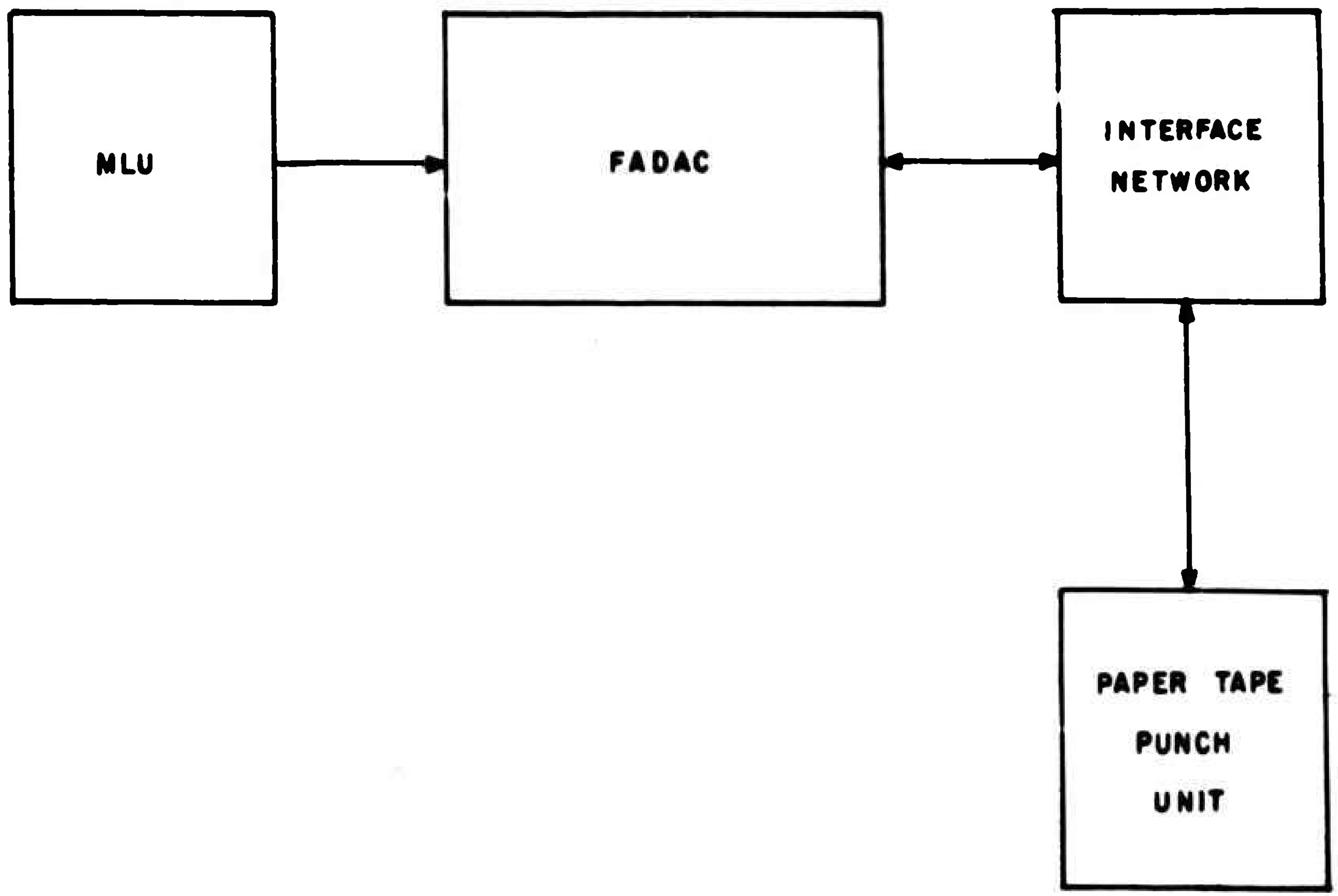

Figure 6. Block Diagram of Equipment Required - Paper Tape Generator Mode

character, to allow space on 8-level tape for the simultaneous print and punch of each character output. In the demonstration program, output is by word in 5-level (TT) code. It is assumed that since FADAC will not be used to control the Paper-Tape Track mode of operation,* equipment which is to be designed for this mode will be capable of decoding the 5-level tape code and generating the BCD code (or other format) desired. As an alternative to the 5-level code suggested in the demonstration program, the special 8-bit code described on p. 14 may be used. The first bit can be stripped by leaving data line D800 open, the channel-identifying 3-bits transmitted over lines D700 through D500, and the 4-bit BCD code transmitted over lines D400 through D100. Again, it will be necessary to design a special conversion routine and output by character in the high-speed Alpha-4 mode.

Special Features of the Mode II Program

Note that in the General Initialization Routine (fig. 2) start-time is entered through FADAC keyboard in BCD, and stored directly into location "Out". This quantity is ignored during output in the Real Time Mode. However, start-time is an important element in the Tape Generator Mode, since it is updated at 10-second intervals and serves as the Time Identification Line for each set of data output. A "set of data" is defined as the following parameters: (1) Time Identification Line, (2) AZ, (3) ΔAZ, (4) ΔAZ/100, (5) EL, (6) ΔEL, (7) ΔEL/100, (8) one word of BCD blanks. After the initial set is transmitted to the tape punch unit, time is up-dated by 10 seconds, and the new value of time stored in location "Out" in BCD, and in "TOB" in binary. See figure 7.

When the Tape Generator Mode is selected by depression of the SEND button, the programmed branching point (SW2) is set to output in Mode II, and location "Out + 7" is filled with BCD blanks. The blanks will generate a space (8-sprocket holes) between each set of data to assure that no characters will be lost when the tape is used as the source of data in the so-called Paper Tape Track mode.*

Tape Considerations

Normally, FADAC programs are punched on fanfold 5-level or 8-level paper tape, 1000 feet in length. This length is more than adequate for a program which will fill FADAC memory, since the tape will hold 10

*Described in Satellite Communications Agency Report 1011, 12 Jan 1965, FAPS.

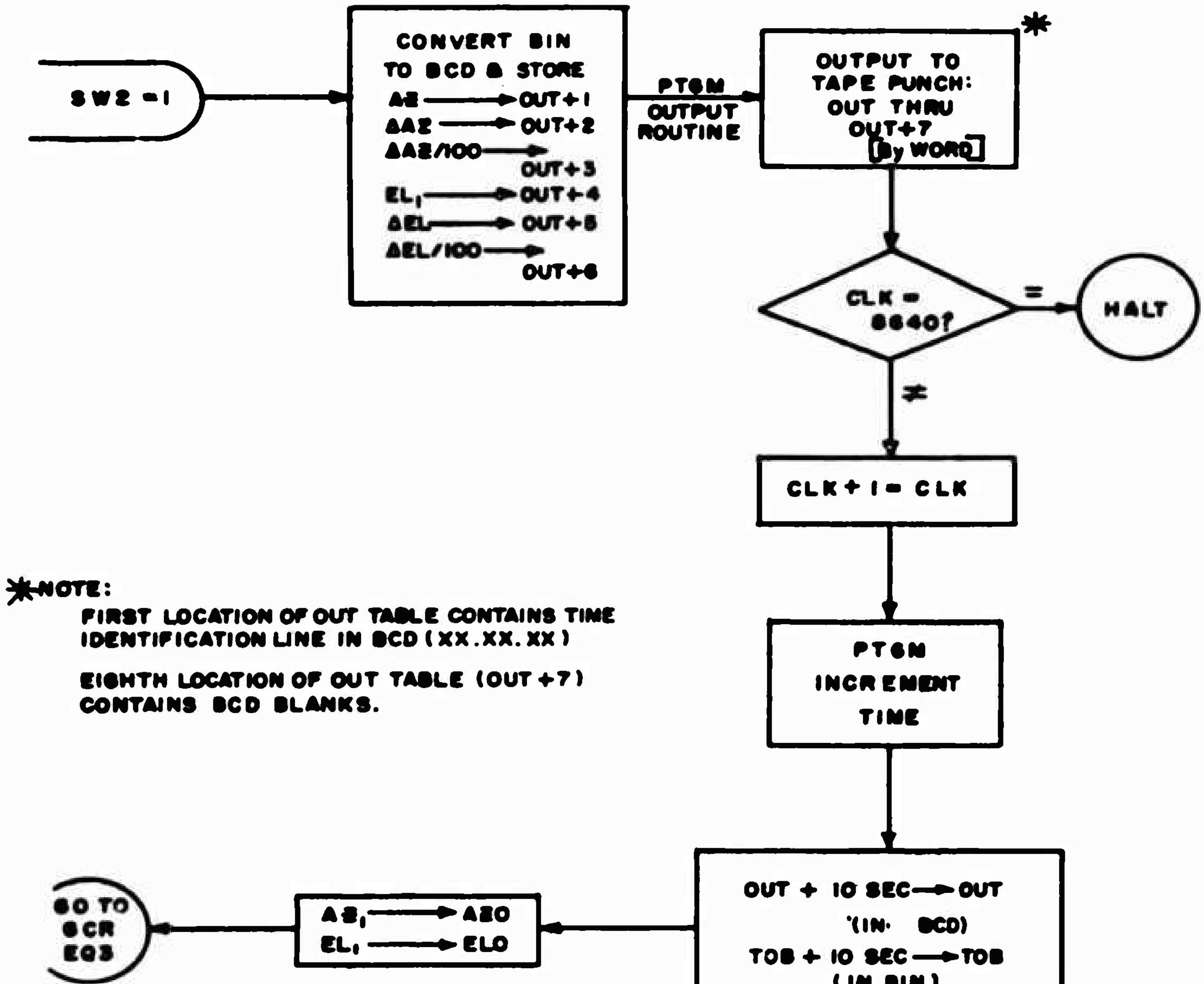

FIG 7 – FLOW DIAGRAM-PAPER TAPE GENERATOR MODE OUTPUT ROUTINE (PTGM)

Figure 7. Flow Diagram - Paper Tape Generator Mode Output Routine (PTGM)

characters per inch (120,000 char/tape). However, the tape generated during Mode II operation must contain 8640 sets of data, with each set made up of 64 characters, a total of 552,960 characters. Thus a tape for one 24-hour period must be greater than 4608 feet long. This problem was discussed with a tape manufacturer, who stated that he could not produce fan-folded tape in 5000-foot lengths, since it would be impossible to handle. Rolled paper tape was recommended. A motor driven jig can be designed to fix rolled tape reels to the MLU, and re-roll the tape as it passes beyond the read heads. As an alternative, many high-speed paper tape readers, designed to handle rolled paper tape, are commercially available.

CONCLUSIONS

The study proved that FADAC can perform all the required functions in both the Real Time and the Paper Tape Generator Modes.

The study proved that in the worst case, less than six seconds real time is required to allow FADAC to compute and transmit control data to the Antenna Position Control Channels. Therefore, the real time intervals specified for operation in the Real Time Mode can be significantly reduced.

RECOMMENDATIONS

In a review of the 11 equations solved to determine current values of AZ and EL, it was noted that Equation 9 of the General Computation Routine, "Above Horizon Determination", appears to serve no purpose. This equation calls for the solution of the product RS. RG. Guidance received from the Satellite Communication Agency indicated that if $RS^2 \leq RS.RG$, then the Satellite is above the horizon. However, there is no requirement to test this relationship, or in any way change the computations or output regardless of the value of RS. RG. It is therefore recommended that this equation be deleted from the final version of the General Computation Routine.

It is recommended that rolled paper tape be used in the final version of this program. It is further recommended that the use of a commercially available high-speed paper tape reader, other than the MLU, be investigated.

APPENDIX A

CONVENTIONS USED IN CODING

1. All symbols begin with an alphabetic character, contain no less than 2 characters, either alphabetic or numeric (except when a register A, L or N, is referenced) and no more than 4 characters.

2. The coding format is as follows:

LOCATION Field - OP-CODE - OP Address, NI Address

3. The blank addressing, symbolic, and other coding conventions used are explained in the following example:

1.	Go	CLA	In + 5
2.		MPY	KO
3.		ADD	
4.		DEC	0.023
5.		TPL	Go, No
6.	No	ADD	In
7.		STA	Out + 2, Able

Interpretation of Example:

Instruction (1): The Lo Field is labelled "Go". The Op Address "In + 5" references the sixth location in a table labelled "In" where "In" is the first location of the table, "In + 1", the second, etc. The blank NI is interpreted as "go to the next instruction in sequence", here Instruction (2).

Instruction (2): Multiply the contents of the A register by the value contained in the location labelled KO, and go to Instruction (3).

Instruction (3): Add the contents of Instruction (4) to the A register and go to Instruction (5). (Note that an instruction containing a blank Op Address is always followed by a decimal quantity. If both the Op Address and NI are blank, then the Op Address is interpreted as the next instruction in sequence and the NI is the next instruction plus 1.)

Instruction (4): The decimal quantity referenced in (3) and designated by the psuedo Op code "DEC".

Instruction (5): Test for plus; if plus, go to the location labelled "Go", if not, go to "No".

Instruction (6): Location "No" contains the command "add the contents of the first location of the table labelled 'In'", and go to Instruction (7).

Instruction (7): Store A in the third location of the table labelled "Out" and go to the location labelled "Able".

APPENDIX B

PROGRAM LISTING - REAL TIME MODE

```
COMP    TRA     SETQ
SETQ    ZEL                     SET SW2
        STL     SW2, GCR        GO TO GENERAL COMPUTATION ROUTINE
GCR     CLA     IN              EQUATION I    THDS TO A
        TRA*    TAN             TRANSFER TO TAN ROUTINE
        STA     TAND            TAN THDS
        MPY
        DEC     0.9999887
        TRA*    A TAN           ARCTAN ROUTINE
        STA     THCS
        TRA*    SINC            SIN/COS ROUTINE
        STL*    COTC            COS(THCS) TO A TO COTC
        STA     SITC            SIN(THCS)
        CLA     N
        MPY     N
        STA     SOCT            COS^2 THCS
        CLA                     (1)
        DEC     18_
        STA     0.1
```

```
ZEL
DIV                     (297)
DEC    297B
STA    D.0              (1/297 TO D.0)
CLA    D.1
SUB    D.0              (1-1/297):AB
STA    D.1              AB TO D.1
MPY    D.0
MPY    SQCT
ADD    D.1              AB+(1/297(AB)(COS^2 THCS)):RE/REE
STA    D.0              RE/REE TO D.0 : RERE
STA    RERE
CLA    IN + 1           (HS)
ZEL
DIV    IN + 2           HS/REE:HSRE
STA    HSRE
ADD    D.0              HS/REE + RE/REE : RS
STA    RS
STA    D.1              RS TO D.1
CLA    IN + 3           LAMS
```

```
        TRA*   SINC
        STA    SILA          SIN LAMS
        STL    COLA          COS LAMS
        MPY    COTC
        MPY    D.1
        STA    YS            RS(SIN LAMS . COS THCS) : YS
        CLA    D.1
        MPY    COTC
        MPY    COLA
        STA    XS            RS(COS THCS . COS LAMS) : XS
        CLA    D.1
        MPY    SITC
        STA    ZS, EQ2       RS(SIN THCS) : ZS
EQ2     CLA    KO            EQUATION 2
KO      DEC    1440B
        MPY    K1
K1      DEC    360B          KO(K1) : K2
        STA    D.1           K2 TO D.1
        CLA    IN + 4        OI TO A
        TRA*   SINC
```

```
        STA    SIOI           SIN OI
        STL    COOI           COS OI
        CLS    L              -COS OI TO A
        MPY    IN + 5         DELO
        SUB    IN + 6         DELC
        MPY    IN + 7         PA TO N
        MPY    N
        STN    D.0            PA TO D.0
        ZEL
        DIV    K2             (((-COS OI)(DELO) - DELC)PA²)/K2) +
        ADD    D.0            + PA =
        STA    PK             = PK
        STA    D.1, EQ3       PK TO A AND D.1
EQ3     CLA    TOB            TIME IN MINUTES (EQUATION 3)
        ZEL
        DIV    D.1
        MPY    KI             360(TOB)/PK
        ADD    IN + 8         MO
        STA    MA,            (360(TOB))/PK + MO : MA
ITER    STA    D.1, EQ4       MA TO D.1 (LET MA : EJ)
```

```
EC4     TRA*   SINC
        MPY    IN + 9          (SIN EJ) (EORB)
        MPY
        DEC    57.295773       180/PI
        ADD    D.1             EJ
        STA    D.0             EJ+1 : (SIN EJ(EORB)180/PI) + (EJ), EJ+1 TO D.0
        SUB    D.1
        ABS    A               /EJ+1-EJ/
        SUB
        DEC    0.01
        TZE    EQU, NOT
EQU     CLA    D.0             EJ+1
        STA    EFIN, EO5
NOT     TPL    , EQ
        CLA    D.0, ITER       (STORE EJ+1 IN EJ)
EO5     TRA*   SINC            EFIN IS IN A
        STA    SIEF            (SIN EFIN)
        STA    D.0             SIN EFIN TO D.0
        STL    COEF            COS EFIN
        CLA    L               COS EFIN TO A
```

```
         SUB    IN + 9          EORB
         MPY    IN + 10         ANOM
         STA    X1              (COS EFIN - EORB)ANOM : X1
         CLA    EORB
         STA    D.1             EORB TO D.1
         MPY    D.1
         CLS    A
         ADD    ONE             1-(EORB)²
         TRA*   SQRT            SQUARE ROOT ROUTINE
         MPY    D.0             SIN EFIN in D.O
         MPY    IN + 10         ANOM
         STA    Y1              (√1 - EORB²) SIN EFIN . ANOM = Y1
         STA    D.1, EQ6
EQ6      CLA    IN + 6          DELC
         MPY    TOB
         ZEL
         DIV    KO
         ADD    IN + 11         CHIO
         STA    CHI             DELC(TOB)/KO + CHIO : CHI
         TRA*   SINC
```

```
         STA    SICH              SIN CHI
         STL    COCH              COS CHI
         MPY    D.1               y1 in D.1
         STA    D.0               Y1(SIN CHI) TO D.0
         CLA    X1
         MPY    COCH
         SUB    D.0
         STA    X2                (-Y1(SIN CHI) + X1(COS CHI) : X2
         CLA    D.1               y1 in D.1
         MPY    N                 (COS CHI IN N)
         STA    D.1               Y1 COS CHI TO D.1
         CLA    X1
         MPY    SICH
         ADD    D.1
         STA    Y2, EQ7           Y1 COS CHI + X1 SIN CHI : Y2
EQ7      STA    D.1               Y2 TO D.1
         MPY    COOI              COS OI
         STA    Y3                Y2(COS OI) : Y3
         CLA    D.1
         MPY    SIOI              SIN OI
```

```
        STA     Z3, EQ8         Y2(SIN OI) : Z3
                                NOTE X2 EQUALS X3
EQ8     CLS     IN + 5          (DELO)
        ZEL
        DIV     KO              KO : 1440
        ADD
        DEC     0.250683        EARTH ROTATION RATE IN DEG/MIN (ΩE)
        MPY     TOB
        SUB     IN + 12         PHRA
        ADD     IN + 13         GHAR
        STA     PHI
        TRA*    SINC
        STA     SIPH
        STA     D.0             SIN PHI TO D.0
        STL     COPH
        STL     D.1             COS PHI TO D.1
        MPY     Y3              Y3 TO N
        STA     R.0             Y3(SIN PHI) TO R.0
        CLA     D.1
        MPY     N
```

```
        STA   R.2               Y3(COS PHI) TO R.2
        CLA   D.1
        MPY   X3                X3 TO N
        STA   R.4               X3(COS PHI)
        CLS   D.0
        MPY   N                 (-X3(SIN PHI))
        ADD   R.2
        STA   Y4                Y3 COS PHI - X3 SIN PHI : Y4
        CLA   R.4
        ADD   R.0
        STA   X4, EQ9           X3(COS PHI) + Y3 SIN PHI : X4
                                (NOTE.. Z3 : Z4)
EQ9     MPY   XS                (X4)(XS) TO A
        STA   D.0               TO D.0
        CLA   ZS
        MPY   Z3                Z3 : Z4
        STA   D.1               (ZS)(Z4) TO D.1
        CLA   YS
        MPY   Y4                Y4 TO N
        ADD   D.0
```

```
        ADD    D.1            (YS)Y4 + (XS)X4 + ZS24 : RSRG)
        STA    RSRG, EQ10
EQ10    CLA    COLA           COS LAMS TO D.0
        STA    D.0
        MPY    N              Y4 IN N
        STA    R.0            Y4(COS LAMS) TO R.0
        CLA    SILA           SIN LAMS TO D.1
        STA    D.1
        MPY    N
        STA    R.3            Y4(SIN LAMS) TO R.3
        CLS    D.1
        MPY    X4             X4 TO N
        STA    R.6            -X4(SIN LAMS) TO R.6
        ADD    R.0            -X4(SIN LAMS) + Y4(COS LAMS):
        STA    X5             : X5
        STA    R.5
        CLA    D.0
        MPY    N
        STA    R.10           X4(COS LAMS) TO R.10
        ADD    R.3
```

```
STA     R.4                 X4(COS LAMS) + Y4(SIN LAMS) TO R.4

CLA     THOS

TRA*    SINC

STA     R.11                SIN THOS TO R.11

STL     R.12                COS THOS TO R.12

MPY     COTC                (SIN THOS) (COS THCS)

STA     R.13                TO R.13

CLA     R.12                COS THOS

MPY     N

STA     R.14                (COS THOS)(COS THCS)

CLA     R.11

MPY     SITC

ADD     R.14               +(COS THOS COS THCS) + (SIN THOS SIN THCS)

STA     R.15                : R.15

CLS     R.12

MPY     N                   -(COS THOS)(SIN THCS) +

ADD     R.13                +(SIN THOS)(COS THCS)

STA     R.16                TO R.16

CLS     R.4                 -(X4 COS LAMS + Y4 SIN LAMS)(SIN THOS)

MPY     R.11                TO
```

```
STA     R.17              R.17
CLA     N                 (SIN THDS)
MPY     Z4                Z4 TO N
STA     R.2               Z4 SIN THDS TO R.2
CLA     N                 (Z4)
MPY     R.12
STA     R.1               Z4(COS THDS)
CLA     R.4               (X4 COS LAMS + Y4 SIN LAMS)
MPY     N
STA     R.7               COS THDS(X4 COS LAMS + Y4 SIN LAMS)
CLA     R.16              (SIN THDS COS THCS - COS THDS SIN THCS)(RERE)-
MPY     RERE
ADD     R.17              -(X4COS LAMS + Y4 SIN LAMS)(SIN THDS) +
ADD     R.1               +Z4(COS THDS) :
STA     Y5                :Y5
STA     R.3               Y5 TO R.3
CLS     R.15              -(SIN THDS SIN THCS + COS THDS COS THCS)(RERE) +
MPY     N                 +
ADD     R.2               + Z4 SIN THDS +
ADD     R.7               + (X4 COS LAMS + Y4 SIN LAMS) COS THDS +
```

```
         ADD    HSRE          + HS/REE :
         STA    Z5            : Z5
         STA    R.12, EQ11    Z5 TO R.12
EQ11     MPY    A             Z5 IS IN A
         STA    D.0           (Z5)^2  TO D.0
         CLA    R.3           Y5
         MPY    A
         STA    D.1           (Y5)^2  TO D.1
         CLA    R.5           X5 TO A
         MPY    A             (X5)^2  IN A
         ADD    D.0
         ADD    D.1
         TRA*   SQRT          TRANSFER TO SQUARE ROOT ROUTINE
         STA*   RANG          (X^2 + Y^2 + Z^2 )   : RANG,  RANG TO N
         CLA    R.12          Z5
         ZEL
         DIV    N             Z5/RANG
         TRA*   ASIN          TRANSFER TO ARC SIN ROUTINE
         STA    EL1           ARC SIN Z5/RANG : EL1
         CLA    X5
```

```
          ZEL
          DIV    Y5
          TRA*   ATAN            TRANSFER TO ARC TAN ROUTINE
          STA    AZ1, LINK       [TAN^-1 (X5/Y5) = AZ1]
                 LINKAGE FROM GENERAL COMPUTE TO ORTM OUTPUT ROUTINE
LINK      CLA    SW1
          TZE    YES, NO         TEST 1ST TIME THRU SW1
YES       ZEL                    FIRST TIME
          STL    DLAZ
          STL    DAZ             DEL AZ : DEL AZ/100 :
          STL    DLEL            : DEL EL : DEL EL/100 = 0
          STL    DEL, TSW2
NO        CLA    AZ1             NOT FIRST TIME
          SUB    AZO
          STA    DLAZ            COMPUTE DEL AZ
          ZEL
          DIV                    100
          STA    DAZ             COMPUTE DEL AZ/100
          CLA    EL1
          SUB    ELO
```

```
        STA   DLEL              COMPUTE CEL EL
        ZEL
        DIV   100
        STA   DEL, TSW2         COMPUTE DEL EL/100
TSW2    CLA   SW2
        TZE    RTM, PTPO        TEST SW2
 RTM    TRA*  QBCD              TRANSFER TO BCD CONVERSION LOOP FOR CONVERSION
                                OF 6  RTM OUTPUT PARAMETERS
        LDR   OUT               OUT TABLE TO R
        LDQ   TEST, R.10        TEST TABLE TO Q, GO TO R.10
                              RTM OUTPUT ROUTINE
              OUT TABLE AS LOADED INTO R LOOP BY LINKAGE ROUTINE
R.0                             (AZ)
R.1                             (△AZ)
R.2                             (△AZ/100)
R.3                             (EL)
R.4                             (△EL)
R.5                             (△EL/100)
R.6     FO(+)
R.7     FO(-)
```

```
R.10      DIA     , R.11              SAMPLE TIME ON LINE M

R.11      ALS2    , R.12              SHIFT SIGNAL ON LINE M TO SIGN POSITION

R.12      TPL     R.10, R.15

R.13      CLA     INIT, REQ

R.14      DEC     1831

R.15      OD1     , Q.0               RESET  SWITCH  TO POSITION 1

R.16      DOF     , R.17              TURN OFF OUTPUT LINE

R.17      OD2     , Q.0               STEP SWITCH TO NEXT  SWITCH  POSITION

INIT      CLA     R.0, Q.1

REQ       STA     TEST, NEXT

                  TEST TABLE AS LOADED INTO Q LOOP BY LINKAGE ROUTINE

Q.0       CLA     (R.0), Q.1          BRING R.(IND)

Q.1       TPL     Q.2, Q.3            TEST SIGN

Q.2       CLA     R.6, Q.4            BRING BCD(+)

Q.3       CLA     R.7, Q.4            BRING BCD(-)

Q.4       WE4     0, Q.5              WRITE SIGN

Q.5       CLA     (R.0), Q.6          R.(IND) TO AR

Q.6       ALS 4 , Q.7                 SHIFT OUT BCD SIGN

Q.7       EOF 6 , Q.10                OUTPUT NEXT 7 CHARACTERS

Q.10      CLA     Q.0, Q.11
```

```
Q.11      SUB    Q.12, Q.13          TEST (IND)

Q.12      CLA    R.5, Q.1

Q.13      TZE    R.13, Q.14          IF IND = 5, REINITIALIZE TEST LOOP

Q.14      CLA    Q.0, Q.15           IF NOT IND + 1 = IND

Q.15      ADD    R.14, Q.16

Q.16      STA    Q.0, Q.17

Q.17      STO    Q.5, R.16

NEXT      STO    TEST + 5, CHK

CHK       CLA    CLK,

          SUB

          DEC    8640831

          TZE    STOP, UP

STOP      HALT   0, 0

UP        CLA    CLK

          ADD    R.14

          STA    CLK,                GO TO INCREMENT TIME ROUTINE
```

APPENDIX C

FADAC MNEMONICS

Mnemonic	Operation
I. Arithmetic	
ADD	Add
SUB	Subtract
MPY	Multiply
CLA	Clear & Add
CLS	Clear & Subtract
DIV	Divide
II. Store & Load	
STN	Store N
STD	Store D
STA	Store A
STL	Store L
STP	Store (N1) Prog Ad
STO	Store Op Ad
STR	Store R loop
LDR*	Load R loop
LDQ	Load Q loop
III. Transfer	
TPL	Transfer on Plus
TZE	Transfer on Zero
TRA	Unconditional Transfer
TOV	Transfer on Overflow

Mnemonic	Operation
IV. Shift & Cycle	
ARC	A right cycle
ARS	A right shift
ALC	A left cycle
ALS	A left shift
LRC	Long right cycle
LRS	Long right shift
LLC	Long left cycle
LLS	Long left shift
V. Special	
EXT	Extract
EQS	Equal Search
GES	Greater than or Equal Search
HLT	Halt
HCM	Halt Compute Mode
ICM	Initiate Compute Mode
ZEL	Zero L
IDM	Initiate Display Mode
HDM	Halt Display Mode
ABS	Take absolute Value
RML	Replace A on Minus from L
RMN	Replace A on Minus from N
VI. Serial I/O	
DIA	Discrete Input to A
DOF	Discrete Outp Off
OD1	Outp Dev Stepping - 1

Mnemonic	Operation
VI. Serial I/O (Cont'd)	
OD2	Outp Dev Stepping - 2
OD3	Outp Dev Stepping - 3
ID1	Inp Dev Stepping - 1
ID2	Inp Dev Stepping - 2
ID3	Inp Dev Stepping - 3
NSL	No Solution Light

Mnemonic	Interpretation
VII. Parallel Input/Output Command Summary	
WEOT	Write FADAC to Ext Dev in Oct (TT)
WFOT	Write FADAC to FADAC in Oct (TT)
WEOF	Write FADAC to Ext Dev in Oct (FD)
WFOF	Write FADAC to FADAC in Oct (FD)
WEDT	Write FADAC to Ext Dev Dec (TT)
WFDT	Write FADAC to FADAC Dec (TT)
WEDF	Write FADAC to Ext Dev Dec (FD)
WFDF	Write FADAC to FADAC Dec (FD)
WE6	Write FADAC to Ext Dev α-6
WF6	Write FADAC to FADAC α-6
WE5*	Write FADAC to Ext Dev α-5
WF5*	Write FADAC to FADAC α-5
WE4	Write FADAC to Ext Dev α-4
WF4	Write FADAC to FADAC α-4
REO	Read Ext Dev in Oct
RTO	Read Tape Dev in Oct
RKO	Read Keyboard in Oct

Mnemonic	Interpretation
VII. Parallel Input/Output Command Summary (Cont'd)	
RED	Read Ext Dev in Dec
RTD	Read Tape Dev in Dec
RKD	Read Keyboard in Dec
RE6	Read Ext Dev in α-6
RT6	Read Tape Dev in α-6
RM6	Read Mag Tape in α-6
RK6	Read Keyboard in α-6
RE5	Read Ext Dev in α-5
RT5	Read Tape Dev in α-5
RM5	Read Mag Tape in α-5
RK5	Read Keyboard in α-5

APPENDIX D

DEFINITION OF SYMBOLS OTHER THAN INPUTS, USED IN GENERAL COMPUTATION ROUTINE

Symbolic Notation as Shown on Program Listing	Symbolic Notation as Shown on Flow Diagram	Definition
THCS	θ_{cs}	Geocentric Latitude
1/297	1/297	Eccentricity Constant
XS, YS, ZS	X_s, Y_s, Z_s	Station Coordinates
TOB	TOB	Time from Epoch (in binary)
EJ	EJ	Let MA = EJ and iterate until $\lvert EJ1 - EJ \rvert \leq 0.01°$; then EFIN = EJ1, and EJ1 = result of each iteration. (See Eq 4.)
ANOM	ANOM	Semi-major axis normalized to Re
X1, Y1	X1, Y1	Satellite Coordinates in Orbit Plane
CHIO	Ψ_o	Argument of Perigee at Epoch
X2, Y2	X2, Y2	Satellite Coordinates Normal to Orbital Plane
X3, Y3, Z3	X3, Y3, Z3	Where X3 is along Ascending Mode, Y3 is in equatorial plane 90° E of X3 and Z3 is along Earth's axis.
GHAR	GHAR	Greenwich Hour Angle of Aries at Epoch
0.250683	ω_E	Earth's rotation rate in degree/min.

Symbolic Notation as Shown on Program Listing	Symbolic Notation as Shown on Flow Diagram	Definition
X4, Y4, Z4	X4, Y4, Z4	Earth Rotating Coordinates
RS	RS	Radius center of Earth to station
RG	RG	Radius center of Earth to satellite
X5	X5	Eastward)
Y5	Y5	Northward) Local Coordinates
Z5	Z5	Zenith)

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) FRANKFORD ARSENAL Philadelphia, Pa. 19137	2a. REPORT SECURITY CLASSIFICATION Unclassified
	2b. GROUP

3. REPORT TITLE
COMPUTER, GUN DIRECTION, M18 (FADAC) APPLICABILITY IN DEFENSE COMMUNICATIONS SATELLITE ANTENNA POSITIONER SYSTEM

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Final Report

5. AUTHOR(S) (Last name, first name, initial)
SMOLEN, Helen

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
October 1965	48	

8a. CONTRACT OR GRANT NO. AMCMS Code 5331.12.918	9a. ORIGINATOR'S REPORT NUMBER(S) R-1777
b. PROJECT NO. 1R322101D253	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. AVAILABILITY/LIMITATION NOTICES
Report authorized for release to CFSTI.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	U. S. Army Satellite Communications Agency

13. ABSTRACT

The study described in this report was undertaken at the request of the U. S. Army Satellite Communications Agency, Ft. Monmouth, N. J., to determine the suitability of Computer, Gun Direction, M18 (FADAC), as the control unit in an Antenna Position Programmer System designed to acquire and track earth satellites.

Results of the study show that FADAC has the capability of performing the required functions, can interface with the remainder of the system, and, at operator's option, either output control data to the antenna positioner within the real time specified, or generate a punched paper tape.

DD FORM 1 JAN 64 1473

Unclassified
Security Classification

14. KEY WORDS	LINK A ROLE	LINK A WT	LINK B ROLE	LINK B WT	LINK C ROLE	LINK C WT
Antenna Position Programmer System						
Satellites, tracking						
Computer, Gun Direction M18 (FADAC)						

INSTRUCTIONS

1. **ORIGINATING ACTIVITY:** Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. **REPORT SECURITY CLASSIFICATION:** Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. **GROUP:** Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. **REPORT TITLE:** Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. **DESCRIPTIVE NOTES:** If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. **AUTHOR(S):** Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. **REPORT DATE:** Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. **TOTAL NUMBER OF PAGES:** The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. **NUMBER OF REFERENCES:** Enter the total number of references cited in the report.

8a. **CONTRACT OR GRANT NUMBER:** If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. **PROJECT NUMBER:** Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. **ORIGINATOR'S REPORT NUMBER(S):** Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. **OTHER REPORT NUMBER(S):** If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. **AVAILABILITY/LIMITATION NOTICES:** Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through ________________."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through ________________."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through ________________."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. **SUPPLEMENTARY NOTES:** Use for additional explanatory notes.

12. **SPONSORING MILITARY ACTIVITY:** Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. **ABSTRACT:** Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS), (S), (C), or (U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. **KEY WORDS:** Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.