

# Tape contents (300 BAND)

FD00 - FD16 - Gate MPX Test

FD20 - FD7E - Digitize "A"  
 i. Fac is the slider address location

Will not run

A = 0008  
 D = 0100  
 S = 0200  
 R = 0300

FD80 - FD9E

ADSR (Trace Debug)

Time Values located at  
 A → FD94

D → FD98

S → FD9C

R → FDA0

all,

FE00 - won't load  
 via keyboard  
 (out of memory?)

AR - diff K env.  
 Creation of AR Values

A → FE0A

R → FE0E

2604  
 A583  
 2502  
 2481

DOKE

MuZof

Side 2  
 300 BAND

1. 0000 - 0020

2. 0030 - 0060

3.

0030 -

-35008

-35000

-35300

6021100(0+7)

6021100(0+7)

AD CONV

A TO D PGM 3E00 → 3E6E

~~MX ADSR PGM 3000 → DAB~~

MX ADSR PG FD00

FD02

> / 11 / > 8

6021100(0+7)

E000 42 VSE V200 CHNVCLEST

000 42 VSE ~~MODIFIER~~

WYB 0E CW K64X62H POCV1000

MAP OF CRT REFRESH LOCATIONS  
 ODD #S ARE ~~ASCII~~ MODIFIERS  
 EVEN #S ARE ASCII CHARACTERS

LINE #	POSITION(0+L)	POSITION(1+L)	POSITION(79+L)
LINE 0		-32766	-32610
LINE 1	-32608	-32606	-32450
LINE 2			

POKE

~~29094~~  
 -29093, 255  
 -29094, 00

E → E080  
 Z → E080  
 D → E088  
 H → E084

E D20 - DEE

(D20 - DEE)  
 K = 0000  
 L = 0000  
 M = 0000  
 H = 0000

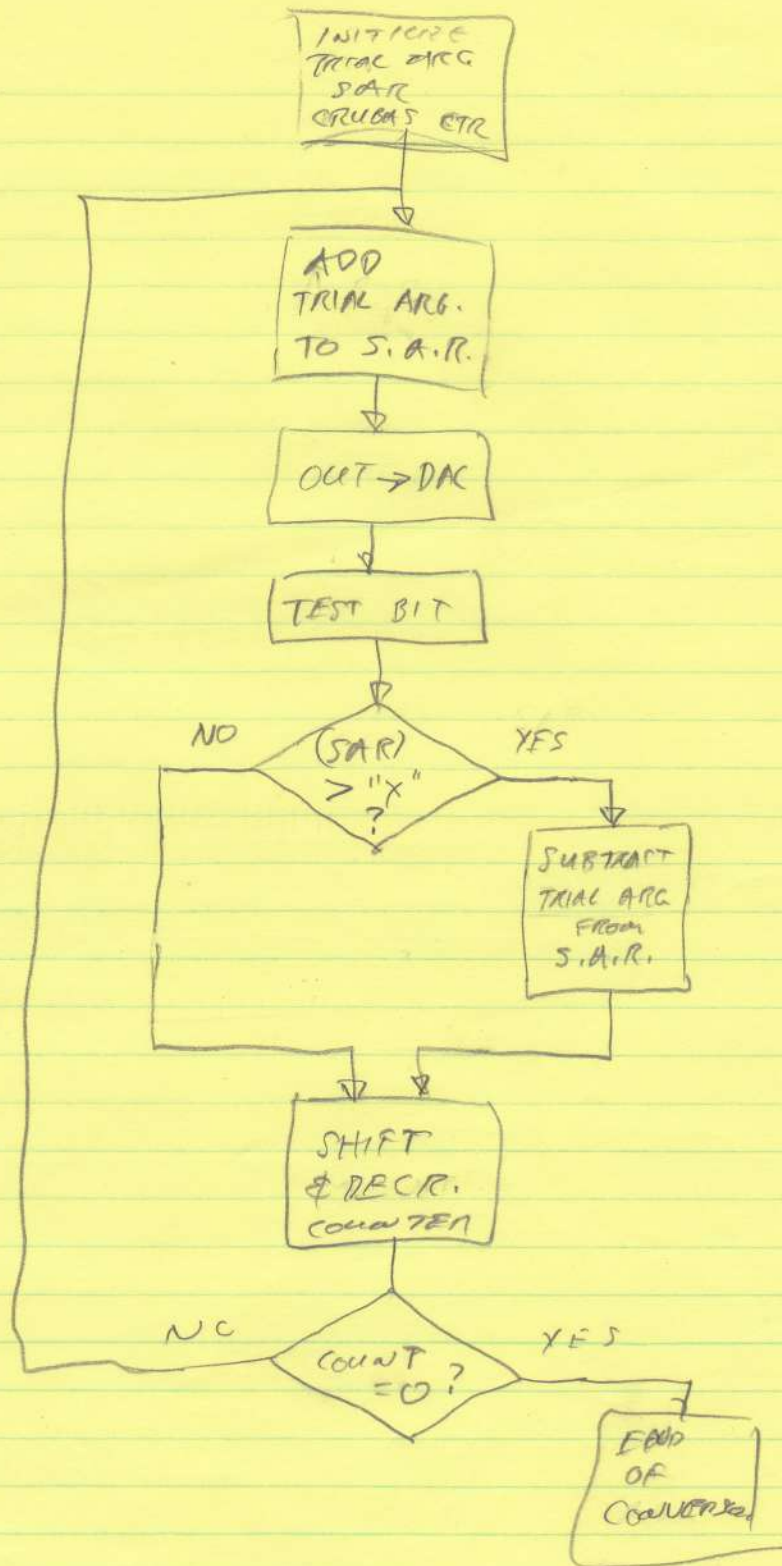
E D20 - DEE

E D20 - DEE

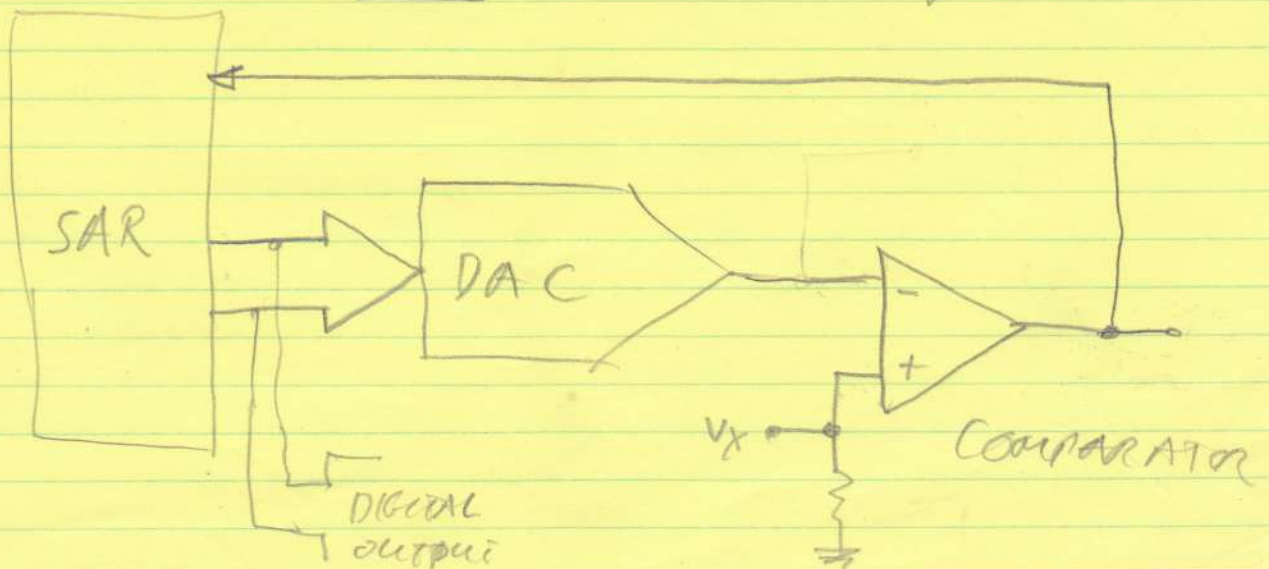
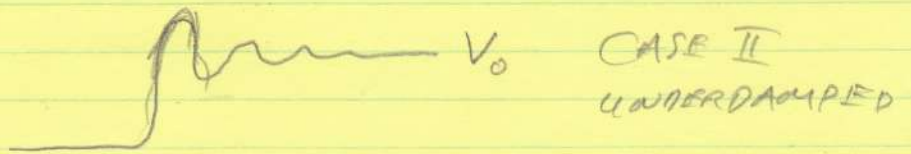
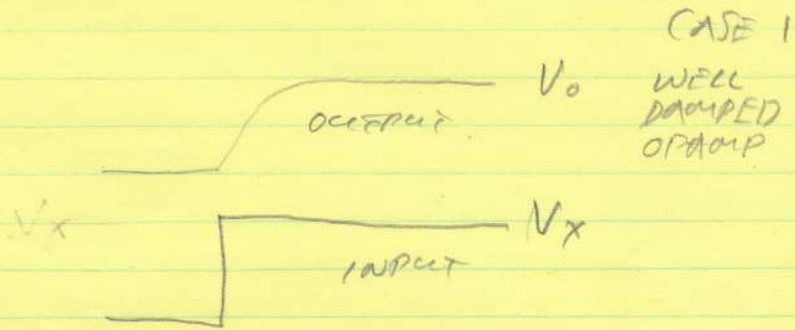
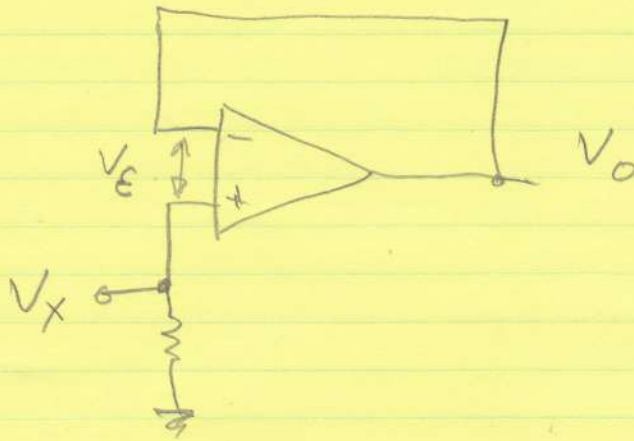
job - 0000 (300 HUND)

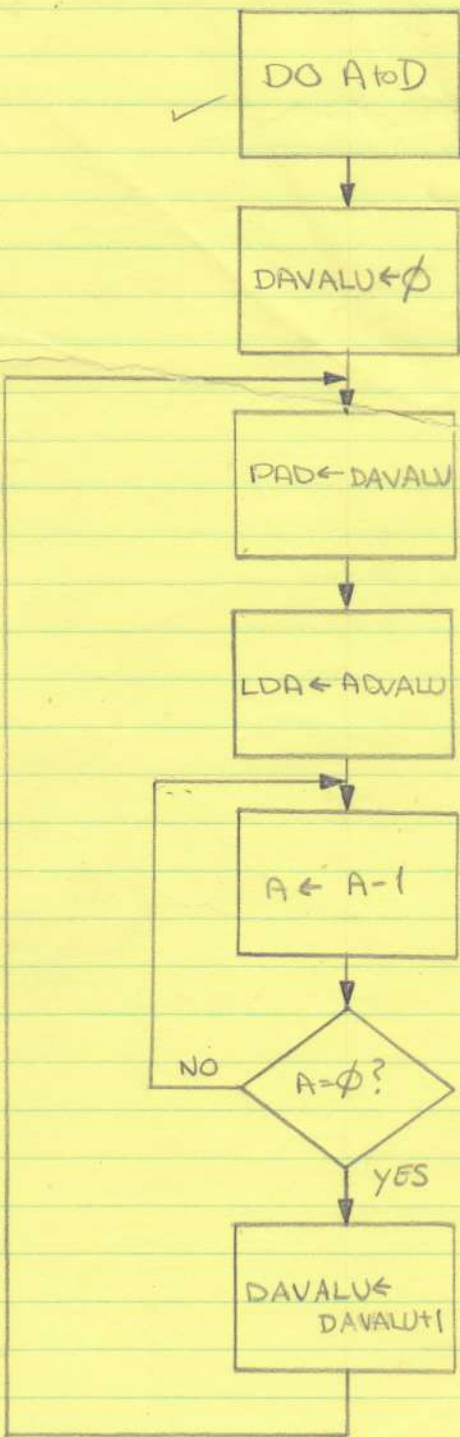


R1 2 ↑ N  
R3 S.A.R.



$$V_o = V_E * A_{OL}$$





A9

STA ADVALU

STAIR ∅

load ADVALU from mem to acc.

DEcrement acc. by one

Branch if acc is Not Equal to Zero  
if it is equal to zero

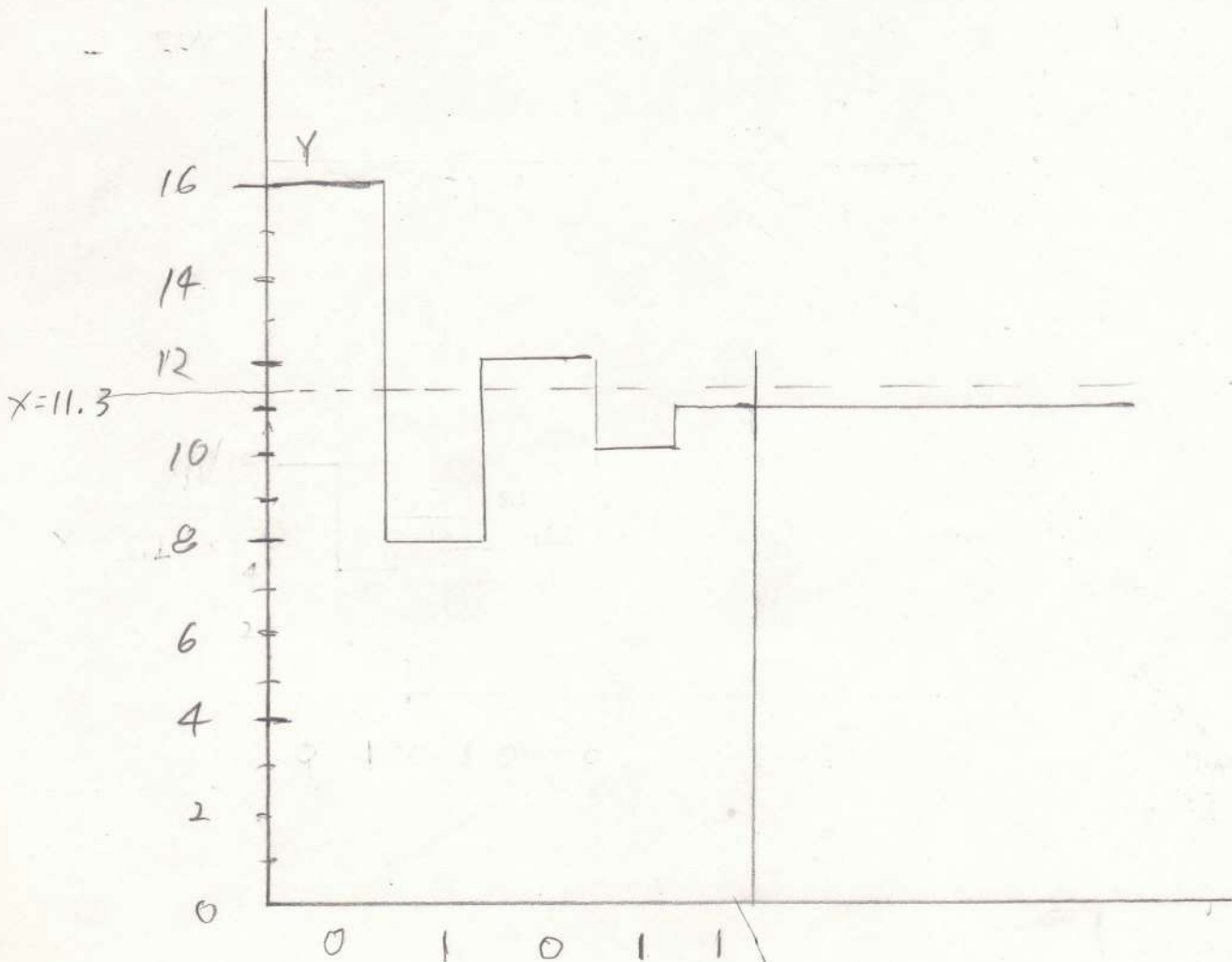
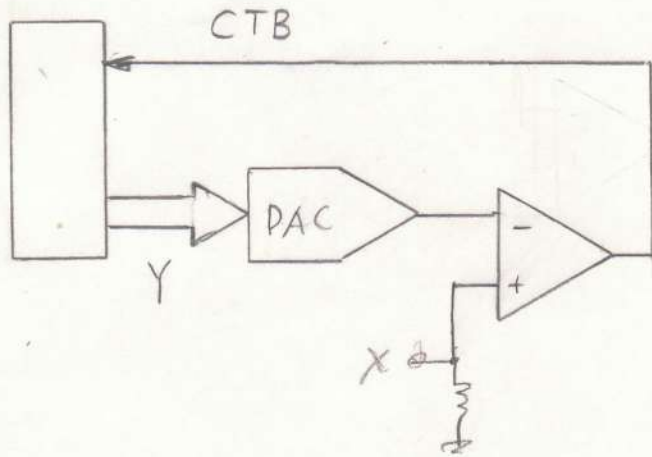
INCrement acc with one & loop



# 5 BIT CONVERSION

## REGISTERS

- R0 SHIFT COUNT
- R1 SHIFTED N
- R2 Y ACCUM
- R12 CRU BAS

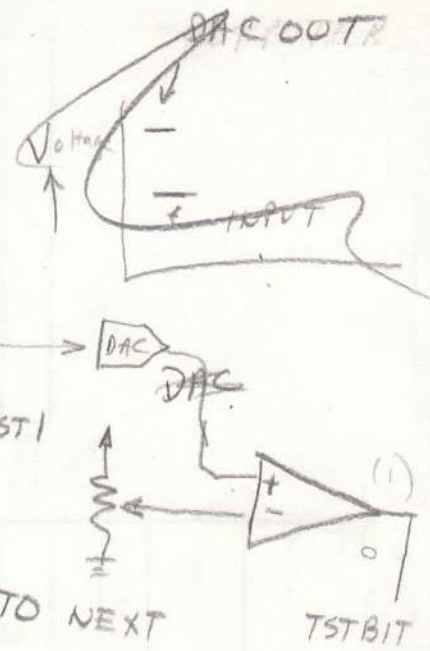




```

LET LOOP_CTR = 810
LET TEST1 = 10000000
LET TEMP = 000000002
NEXT LET TEST2 = TEMP + TEST1
      OUTPUT TEST2 TO DAC
      IF TST BIT = 0 THEN TEMP = TEMP + TEST1
      ROTATE TEST1 RIGHT
      IF LOOP_CTR - 1 ≠ 0 THEN GO TO NEXT
      ADVALU = TEMP
END

```





A/D CONV  
TM 990/100

WORKSPACE REG'S

R0	
R1	TRIAL ARGUMENT
R2	Y ACCUMULATOR
R3	STEP COUNTER
R4	
R5	
R6	
R7	
R8	
R9	
R10	
R11	
R12	CRU BAS > 130
R13	
R14	
R15	

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
FE00		INIT	LWPI > FF20	A/D WORKSPACE
02	FF20			
04			CLR R0 + 1000	
06			LI R1, > 4000	TRIAL ARGUMENT
08			<del>LI R2, &gt; 1000</del>	
0A			CCR R2	
0C			LI R3, > 7	INIT STEP COUNTER
0E				
0E			CCR R4	
FE10		INIT	CCR R5	
2			CCR R6	
4			CCR R7	
6			CCR R8	
8			CCR R9	
A			CCR R10	
C			CCR R11	
1E			LI R12, > 130	SET CRUBAS
FE20				
22			CLR R13	
24			CLR R14	
26			CCR R15	
28			JMP \$+2	NOP
2A			JMP \$+2	NOP
2C			JMP \$+2	NOP
2E			JMP \$+2	NOP

date 4/19/78

PROGRAM A/D CONV

pg 1 of 2



ADSR

WORKSPACE REG'S

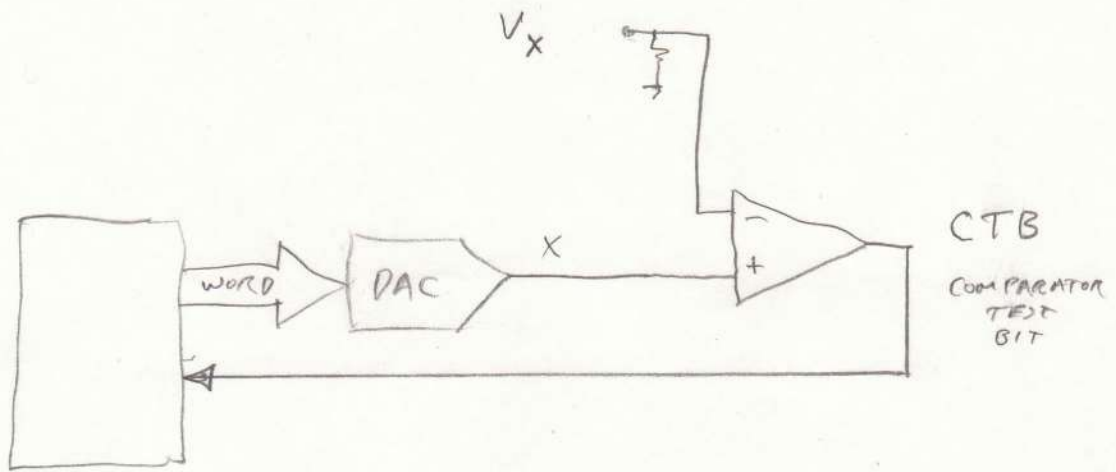
R0	
R1	
R2	Y = OUTPUT AMPL
R3	SCRATCH MSW
R4	SCRATCH LSW
R5	G = GATE AMPL
R6	A = ATTACK CONST
R7	D = DECAY "
R8	S = SUSTAIN LEVEL
R9	R = RELEASE COMP
R10	
R11	
R12	CRU BASE ADDRESS
R13	
R14	
R15	

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
3B		ATTACK		
FD40	3CC6		DIV R6, R3 SRL R3, 1 MOV R5, R1 A R13, R2 S / R1	LG/A → R3 SR3 = G/2A R1 = G

date

PROGRAM

of



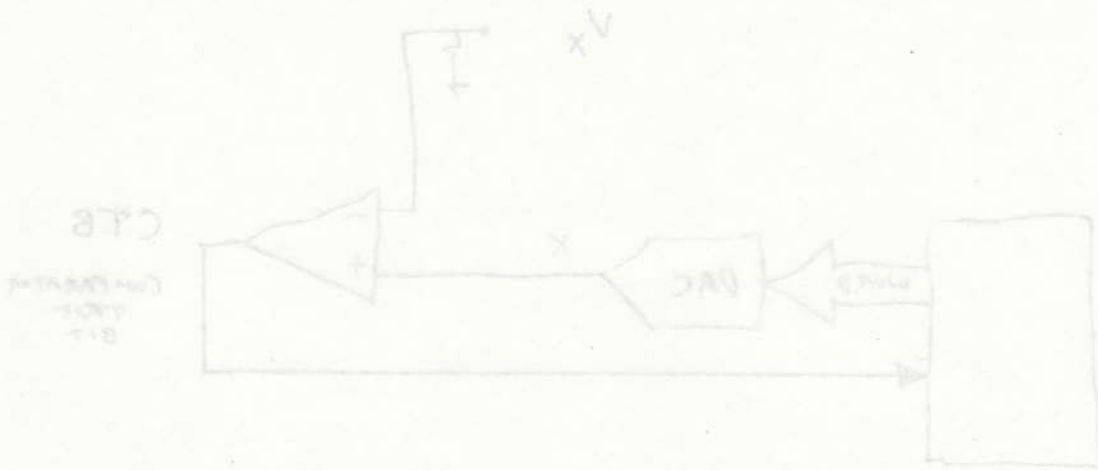
IF  $X < V_x$  THEN  $CTB = 1$   
 IF  $X > V_x$  THEN  $CTB = 0$

$X = R \cos(A)$   
 $Y = R \sin(A)$



$R = \sqrt{X^2 + Y^2}$



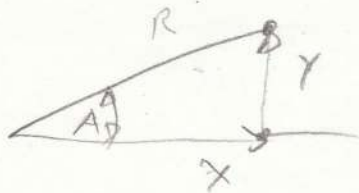


if  $X > V$  then  $CTB = 1$   
 if  $X < V$  then  $CTB = 0$

$$\text{LET } A = \text{ATAN}(1/9) = 5^\circ$$

$$X = R \cos(A)$$

$$Y = R \sin(A)$$



$$\text{LET } R = \text{SQR}(A)$$

ADSR

WORKSPACE REG'S

R0	
R1	SCRATCH (Y-B)
R2	Y = OUTPUT AMPL
R3	SCRATCH MSW
R4	SCRATCH LSW
R5	G = GATE AMPL
R6	A = ATTACK CONST
R7	D = DECAY "
R8	S = SUSTAIN LEVEL
R9	R = RELEASE CONST
R10	
R11	
R12	CRU BASE ADDRESS
R13	
R14	
R15	

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
FD 60	6083		S R3, R2	Y = Y - (Y/R)
2	1000		JMP \$ + 2	NOP
4			JMP \$ + 2	
6			JMP \$ + 2	
8			JMP \$ + 2	
6A			JMP \$ + 2	
6C			JMP \$ + 2	
6E	10E0		JMP \$ - 60	GOTO "OUTPUT" & CONTINUE
FD 70	04C3	DECAY	CLR R3	MS 16 BITS 32 BIT WORD
72	C102		MOV R2, R4	R2 ← Y → R4
74	6108		S R8, R4	(Y-S) → R4
76	3CC7		DN R7, R3	R3 = (Y-S) / D
78	6083		S R3, R2	NEW Y DECAY TOWARD S
7A	1000		JMP \$ + 2	NOP
7C			JMP \$ + 2	
7E			JMP \$ + 2	
FD 80			JMP \$ + 2	
82			JMP \$ + 2	
84			JMP \$ + 2	
FD 86	10D5		JMP \$ - 84	GOTO "OUTPUT" & CONTINUE

data PROGRAM



ALG 990/100

WORKSPACE REG'S

R0	
R1	TESTER (TRIAL ARG)
R2	
R3	S.A.R.
R4	
R5	COUNTER
R6	
R7	
R8	
R9	
R10	
R11	
R12	CRU BASE
R13	
R14	
R15	

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
FD00		INIT	LD R1, >FF00	SET WORKSPACE POINTER
02			LD R12, >130	SET CRU BASE
04			LD R5, 8	COUNTER = 8
06			LD R1, >4000	ESTEC = 0100 0000 0000 0000
08				
0A				
0C				
0E				
10		NEXT	LD R3	
12		NEXT	LD R1, R3	
14		NEXT	LD R2, R3	
16			LD R3, R3	
18			LD R3, R3	
1A			LD R3, R3	
1C		SUBTRAT	LD R3, R3	
1E		SHIFT	LD R3, R3	
20			LD R3, R3	
22			LD R3, R3	
24			LD R3, R3	
26			LD R3, R3	
28			LD R3, R3	

date 4/19/78

PROGRAM

pg 2 of 2



ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
FD00		INIT	LWPI > FFO0	SET WORKSPACE POINTER
02				
04			L1 R12, 7130	SET CRU BASE
06				
08			L1 R5, 8	COUNTER=8
0A				
0C			L1 R1, 7A00	TESTER = 0100 000 000 000
0E				
FD10			CLR R3	
12			> MIP \$12	
14		NEXT	A R1, R3	
16			(PCR R3, 8	
18			TB > FB	
1A			JEQ \$ +4	
1C			S R1, R3	
1E			SRL #1	
FD20			DEC R5	IF COUNTER # 0,
22			JNES \$ -14	GOTO NEXT
24		END	IDLE	ELSE, HANG

FF00

WORKSPACE REG'S

R0	
R1	TESTER (TRIAL ARG)
R2	
R3	
R4	S.A.R
R5	COUNTER
R6	
R7	
R8	
R9	
R10	
R11	
R12	CRU BASE
R13	
R14	
R15	



W-
P-
S-

WORKSPACE REG'S

	R0
	R1
	R2
	R3
	R4
	R5
	R6
	R7
	R8
	R9
	R10
	R11
	R12
	R13
	R14
	R15

ADDRESS	OPCODE	LABEL	INSTRUCTION	COMMENTS
0				
2				
4				
6				
8				
A				
C				
E				
0				
2				
4				
6				
8				
A				
C				
E				
0				
2				
4				
6				
8				
A				
C				
E				

PROGRAM

data

## DISK

### STARTING UP:

- 1) After loading PTOS, load tape "disk obj." or # Disk from 4000 to 43CE, making certain there is memory there.
- 2) Put your disk in the drive and do an "exec 4000".

### COMMANDS

#### 4000 - DISK DRIVER INITIALIZATION

Must be done when a new disk is put on the drive. May be done at any time, particularly if you are suspicious of problems.

#### 4003 - DISK READ

Command to move data from disk to memory. All moves are in 128-byte sections (80h). You are prompted for responses (see track/sector below).

#### 4006 - DISK WRITE

The inverse of disk read.

### SUBROUTINES

#### 4009 - ENABLE DISK

Corresponds to disk initialization. Put drive number in A.

#### 400C - READ DISK

HL  $\rightarrow$  128 byte buffer  
D = track  
E = sector  
on return, CY = 0 if good

#### 400F - WRITE DISK

Same as above except disregard CY.

#### 4012 - VERIFY DISK

Just like read, but no data is moved, so HL is irrelevant. A good idea after a write (this is what the command DISK WRITE uses.).

### T/S FORMAT

The commands ask for T/S (Track / sector) in the format "ttss" where tt is a hex number for the track and ss is the hex sector.  $0 \leq tt \leq 4c16$   
 $0 \leq ss \leq 1f16$ .

NOTE: These commands do not relieve the necessity of using FIXF.  
Keep a track map of each disk!