CHAPTER 4

DESCRIPTION OF SOFTWARE AND IT'S OPERATION

4.0 INTRODUCTION :

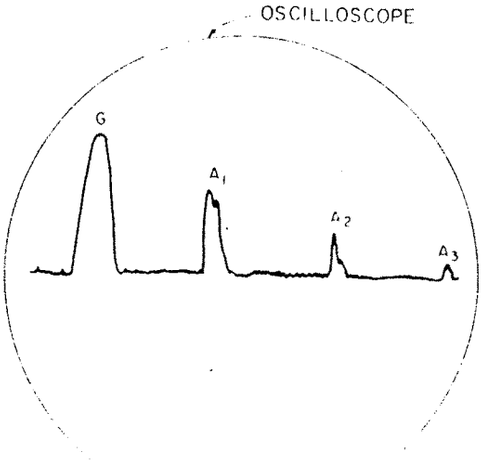
The necessary software for digital data acquisition and processing of ionosonde is developed under higher level language (C) and interfaced with lower level language (ASSEMBLY) under MS-DOS as an operating system. Two programs are developed for :

(i) Special investigation with 'A'/'B' scope in manual mode operation of ionosonde.

(ii) Automatic mode of operation of ionosonde; data storage and construction of ionograms.

4.1 NATURE OF SIGNAL :

The visual display of ionosonde shows the on line panaromic type or differentiated 'A' scope. A signal before differentiation as observed on CRO is shown in fig 4.1.



G : GROUND PULSE, A_1, A_2, A_3 : ONCE, TWICE, THRICE REFLECTED SIGNALS

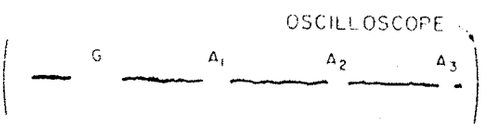
(a) A-SCAN

Fig. 4.1 : Types of ionosonde displays A scan.

Generally three types of display types are used.

A scan : This is illustrated in fig 4.1. The time base of the cathode ray oscilloscope is synchronized with PRF and stationary pulse pattern is obtained. The time base is applied to X plates and receiver output is applied to Y plate of CRO. In our system we have photographed the CRO display. Over the entire range of frequency results are fair but excellent in BAND IV. See photograph no. 1.

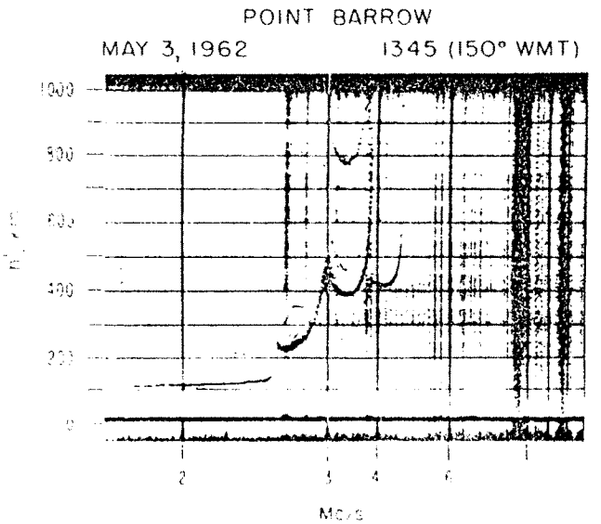
B scan : This is similar to 'A' scan except that the receiver output is applied as blanking pulse of the oscilloscope and Y plates are grounded. The output as shown in fig 4.2.



(b) B-SCAN

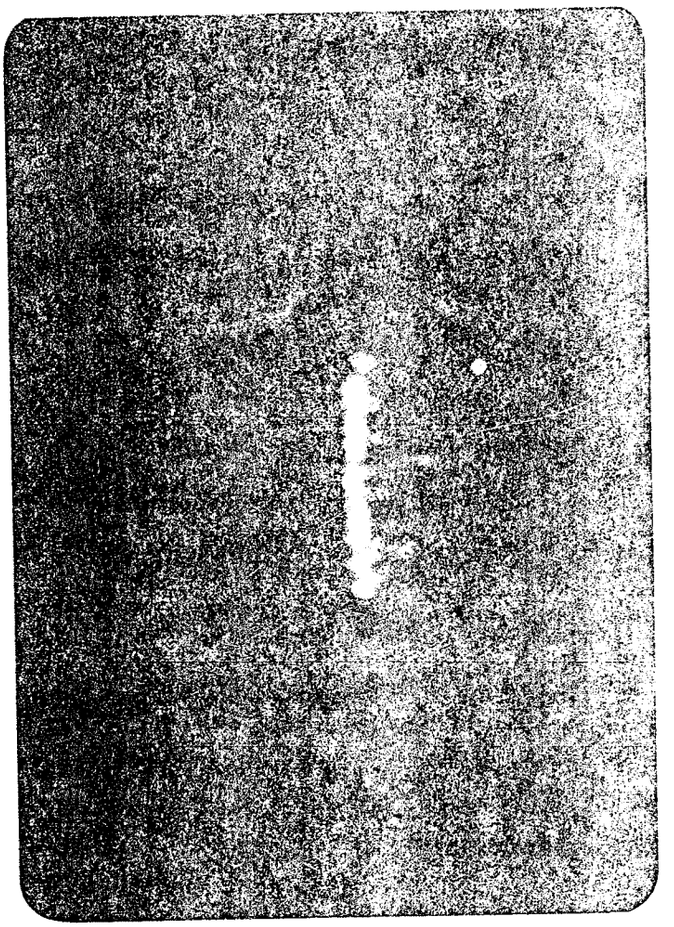
Fig-4·2 *Types of ionosonde display— B scan.*

Panoramic display : The 'B' scan is used with the time base applied to the Y plates and a voltage applied to the X plates which is a function of transmitter frequency. The picture photographed as illustrated in fig 4.3.



(C) PANORAMIC DISPLAY

Fig 4·3 *Types of ionosonde display— panoramic.*



4.2 (A) SPECIAL INVESTIGATION WITH 'A' / 'B' SCOPE:

This program is developed in 'C' language and interfaced with assembly language subroutine. The first part of program is in 'C' in which, initialization of variables, data processing, storage and file operations are carried out. In an assembly subroutine data acquisition is done with high speed (approx. 30,000 samples/sec.). The software is dedicated to manual mode of operation of an ionosonde, operator needs to have the knowledge of ionosonde operation.

Basically the 'A' and 'B' scope are being accessed for the more detailed study purpose. By proper operation of ionosonde, one can very easily get the two above mentioned types of records for finite time (min: 20 to max :234 milliseconds) at fixed frequency.

The parameters required for this program are as follows:

1) Output file name.

2) Time in milliseconds.

3) For printing purpose defaults/ options are menu driven.

SIGNAL CONDITIONS:

1) Input signal to PC-Add on card is in the range 0 to +5 V.

2) Analog to digital conversion with 12 bit resolution.

FEATURES:

A) IBM PC compatible, user friendly software for data acquisition and display.

B) Single channel is used.

C) OFF-line display of data.

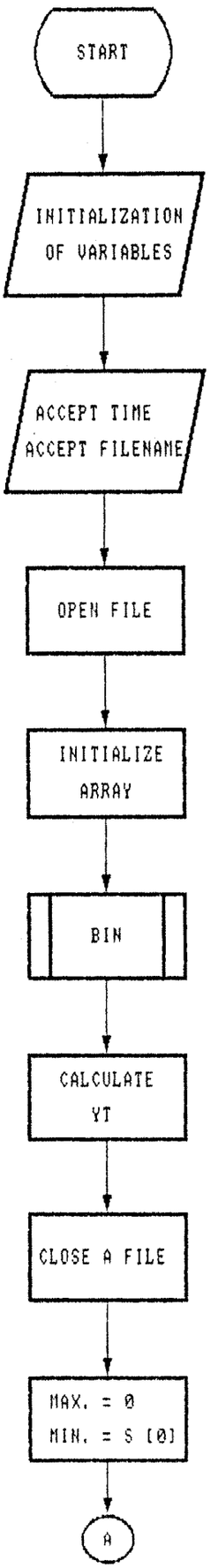
D) The most part of software set up with self explanatory

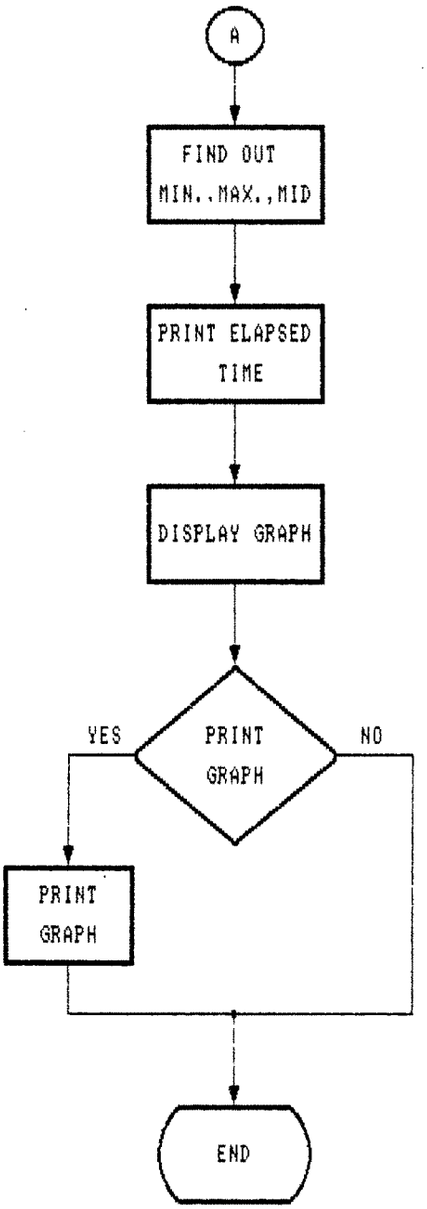
menu.

E) Programs are available in the form of executable file .

FLOWCHARTS AND LISTING OF THE PROGRAM :

The flowcharts and source listing of the program for special investigation is given below in two parts. first part is 'C' language program and second the assembly language programming as a subroutine, of main line program.


START
INITIALIZATION OF VARIABLES
ACCEPT TIME
ACCEPT FILENAME
OPEN FILE
INITIALIZE ARRAY
BIN
CALCULATE YT
CLOSE A FILE
MAX. = 0
MIN. = S [0]
A



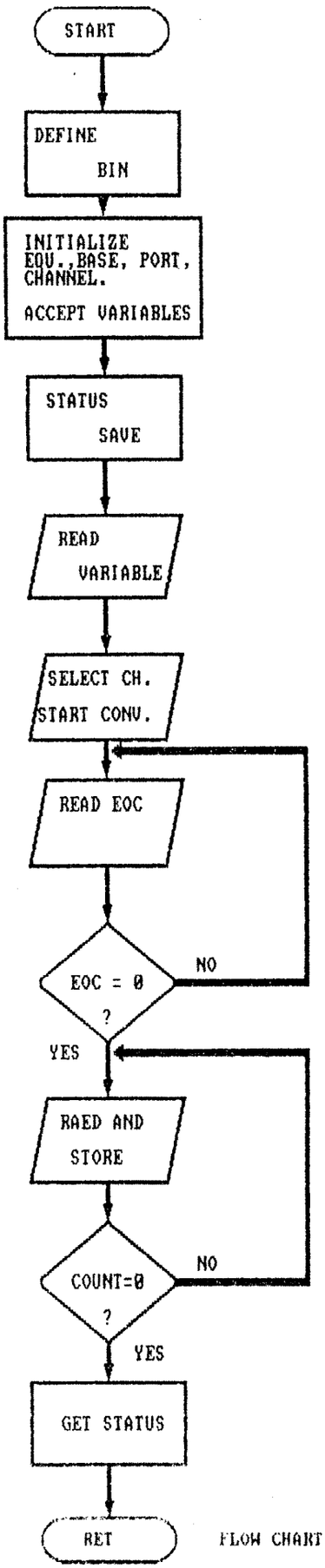
FLOW CHART NO 1(a) cont...

MODULE No.1

```
/* Program to call pointers */
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <graphics.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#include <process.h>

main(int argc, char *argv[])
{
int s[8192];
FILE *fp;
FILE *fopen();
int fclose();
int i,max,min,diff,mid,yt,scr,j,stat,t;
float xt;
char ch,gc;
time_t start,end;
void gotoxy(int x, int y);
int graphdriver = DETECT,graphmode;
printf("Enter Time in miliseconds(min: 20  max: 234)  : ");
scanf("%d",&t);
t = t * 34.48;
fp=fopen(argv[1],"w");
```

```
for (i=0;i<t-1;i++)
   s[i] = 0;
   s[t] = '\0';

start = time(NULL);
printf(ctime(&start));
fprintf(fp, ctime(&start));
BINN(s,t);
printf("%d  %d",sizeof(s),t);
for (i=0;i<t;i++)
{
              yt = s[i];
              xt = ((float) yt/4096) * 350;
              yt = xt;
              fprintf(fp, "%4d %4d\n",abs(i*34.48),yt);
              s[i] = yt;
          }
          fclose(fp);
          max = 0;
          min = s[0];
          for (i=0;i<598;i++)
              {
                if (max < s[i])
                   max = s[i];
                if (min > s[i])
                  min = s[i]; }
          /* printf(" %d   %d   \n",max,min); */
          diff =( max - min)/2;
          mid = min + diff;
          mid = 175 - mid;
          printf(" %d   %d   %d\n",max,min,mid);
          for (i=0;i<t;i++)
               s[i] = s[i]+mid;

          end = time(NULL);
          printf(ctime(&end));
          printf("Elapsed time is %f sec.\n",difftime(end,start));
          getch();
          initgraph(&graphdriver,&graphmode," ");
          rectangle(20,50,630,300);
          i = j = 0;
          for(j=0;j<(t-600);j=j++)
          {
              i = j;
              for (scr =21;scr<630;scr++)
              {
```

```
        putpixel(scr,350 - s[i] ,15);
        i++;
       }
       i = j;
       for(scr = 21; scr<630;scr++)
       {
        putpixel(scr,350 - s[i] ,0);
        i++;
        }
}
 closegraph();
textmode(BW80);
printf("Press  'Y  or y' to print  graph  otherwise  any
other key to exit : ");
gc = getch();
if (gc == 'y' || gc == 'Y')
{     printf("\n\n************PRESS  ANY KEY  FOR  GRAPH
PRINTING**********\n\n");

    getch();
    stat = execl("graphs.exe",NULL);
    printf("execl error = %d\n",stat);
}

}
```



FLOW CHART NO.1(b) SUBROUTINE BINN

MODULE No.1

```
        PUBLIC  _BINN
        NAME BINN
_TEXT           SEGMENT BYTE PUBLIC 'CODE'
                ASSUME CS: _TEXT
BASE    EQU     0220H
PORT    EQU     0225H
CHANNEL         EQU     022AH
PARMA   EQU     4
PRAMB    EQU     6
_BINN           PROC    NEAR
                PUSH    BP
                MOV     BP,SP
                PUSH    DI
                MOV     DI,[BP]+PARMA
                MOV     CX,[BP]+PRAMB
;START CONVERSION
ASD:            MOV AL,000H
                MOV DX,CHANNEL
                OUT DX,AL
                MOV AL,0FFH
                INC DX
                INC DX
                OUT DX,AL
```

```
;TEST OF CONVERSION
                MOV DX,PORT
                LAG:    IN AL,DX
                AND AL,010H
                JNZ LAG
;READ AND STORE
                DEC DX
                IN AL,DX
                MOV AH,AL
                INC DX
                IN AL,DX
                AND AL,0Fh
                MOV DX,00000H
                MOV DL,AL
                MOV AL,AH
                MOV AH,DL
                MOV WORD PTR[DI],AX
                INC DI
                INC DI
                DEC CX
                CMP CX,0000
                JNZ ASD
                POP     DI
                POP     BP
                RET
_BINN           ENDP
_TEXT           ENDS
                END
```

4.3 (B) AUTOMATIC MODE OF OPERATION OF IONOSONDE (SIMULATED SIGNALS FROM SUBSYSTEM):

There are three programs:

i) signal generation using subsystem.

ii) data acquisition

iii) data processing

SIGNAL GENERATION :

Using single board microcomputer the similar to ionoisonde output signals are generated by varying the delay between the first and second pulse. The frames of signals ('A' scope) are derived by using interrupt facility of CPU 8085. A sample program is given below along with flowchart. By calling through a simple program of interrupt initialization the repetative signal is generated through interrupt service routine.

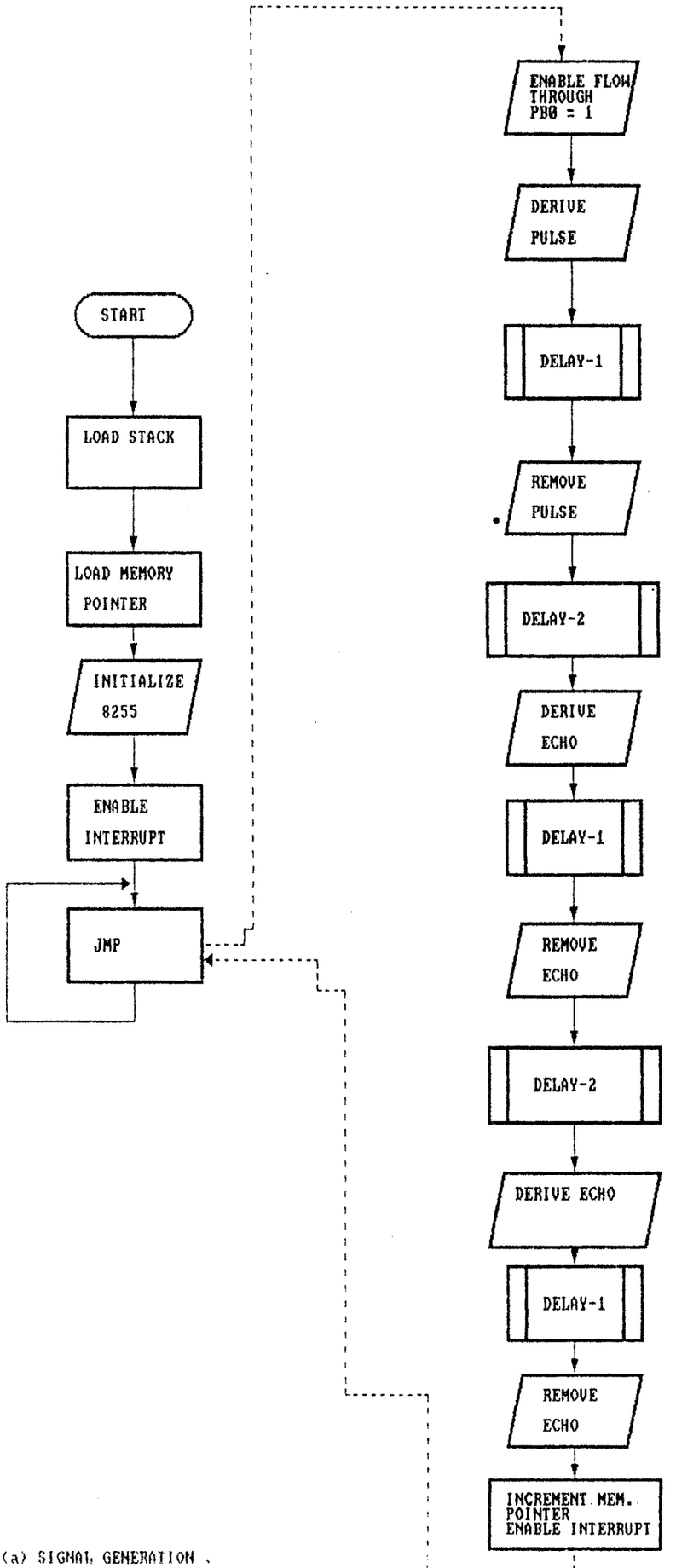
DATA ACQUISITION:

The program is developed in assembly language where initially the initialization of variables segments and file operations are considered. A external trigger pulse which is required to interrupt the subsystem (in case of real ionsonde it is used as a external triggering pulse). Pulse is derived, and incoming 'A' scope frame is grabbed and stored. Thus the triggering is repeated with fixed interval, which will help in calibrating the frequency axis while constructing the ionogram. Since the increase in frequency of ionosonde is either linear or logarithmic with respect to time. For a frame 300 samples of data are collected. In a simulation the number of such frames are also restricted. With the small change in the program it can be increased.

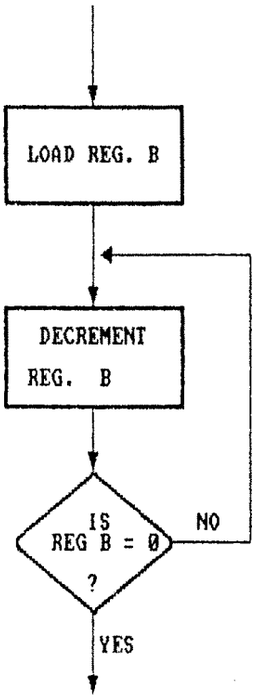
DATA PROCESSING :

In data processing the format of data and in addition identification of ground pulse, echo's and their respective time delayscan be rerecorded. At a time, a frame of 300 samples is processed by comparing the voltage levels with a certain value and for those points which are above a certain voltage level, the time differences is calculated and used as a single record of ionogram. For the defined numbers of frames the program is kept in loop. In data processing, the data file is generated to plot the ionogram.

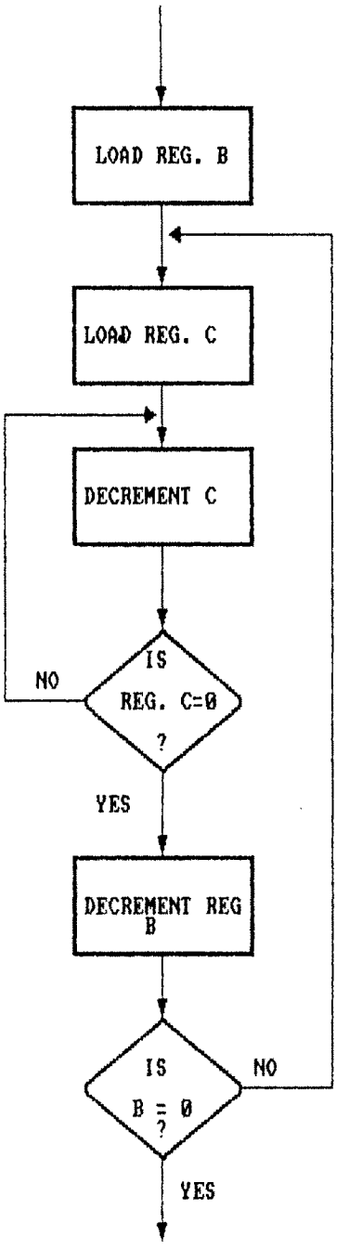
The sequential program flowcharts and listing are given below.



FLOW CHART NO. 2(a) SIGNAL GENERATION .



DELAY-1



DELAY-2

PROGRAM FOR SIGNAL GENERATION ON SINGLE BOARD COMPUTER

A) MAIN PROGRAM:

LABLE	OPCODE	OPERAND	COMMENTS
START:	LXI	SP, FEFF	; LOAD SACK
	LXI	HL, D500	;LOAD MEMORY POINTER
	MVI	A, 89	; INITIALIZE 8255
	OUT	CWR	
	EI		; ENABLE INTERRUPT
	MVI	A, 1D	•; RST 6.5
	SIM		
LOOP1:	JMP	LOOP1	

RST 6.4 **0034**

0034 **JMP ISR**

B) INTERRUPT SERVICE ROUTINE:

LABLE	OPCODE	OPERAND	COMMENTS
	MVI	A, 01	; ENABLE FLOW THROUGH
	OUT	PORT B	
	MVI	A, CC	; OUT GROUND PULSE
	OUT	PORT A	
	CALL	DELAY-1	; PULSE PERIOD
	SUB	A	
	OUT	PORT A	

```
                CALL    DELAY-2         ; HEGHT EQUIVALENT
                                        ; DELAY
                MVI     A, 99           ; FIRST ECHO
                OUT     PORT A
                CALL    DELAY-1
                SUB     A
                OUT     PORT A
                CALL    DELAY-2
                MVI     A, 33             ; SECOND ECHO
                OUT     PORT A
                CALL    DELAY-1
                SUB     A
                OUT     PORT A
                INX     H
                EI
                RET
```

C) DELAY SUBROUTINES:

LABLE	OPCODE	OPERAND	COMMENTS

DELAY-1:

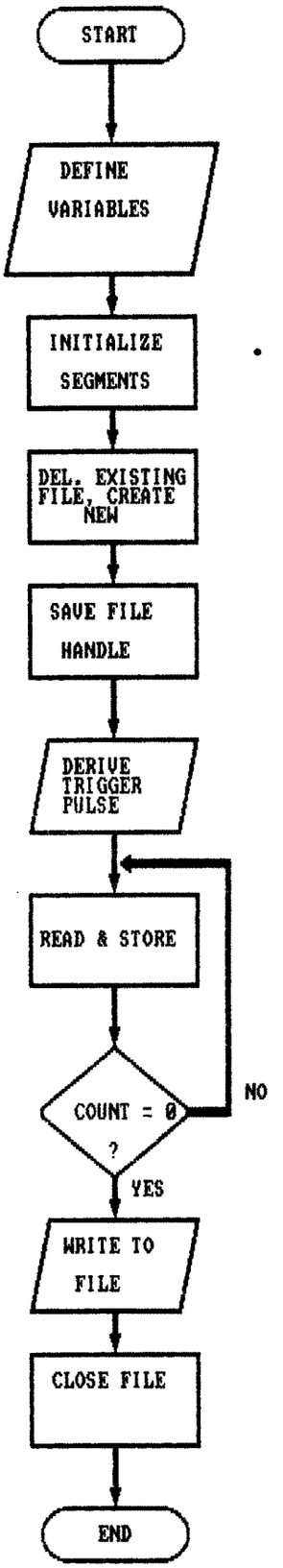
```
                MVI     B, FF
LOOP2:          DCR     B
                JNZ     LOOP2
                RET
```

DELAY-2

```
        MVI     B, 00
        MOV     C, M
LOOP3:  DCX     B
        MOV     A, C
        ORA     B
        JNZ     LOOP3
        RET
```

NOTE:

LOAD VARIOUS DELAY COUNTS IN MEMORY LOCATION DEFINED IN HL PAIR.



FLOW CHART NO.2(B) EXCITATION OF IONOSONDE & DATA ACQUISITION.

ASSEMBLY LANGUAGE PROGRAM FOR EXCITATION OF IONOSONDE

AND DATA ACQUISITION

MODULE 2

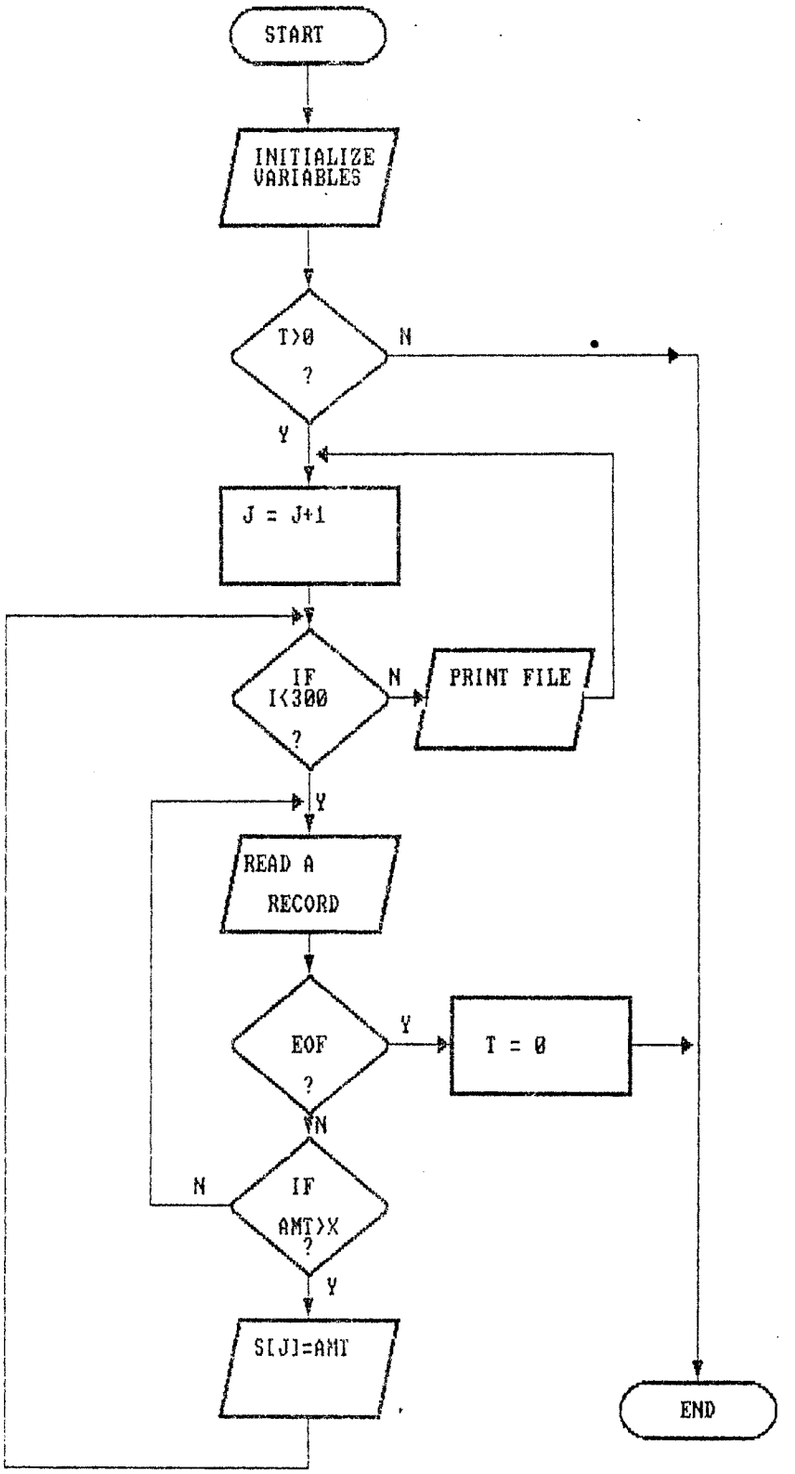
```
;Trial for recording thru pc controlled transmission for
few; records
;Latest succesful program for multiple patches of 300
;records with appending
;     File Creating ,opening & Writing to the FILE .
DATA    SEGMENT
DAT     DB  1200 DUP(0)
NAM     DB   'BUT',0
DATA    ENDS

STACK   SEGMENT STACK
        DB      100 DUP('STACK')
STACK   ENDS

CODE    SEGMENT
        ASSUME CS:CODE,DS:DATA

;INPUT SAMPLES  & STORE IN RAM

BASE    EQU     0220H
PORT    EQU     0225H
CHANNEL         EQU     022AH
;DELETE EXISTING FILE
                MOV AX,DATA
                MOV DS,AX
```

```
                MOV DX,OFFSET NAM
                MOV AH,41h
                INT 21h


;CREATE FILE

                MOV AX,DATA
                MOV DS,AX
                MOV DX,OFFSET NAM  .
                MOV CX,0
                MOV AH,5Bh
                INT 21h
                MOV BX,AX          ;  Save FILE
                                      handle  for  further
                                      use.
                PUSH BX
                PUSH AX
                PUSB BX
                PUSH CX
                PUSH DX
                MOV AL,0FFH        ;PUT A PULSE FOR EXCITING
                                    IONOSONDE
                MOV DX,0224H
                OUT DX,AL
                MOV AL,00FH
                INC DX
                OUT DX,AL
```

```
                MOV CX,000f0H
          AS:   DEC CX
                JNZ AS
                MOV AL,000H
                MOV DX,0224H
                OUT DX,AL
                MOV AL,000H
                INC DX
                OUT DX,AL

;START CONVERSION

ASD1:           MOV CX,300 ;INITIALIZE COUNTER  FOR  300
SAMPLES
                MOV AX,DATA
                MOV DS,AX
                LEA BX,DAT
ASD2:           MOV AL,000H
                MOV DX,CHANNEL
                OUT DX,AL
                MOV AL,0FFH
                INC DX
                INC DX
                OUT DX,AL

;TEST OF CONVERSION
                MOV DX,PORT
        LAG1:   IN AL,DX
                AND AL,010H
```

```
                JNZ LAG1
;READ AND STORE
                DEC DX
                IN AL,DX
                MOV [BX],AL
                INC BX
                INC DX
                IN AL,DX
                AND AL,0Fh
                MOV [BX],AL
                INC BX
                LOOP ASD2

                POP DX
                POP CX
                POP BX
                POP AX

;WRITE TO THE FILE

                MOV AX,DATA
                 MOV DS,AX
                LEA DX,DAT
                MOV CX,600
                MOV AH,40h
                INT 21h

;CLOSE THE FILE
                POP BX
```

```
                MOV  AH,3Eh
                INT 21h

;TERMINATE PROGRAM

                MOV AL,0
                MOV AH,4Ch
                INT 21h

CODE            ENDS
                END
```


START
INITIALIZE VARIABLES
T>0 ?
N
Y
J = J+1
IF I<300 ?
N
PRINT FILE
Y
READ A RECORD
EOF ?
Y
T = 0
N
IF AMT>X ?
N
Y
S[J]=AMT
END

MODULE No. 2

```
#include <io.h>

#include <stdio.h>

#include <dos.h>

#include <conio.h>

#include <stdlib.h>

#include <process.h>

#include <fcntrl.h>

#include <fcntl.h>

main()

{int i,amt,No(u),t,j;
     int t=1;
 printf("ENTER THE VALUE");
 scanf(&amt);

FILE *fp,*fp1;


    fp=fopen("but.dat","r");

    fp1=fopen("but1.dat","w");

while (t>0)
      { j = j+1;
         for (i = 0; i< 300; i+1)
              { x= getw(fp);
                if(x = EOF)
                     {t = 0;
                          break;
                      }
                if (t>0 ) && (x>amt)
                     No(i) = amt;
                     if (t = 0);
                       break;
                }
fprintf("%d %t %d %t %d %t %d %t %d",i, s[0],s[1],s[2],s[3]);
          }
   fclose(fp);
   fclose(fp1);
     }
```

REFERENCES

1) MASTERING TURBO C
BY STAIN KENNLY-BOOTLE
B.P.B.
2) ASSEMBLY LANGUAGE PROGRAMMING
BY QUE
QUE PUBLICATIONS
3) IBM PC ASSEMBLY LANGUAGE PROGRAMMING
BY MILLER
B.P.B.
4) PROGRAMMING C!
PRINCIPLES AND PROGRAMMING
BY TIM GRADY
Mc-GRAW HILL