BIBILOGRAPHY

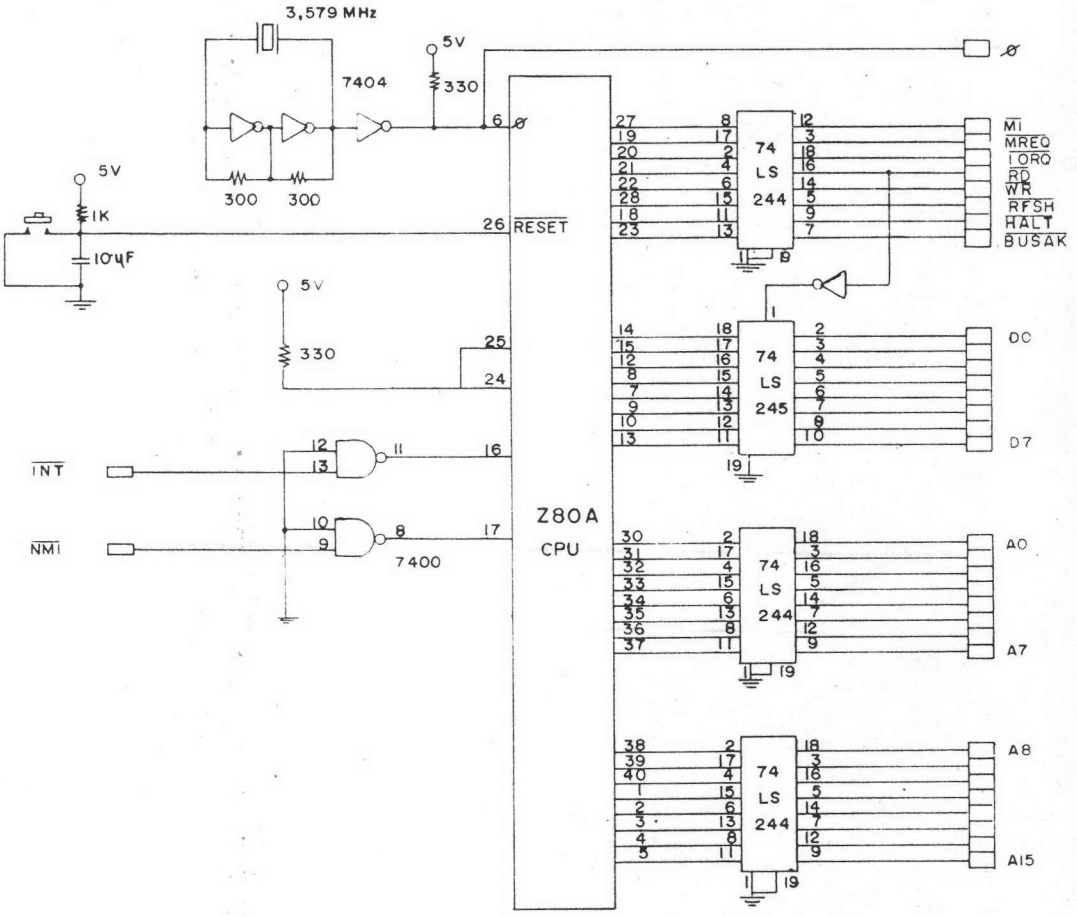
1. Gordon B. Davis. Computer Data Processing. 2nd Edition: Mc Graw-Hill Book Company.

2. Steve Ciarcia. Byte. Vol.7. No.3. March 1982.

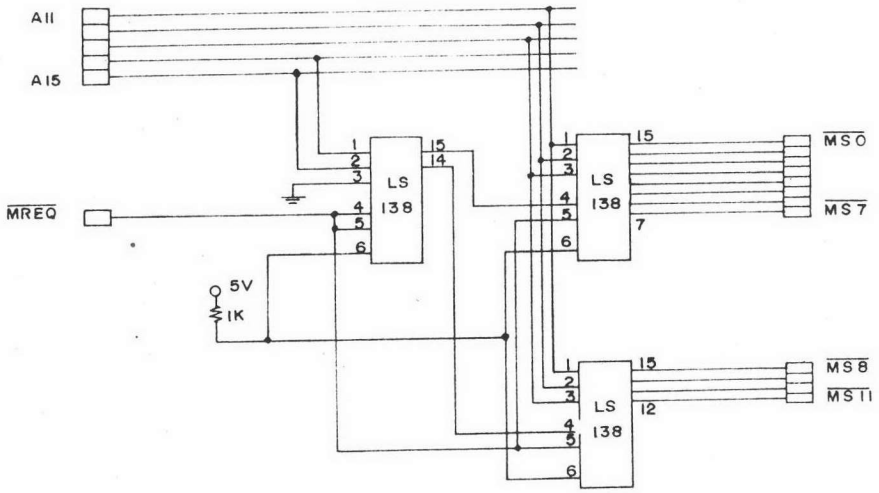
3. Lawrence Rabiner. Digital Processing of Speech Signal.: Prentice-Hall. 1978.

4. N. Rex Dixon and Thomas B. Martin. Automatic Speech & Speaker Recognition. : IEEE PRESS.

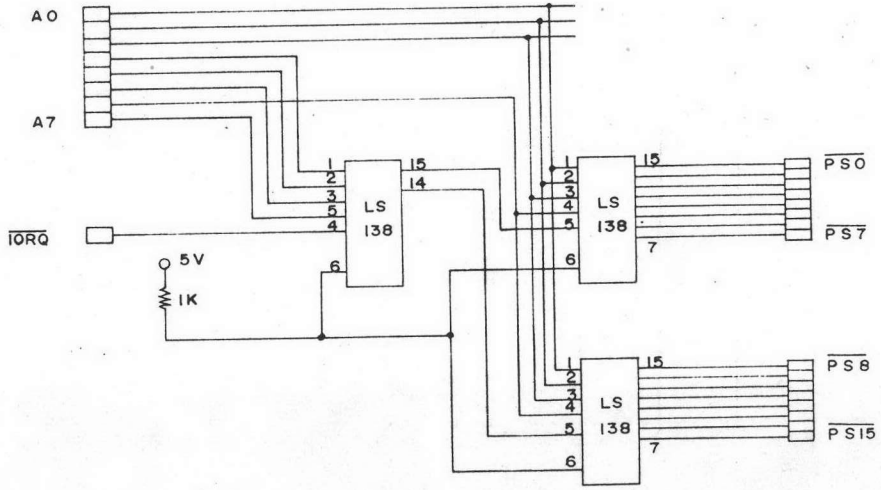
ภาคผนวก ก.
แผนผังวงจรโดยละเอียด


3,579 MHz
7404
5V
330
300
300
1K
10μF
RESET
Z80A
CPU
7400
INT
NMI
74 LS 244
74 LS 245
MI
MREQ
IORQ
RD
WR
RFSH
HALT
BUSAK
D0
D7
A0
A7
A8
A15

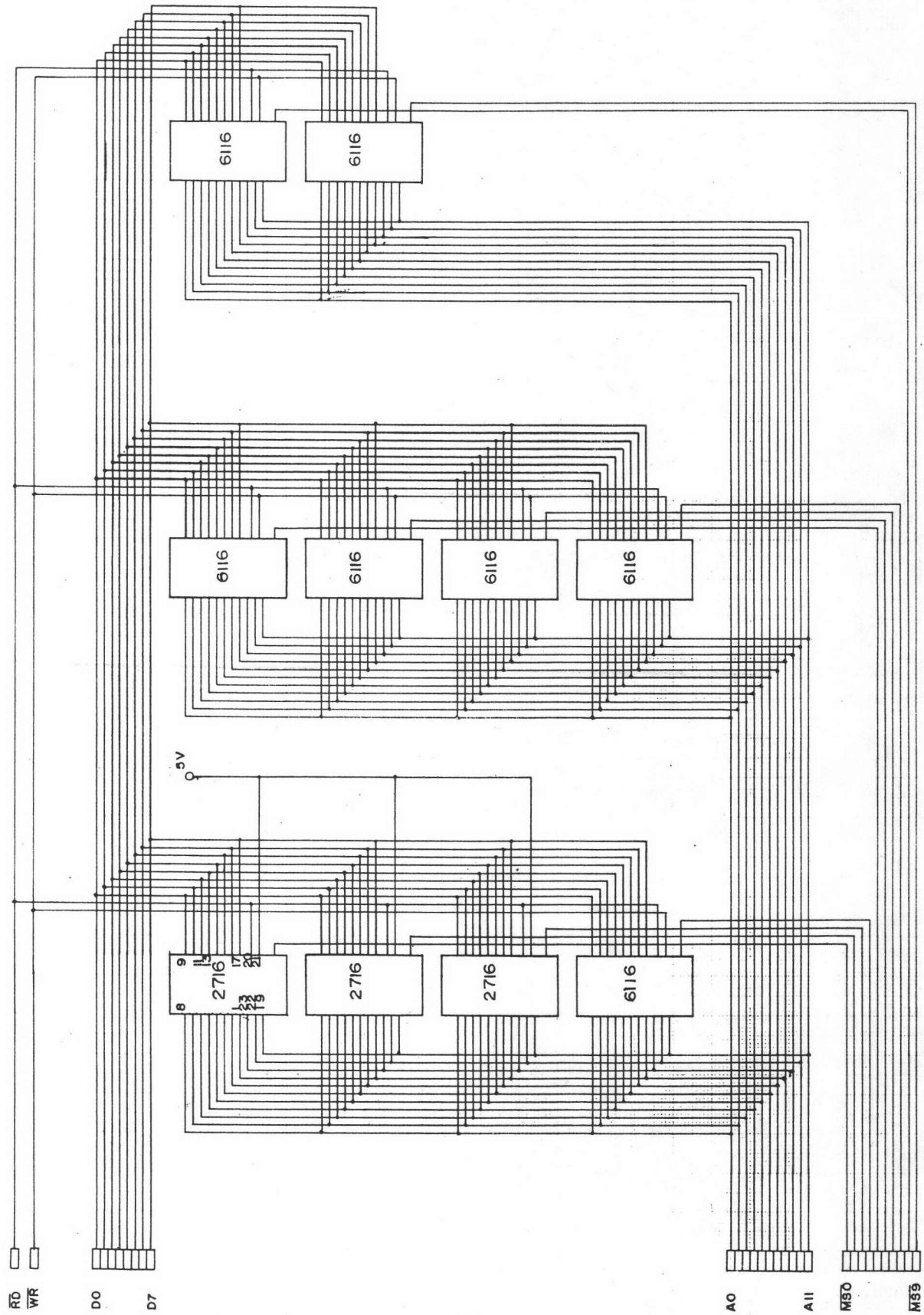
CPU



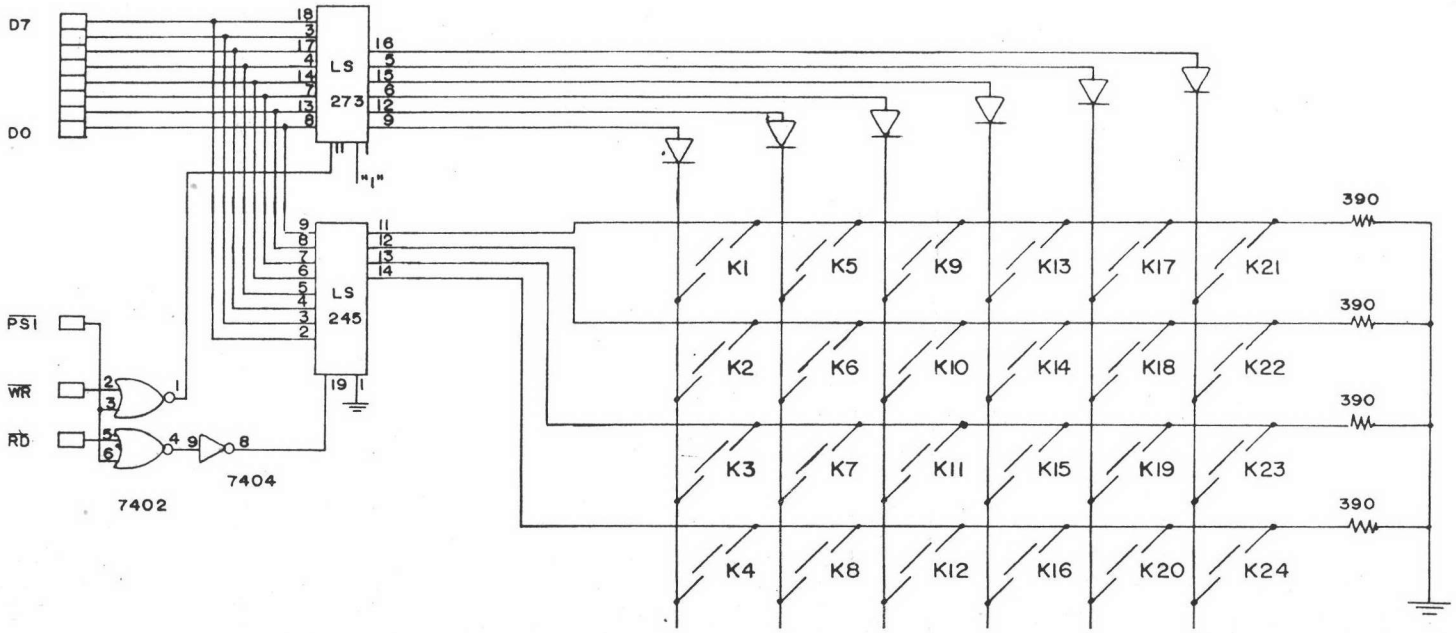
MEMORY DECODER



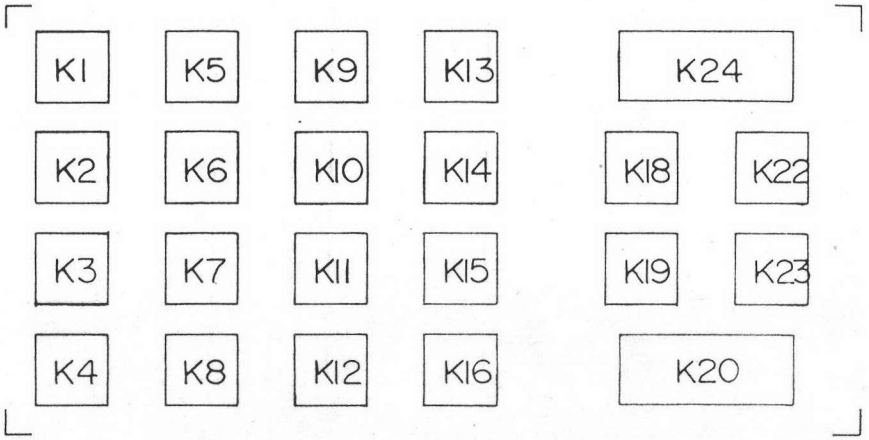
DEVICE DECODER

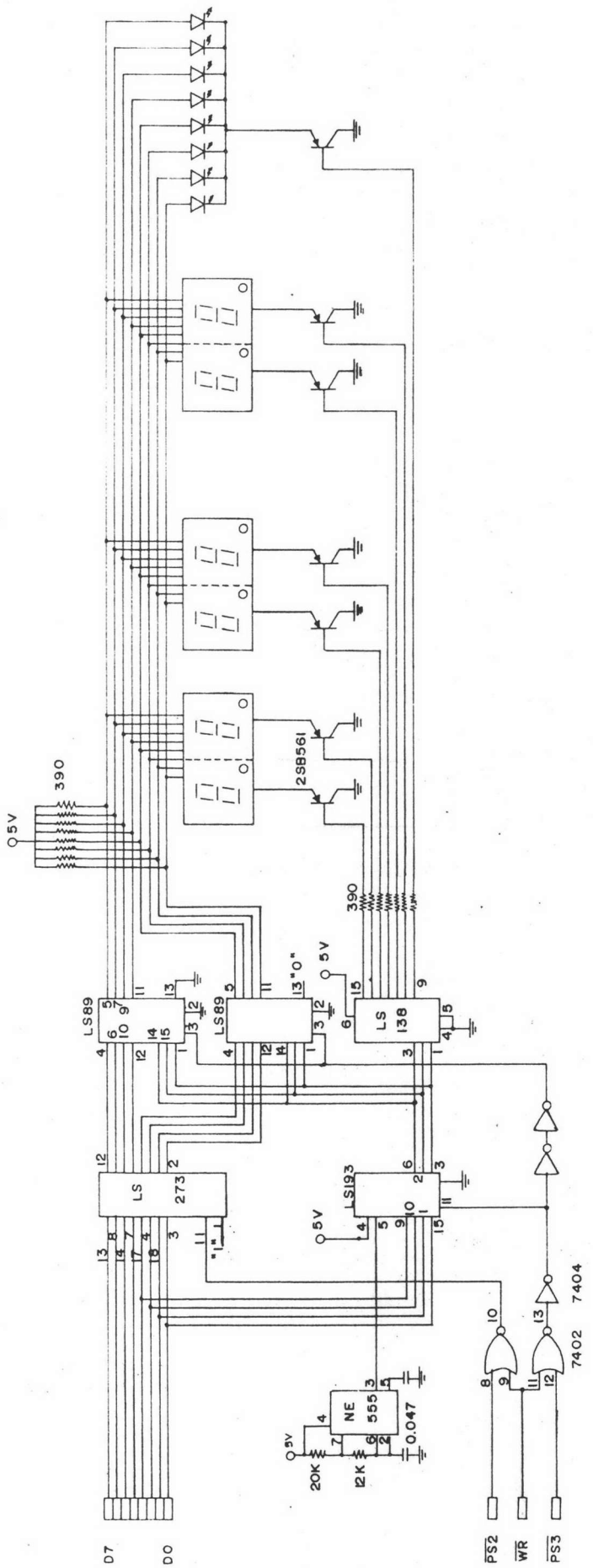


MEMORY (ROM & RAM)

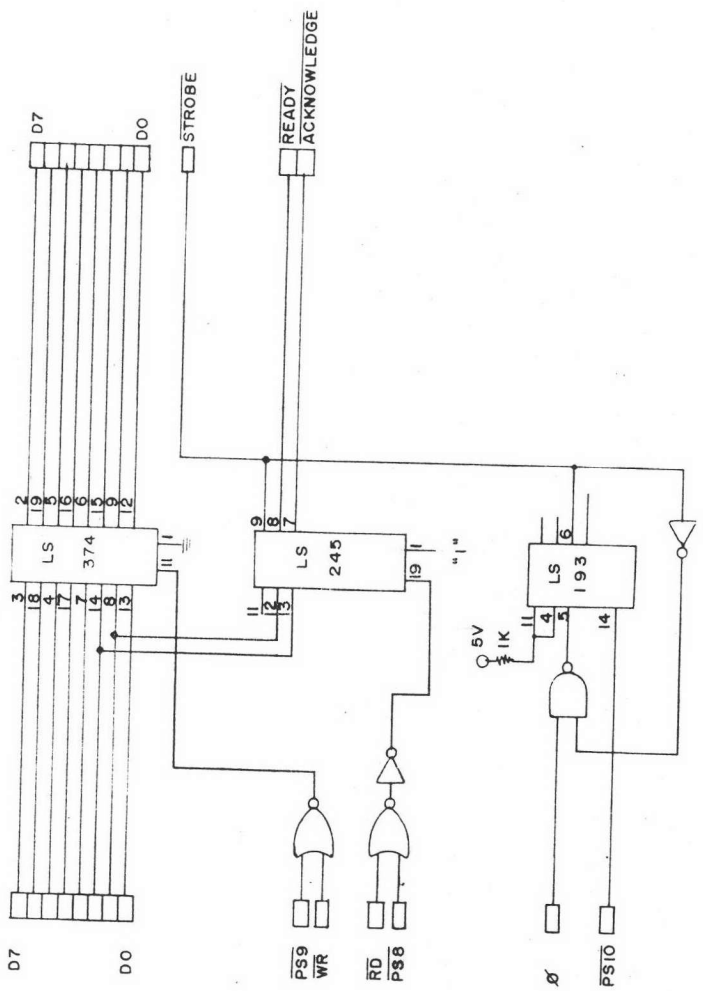


KEY BOARD & KEY BOARD LAYOUT

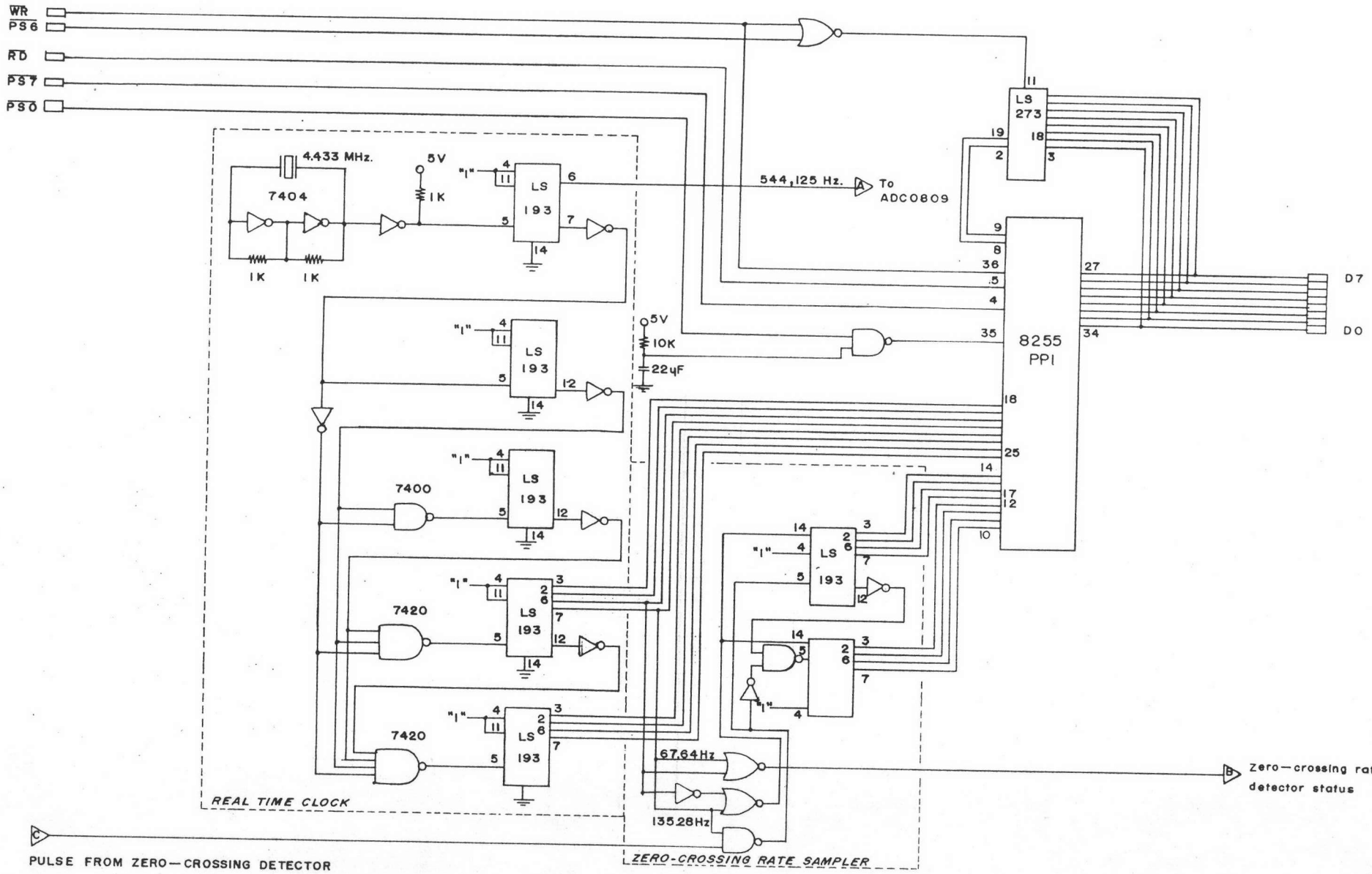


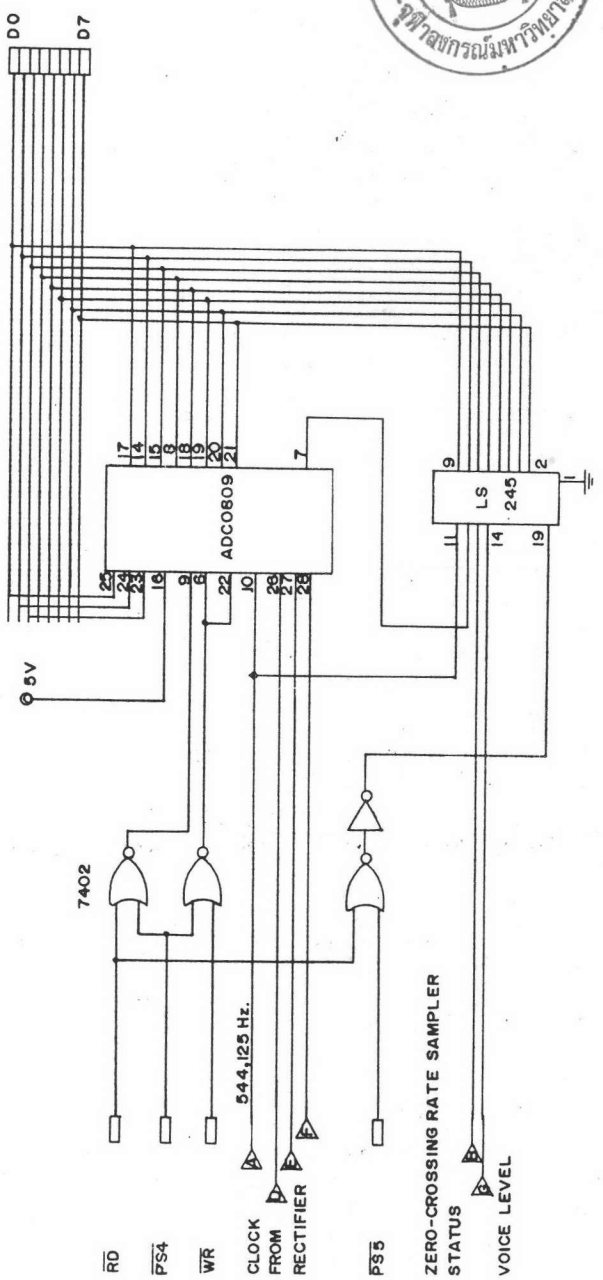


DISPLAY UNIT

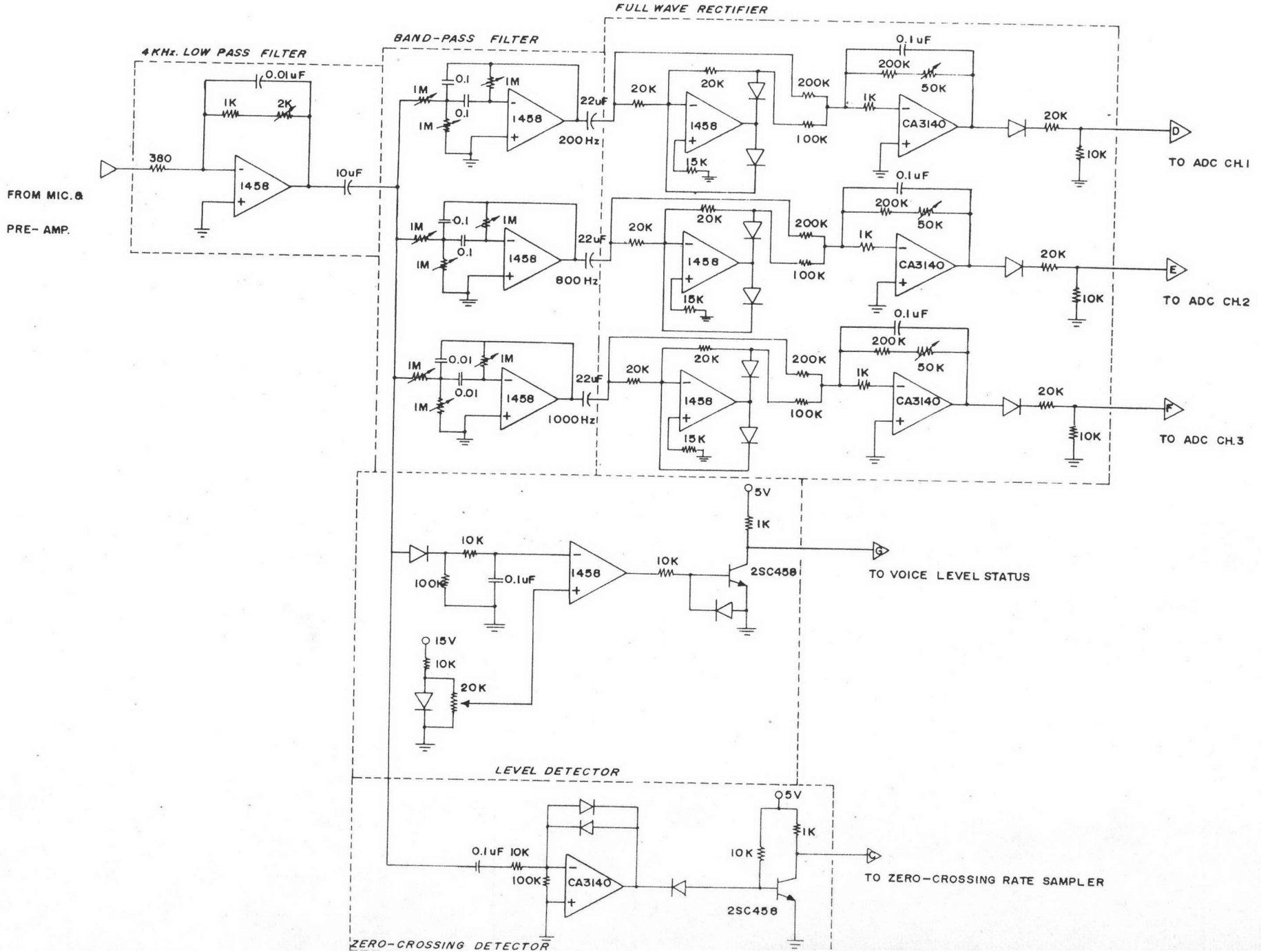


PRINTER INTERFACE


WR
PS6
RD
PS7
PS0
4.433 MHz.
7404
1K
1K
5V
1K
"1"
LS
193
544,125 Hz.
To
ADC0809
LS
273
8255
PPI
D7
D0
5V
10K
22μF
7400
7420
7420
REAL TIME CLOCK
67.64Hz
135.28Hz
ZERO-CROSSING RATE SAMPLER
Zero—crossing rate
detector status
PULSE FROM ZERO—CROSSING DETECTOR



ANALOG TO DIGITAL CONVERTER


4KHz. LOW PASS FILTER
BAND-PASS FILTER
FULL WAVE RECTIFIER
FROM MIC.& PRE-AMP.
380
0.01uF
1K
2K
1458
10uF
1M
0.1
22uF
200Hz
800Hz
1000Hz
0.01
20K
200K
100K
15K
50K
0.1uF
CA3140
10K
TO ADC CH.1
TO ADC CH.2
TO ADC CH.3
5V
1K
2SC458
TO VOICE LEVEL STATUS
15V
LEVEL DETECTOR
ZERO-CROSSING DETECTOR
TO ZERO-CROSSING RATE SAMPLER

ภาคผนวก ข.

รายละเอียดโปรแกรมควบคุม

PROGRAM NAME = ZMON.Z80 PAGE 0001

```
0001    ;MONITOR PROGRAM FOR Z80 CPU SINGLE BOARD
0002    ;DEVELOP BY SUPONGSE PEKANAN
0003    ;
0004    ZMON    ORG     0000H
0005            JP      START   ;START ROUTINE
0006            ORG     0100H
0007    START   LD      HL,0FFFFH  ;SET END OF ACCESSIBLE ADDR
0008    FSTAK   LD      A,(HL)  ;FIND HIGH AVAILABLE ADDR
0009            LD      B,A
0010            CPL
0011            LD      (HL),A
0012            CP      (HL)    ;CHECK IF RAM
0013            JR      Z,EXIT  ;IF FOUND THEN EXIT
0014            LD      A,H
0015            SUB     01H     ;DECREMENT BY 256 BYTES
0016            JP      Z,ERROR ;IF TOP OF MEM THEN ERROR
0017            LD      H,A
0018            JR      FSTAK   ;DO WHILE NOT END IF MEM.
0019    EXIT    LD      (HL),B  ;RETURN MEM. CONTENT
0020            INC     HL
0021            LD      SP,HL   ;SET STACK POINTER
0022            DEC     HL
0023    FHADDR  LD      A,(HL)  ;FIND LOW ADDRESS.
0024            LD      B,A
0025            CPL
0026            LD      (HL),A
0027            CP      (HL)    ;CHECK IF RAM
0028            JR      NZ,EX1  ;IF NOT THEN CHECK IF ROM
0029            LD      (HL),B  ;RETURN MEM CONTENT
0030            DEC     HL
0031            LD      A,H
0032            OR      L
0033            JR      NZ,FHADDR  ;DO WHILE NOT END OF ADDRESS
0034            JR      CONT    ;
0035    EX1     LD      A,B
0036            CP      (HL)    ;CHECK IF ROM
0037            JP      NZ,ERROR ;IF NOT THEN SHOW RAM ERROR STATUS
0038            INC     HL
0039    CONT    EX      DE,HL   ;SET DE AS LOCATION COUNTER
0040    RUTN    LD      A,(DE)
0041            LD      C,A     ;SET REG. C AS ACC.
0042            CALL    DSPLY
0043            CALL    KYBD
0044            JR      RUTN    ;DO WHILE TRUE
0045    DSPLY   LD      A,C     ;DISPLAY MEM. CONTENT,MEM ADDR
0046            LD      B,05H   ;SET HIGH DIGIT
0047            CALL    DSPL1
0048            LD      A,E
0049            CALL    DSPL1 ;DISPLAY (L)
0050            LD      A,D
```

```
0051            CALL    DSPL1 ;DISPLAY (H)
0052            RET             ;END SUBROUTINE
0053    DSPL1   PUSH    AF      ;SAVE A
0054            CALL    TRANS
0055            OUT     (02H),A
0056            LD      A,B
0057            OUT     (03H),A ;LATCH DIGIT
0058            DEC     B       ;DECREMENT DIGIT
0059            POP     AF
0060            SRL     A
0061            SRL     A
0062            SRL     A
0063            SRL     A
0064            CALL    TRANS ;TRANSFORM HIGH HALF BYTE
0065            OUT     (02H),A ;LATCH DATA
0066            LD      A,B
0067            OUT     (03H),A ;LATCH DIGIT
0068            DEC     B
0069            RET
0070    TRANS   AND     0FH     ;TRANSFORM TO DISPLAY CHR.
0071            LD      HL,DTAB
0072            PUSH    DE
0073            LD      D,00H
0074            LD      E,A
0075            ADD     HL,DE
0076            POP     DE
0077            LD      A,(HL)
0078            CPL
0079            RET
0080    KYBD    PUSH    DE
0081            PUSH    BC
0082    KWAIT   CALL    SCAN1   ;CALL SCAN KEYBOARD SUBROUTINE
0083            JR      Z,KWAIT ;WAIT UNTILL A KEY IS PRESSED
0084            DEC     A
0085            LD      D,00H
0086            LD      E,A
0087            LD      HL,KTABLE ;KEY BOARD IN MONITOR MODE
0088            ADD     HL,DE
0089            LD      A,(HL)  ;READ MEANING
0090            POP     BC
0091            POP     DE
0092            CP      'H'     ;COMPARE IF 'H' COMMAND
0093            JR      NZ,KYBD1
0094            LD      D,C
0095            JR      FUNC
0096    KYBD1   CP      'L'     ;........'L' COMMAND
0097            JR      NZ,KYBD2
0098            LD      E,C     ;SET LOW ADDR
0099            JR      FUNC
0100    KYBD2   CP      'S'     ;CHECK IF STORE
```

```
0101            JR      NZ,KYBD3
0102            LD      A,C
0103            LD      (DE),A
0104            INC     DE
0105            JR      FUNC
0106    KYBD3   CP      'B'     ;CHECK IF BACKSPACE
0107            JR      NZ,KYBD4
0108            DEC     DE
0109            JR      FUNC
0110    KYBD4   CP      'F'     ;CHECK IF FORWARD
0111            JR      NZ,KYBD5
0112            INC     DE
0113            JR      FUNC
0114    KYBD5   CP      'G'     ;CHECK IF EXC
0115            JR      NZ,KYBD6
0116    RELS    CALL    KDELAY
0117            CALL    SCAN1   ;WAIT FOR RELEASE KEY
0118            JR      NZ,RELS
0119            EX      DE,HL
0120            JP      (HL)
0121    KYBD6   CP      0FFH    ;CHECK IF NON-DEFINED IS PRESSED
0122            JR      Z,KYBD ;IF YES THEN WAIT
0123            LD      B,A     ;SAVE KEY
0124            LD      A,C
0125            ADD     A
0126            ADD     A
0127            ADD     A
0128            ADD     A       ;SHIFT LEFT 4 BIT
0129            OR      B
0130            LD      C,A     ;RETURN TO IMAGE ACC.
0131            CALL    DSPLY
0132            JR      RELS1
0133    FUNC    LD      A,(DE)
0134            LD      C,A
0135            CALL    DSPLY
0136    RELS1   CALL    KDELAY
0137            CALL    SCAN1   ;WAIT FOR KEY DEPRESSED
0138            JR      NZ,RELS1
0139            JR      KYBD
0140    ERROR   LD      B,05H   ;ERROR STATUS
0141    ELOP1   LD      A,00H
0142            OUT     (02H),A
0143            LD      A,B
0144            OUT     (03H),A
0145            DJNZ    ELOP1
0146            HALT
0147    KDELAY  PUSH    HL
0148            LD      HL,04FFH
0149    KDLP    DEC     HL
0150            LD      A,H
```

```
0151            OR      L
0152            JR      NZ,KDLP
0153            POP     HL
0154            RET             ;END DELAY LOOP
0155            DB      00H,00H
0156    DTAB    DB      3FH,06H,5BH,4FH
0157            DB      66H,6DH,7DH,07H
0158            DB      7FH,6FH,77H,7CH
0159            DB      58H,5EH,79H,71H
0160            DB      00H
0161    KTABLE  DB      07H,04H,01H,00H
0162            DB      08H,05H,02H,0AH
0163            DB      09H,06H,03H,0BH
0164            DB      0FH,0EH,0DH,0CH
0165            DB      0FFH,'B','H','S'
0166            DB      0FFH,'F','L','G'
0167    SCAN1   EQU     02ABH   ;KEY BOARD SCAN SUBROUTINE
0168    FIN     ORG     04FFH
0169            DB      0FFH
0170            END     100H
```

PROGRAM NAME = SUBR.Z80 PAGE 0001

```
0001    ;DISPLAY 1 DIGIT SUBROUTINE
0002    ;DISPLAY CONTENT IN ACC.
0003    ;DIGIT TO BE DISPLAY INDICATE BY REG B
0004            ORG     0220H
0005    DSP1    PUSH    BC
0006            PUSH    AF
0007            LD      C,03H
0008            OUT     (02H),A
0009            OUT     (C),B
0010            POP     AF
0011            POP     BC
0012            RET
0013    ;DISPLAY STRING SUBROUTINE
0014    ;DISPLAY MESSAGE LOCATION POINTED BY REG HL
0015    ;NO. OF DIGIT TO BE DISPLAY COUNT BY REG B
0016    ;START DIGIT INDICATE BY ACC.
0017    DSP2    PUSH    HL
0018            PUSH    BC
0019            PUSH    AF
0020            LD      C,02H
0021    LOP1    OUTI
0022            OUT     (03H),A
0023            JP      Z,RTN1
0024            INC     A
0025            JP      LOP1
0026    RTN1    POP     AF
0027            POP     BC
0028            POP     HL
0029            RET
0030    ;SUBROUTINE CONVERT HEX TO ASCII
0031    ;SOURCE POINTED BY REG HL
0032    ;DESTINATION ...BY REG DE
0033    ;NO. OF BYTE TO BE CONVERTED COUNT BY REG BC
0034    CONV1   PUSH    HL
0035            PUSH    DE
0036            PUSH    BC
0037            PUSH    AF
0038    LOP2    LD      A,(HL)
0039            SRL     A
0040            SRL     A
0041            SRL     A
0042            SRL     A
0043            CALL    CONV2
0044            LD      (DE),A
0045            INC     DE
0046            LD      A,(HL)
0047            AND     0FH
0048            CALL    CONV2
0049            LD      (DE),A
0050            INC     DE
```

```
0051                INC     HL
0052                DEC     BC
0053                LD      A,C
0054                OR      B
0055                JP      NZ,LOP2
0056                POP     AF
0057                POP     BC
0058                POP     DE
0059                POP     HL
0060                RET
0061    CONV2       ADD     30H
0062                CP      3AH
0063                RET     C
0064                ADD     07H
0065                RET
0066    ;SUBROUTINE CONVERT ASCII TO DISPLAY CHAR.
0067    ;SOURCE POINTED BY HL
0068    ;DESTINATION ... BY DE
0069    ;NO. OF BYTE ... BY BC
0070    CONV4       PUSH    HL
0071                PUSH    DE
0072                PUSH    BC
0073                PUSH    AF
0074    LOP3        LD      A,(HL)
0075                CP      3AH
0076                JP      C,CONV5
0077                SUB     37H
0078    CONV5       CALL    CONV6
0079                LD      (DE),A
0080                INC     DE
0081                INC     HL
0082                DEC     BC
0083                LD      A,C
0084                OR      B
0085                JP      NZ,LOP3
0086                POP     AF
0087                POP     BC
0088                POP     DE
0089                POP     HL
0090                RET
0091    CONV6       PUSH    HL
0092                PUSH    DE
0093                AND     0FH
0094                LD      D,00H
0095                LD      E,A
0096                LD      HL,TABLE
0097                ADD     HL,DE
0098                LD      A,(HL)
0099                POP     DE
0100                POP     HL
```

```
0101                RET
0102                DB      00H
0103    TABLE       DB      0C0H,0F9H,0A4H,0B0H
0104                DB      099H,092H,082H,0F8H
0105                DB      080H,090H,088H,083H
0106                DB      0A7H,0A1H,086H,08EH
0107                DB      00H
0108    ;KEYBOARD SCAN SUBROUTINE
0109    ;RETURN KEY POSITION IN ACC.
0110    ;IF NON-KEY IS PRESSED THEN RETURN ZERO FLAG
0111    SCAN1       PUSH    BC
0112                LD      A,01H
0113                LD      B,01H
0114    SCAN2       CALL    SCAN3
0115                JP      C,RTN2
0116                SLA     A
0117                JP      Z,RTN2
0118                INC     B
0119                JP      SCAN2
0120    RTN2        POP     BC
0121                RET
0122    SCAN3       AND     A           ;CLEAR CARRY FLAG
0123                PUSH    AF
0124                OUT     (01H),A
0125                IN      A,(01H)
0126                OR      A
0127                JP      NZ,SCAN5
0128    SCAN4       POP     AF
0129                RET
0130    SCAN5       CALL    DELAY
0131                POP     AF
0132                PUSH    AF
0133                OUT     (01H),A
0134                IN      A,(01H)
0135                OR      A
0136                JP      Z,SCAN4
0137                LD      C,00H
0138    LOP4        INC     C
0139                AND     A
0140                RRA
0141                JP      NC,LOP4
0142                AND     A
0143                JP      NZ,SCAN4
0144                DEC     B
0145                SLA     B
0146                SLA     B
0147                LD      A,B
0148                ADD     C
0149                SCF
0150                POP     BC
```

```
0151              RET
0152    DELAY     PUSH    BC
0153              LD      B,06H
0154    LOP5      LD      C,0FFH
0155    LOP6      DEC     C
0156              JP      NZ,LOP6
0157              DEC     B
0158              JP      NZ,LOP5
0159              POP     BC
0160              RET
0161    ;SUBROUTINE SEARCH TABLE
0162    ;IX = TABLE ADDRESS , IY = RECORD ADDRESS
0163    ;HL = TABLE ENTRY LENGTH
0164    ;DE = END OF TABLE ADDRESS
0165    ;B  = KEY LENGTH
0166    ;RETURN ZERO FLAG IF FOUND, RETURN TABLE ADDR IN HL
0167    SEARCH  PUSH    IX
0168            PUSH    IY
0169            PUSH    BC
0170    CHECK   LD      A,(IY)
0171            CP      A,(IX)
0172            JP      NZ,CONT
0173            INC     IX
0174            INC     IY
0175            DJNZ    CHECK
0176            POP     BC
0177            POP     IY
0178            POP     HL
0179            LD      A,00H
0180            OR      A
0181            RET
0182    CONT    POP     BC
0183            POP     IY
0184            POP     IX
0185            PUSH    HL
0186            PUSH    BC
0187            PUSH    IX
0188            POP     BC
0189            ADD     HL,BC
0190            POP     BC
0191            PUSH    HL
0192            POP     IX
0193            AND     A
0194            SBC     HL,DE
0195            POP     HL
0196            JP      C,SEARCH
0197            LD      A,0FFH
0198            AND     A
0199            RET
0200    ;SYSTEM INTERFACE SUBROUTINE
```

PROGRAM NAME = SUBR. Z80 PAGE 0005

```
0201    ;ANALOG CHANNEL = AI0-AI2
0202    ;CHANNEL :
0203    ;          PORT 04 = ADC COMMUNICATION CONTROL CH.
0204    ;          PORT 05 = INTERFACE STATUS CHANNEL.
0205    ;                    BIT 0 = ADC CLOCK.
0206    ;                    BIT 1 = ADC STATUS.
0207    ;                            0 = BUSY: 1 = IDLE OR READY.
0208    ;                    BIT 2 = ZERO-CROSSING SAMPLING UNIT STATUS.
0209    ;                            0 = BUSY: 1 = READY.
0210    ;                    BIT 3 = SIGNAL LEVEL.
0211    ;                            0 = SIGNAL : 1 = NO SIGNAL ****
0212    ;          PORT 06 = 8255 CONTROL CHANNEL.
0213    ;                    BIT 0 - BIT 1 :
0214    ;                                    00 = REQUEST SUB-CH. A.
0215    ;                                    01 = ..... B.
0216    ;                                    10 = ..... C.
0217    ;                                    11 = REQUEST 8255 CONTROL.
0218    ;          PORT 07 = 8255 COMMUNICATION CHANNEL.
0219    ;                    SUBCHANNEL A = SPARE.
0220    ;                    SUBCHANNLE B = REAL TIME CLOCK.
0221    ;                    SUBCHANNEL C = ZERO CROSSING RATE.
0222    ;================================================================
0223    ;SUBROUTINE READ ANALOG TO DIGITAL DATA
0224    ;           ANALOG CHANNEL = AI0 TO AI2
0225    ;           DATA WILL BE STORED AT LOC. (HL) - (HL)+2
0226    ;BEGIN:
0227    RDADC   PUSH    HL
0228            PUSH    AF
0229            LD      A,02H    ;SET ANALOG CHANNEL
0230    READ    OUT     (04H),A  ;TRIG ADC UNIT
0231            PUSH    AF       ;SAVE CHANNEL COUNTER
0232    WAIT1   IN      A,(05H)  ;READ ADC STATUS
0233            BIT     1,A      ;CHECK ADC STATUS
0234            JR      NZ,WAIT1         ;WAIT FOR ADC START
0235    WAIT2   IN      A,(05H)  ;READ ADC STATUS
0236            BIT     1,A      ;CHECK ADC STATUS
0237            JR      Z,WAIT2  ;WAIT FOR ADC COMPLETE
0238            IN      A,(04H)  ;READ DATA
0239            LD      (HL),A   ;STORE
0240            INC     HL
0241            POP     AF       ;RETURN ANALOG CHANNEL COUNTER
0242            SUB     01H
0243            JR      NC,READ  ;READ UNTILL END
0244            POP     AF
0245            POP     HL
0246            RET
0247    ;END SUBROUTINE
0248    ;SUBROUTINE SET 8255
0249    ;MODE CONTROL  1001 1011 = 9BH : ALL SUB CHANNEL = INPUT
0250    SET8    PUSH    AF
```

```
0251            OUT     (00H),A ;CLEAR 8255
0252            LD      A,03H   ;SET TO 8255 CONTROL
0253            OUT     (06H),A
0254            LD      A,9BH   ;LOAD 8255 CONTROL MODE
0255            OUT     (07H),A
0256            XOR     A
0257            OUT     (06H),A ;RESET COMMUNICATION CH.
0258            POP     AF
0259            RET
0260    ;END SUBROUTINE
0261    ;SUBROUTINE READ REAL TIME CLOCK
0262    ;*****CLOCK STORED AREA MUST BE ASSIGNED IN HL.
0263    ;       (HL) = PRESENT CLOCK.
0264    ;     (HL)+1 = LAST CLOCK.
0265    ; ALSO RETURN CLOCK IN ACC.
0266    ;
0267    RDCLK   PUSH    HL
0268            PUSH    BC
0269            LD      A,01H
0270            OUT     (06H),A ;SET COMMUNICATE TO SUB-CH. C.
0271            IN      A,(07H) ;READ CLOCK
0272            LD      B,(HL)
0273            LD      (HL),A  ;STORE CLOCK
0274            INC     HL
0275            LD      (HL),B  ;STORE LAST CLOCK
0276            POP     BC
0277            POP     HL
0278            RET
0279    ;END SUBROUTINE
0280    ;SUBROUTINE READ ZERO-CROSSING RATE
0281    ;RESULT WILL BE STORE IN (HL)
0282    ;
0283    RDZERO  LD      A,02H
0284            OUT     (06H),A ;SET COMM. TO SUB-CH. C.
0285    WAIT3   IN      A,(05H) ;READ STATUS.
0286            BIT     2,A
0287            JR      Z,WAIT3 ;WAIT FOR READY
0288            IN      A,(07H) ;READ Z-RATE
0289            LD      (HL),A  ;STORE RESULT
0290            AND     A
0291            RET
0292    ;END SUBROUTINE
0293    ;SUBROUTINE EXAMINE VOICE INPUT
0294    ;RETURN NON-ZERO FLAG IF VOICE.
0295    EXVC    IN      A,(05H) ;READ STATUS
0296            CPL
0297            BIT     3,A
0298            CPL
0299            RET
0300    ;END SUBROUTINE
```

```
0301    ;SUBROUTINE MULTIPLY
0302    ;MULTIPLIER IS INPUT IN DE
0303    ;MULTIPLICAND IS IN BC
0304    ;UNSIGNED VALUE******
0305    ;4 BYTE RESULT PASSED IN D,E,H,L
0306    MULT16  LD      A,16
0307            LD      HL,0
0308    MLOOP   BIT     7,D
0309            JR      Z,JUMP1
0310            ADD     HL,BC
0311            JR      NC,JUMP1
0312            INC     DE
0313    JUMP1   DEC     A
0314            RET     Z       ;***** RETURN FROM SUBROUTINE
0315            EX      DE,HL
0316            ADD     HL,HL   ;SHIFT DE CONTENT
0317            EX      DE,HL
0318            ADD     HL,HL   ;SHIFT HL CONTENT
0319            JR      NC,MLOOP
0320            INC     DE      ;IF B15 OF HL = 1 THEN ADD TO DE
0321            JR      MLOOP
0322    ;END MULTIPLY SUBROUTINE
0323    ;SUBROUTINE DEVIDE 16 BIT
0324    ;32 BIT DIVIDENED IS INPUT IN H,L,D,E
0325    ;DEVISOR 16 BIT IN B,C
0326    ;QUOTIENT IN DE . REMAINDER IN HL
0327    ;
0328    DVDE16  LD      A,16
0329    DVLOP   ADD     HL,HL   ;SHIFT HL
0330            EX      DE,HL
0331            ADD     HL,HL   ;SHIFT DE
0332            EX      DE,HL
0333            JR      NC,JUMP2
0334            INC     HL
0335    JUMP2   OR      A
0336            SBC     HL,BC
0337            INC     DE
0338            JP      P,JUMP3
0339            ADD     HL,BC
0340            RES     0,E
0341    JUMP3   DEC     A
0342            JR      NZ,DVLOP
0343            RET
0344    ;END SUBROUTINE
0345    ;SUBROUTINE CLEAR MEMORY
0346    ;START ADDRESS IN HL
0347    ;NO. OF BYTE TO BE CLEAR IN BC
0348    ;PADDING CHARACTER IN A
0349    ;
0350    PADD    PUSH    HL
```

```
0351            PUSH    DE
0352            PUSH    BC
0353            PUSH    AF
0354            PUSH    BC
0355            POP     DE
0356            LD      BC,0001H
0357    CLR     LD      (HL),A
0358            INC     HL          ;INCREMENT MEM. ADDR.
0359            AND     A           ;CLEAR CARRY FLAG
0360            EX      DE,HL
0361            SBC     HL,BC
0362            EX      DE,HL
0363            JR      NC,CLR
0364            POP     AF
0365            POP     BC
0366            POP     DE
0367            POP     HL
0368            RET
0369    ;END SUBROUTINE
0370    ;SUBROUTINE INTERFACE EPSON PRINTER
0371    ;A CHARACTER TO BE WRITTEN TO EPSON
0372    ;FILL IN ACCUMULATOR
0373    ;
0374    EPSEND  PUSH    AF
0375    EPWAIT  IN      A,(08H) ;READ PRINTER STATUS
0376            AND     06H     ;MASK AND CHECK
0377            XOR     04H     ;BIT 1 IF 0 = READY
0378            JR      NZ,EPWAIT       ;BIT 2 = ACKNOWLEDGE
0379            POP     AF      ;WAIT UNTILL READY
0380            OUT     (09H),A ;SEND CHARACTER TO DATA PORT
0381            OUT     (0AH),A ;TRIG EPSON DATA STROBE
0382            RET
0383    ;END SUBROUTINE
0384    ;
0385    ;SUBROUTINE SEND A STRING FROM DATA BUFFER
0386    ;TO EPSON PRINTER
0387    ;START ADDRESS IN REG. HL
0388    ;END   ADDRESS IN REG. DE
0389    ;
0390    EPSON   PUSH    HL
0391            PUSH    DE
0392    EP00    LD      A,(HL)  ;GET CHARACTER
0393            CALL    EPSEND
0394            INC     HL      ;INCREMENT POINTER
0395            PUSH    HL
0396            AND     A       ;CLEAR CARRY FLAG
0397            SBC     HL,DE   ;CHECK ADDRESS
0398            POP     HL
0399            JR      C,EP00  ;IF NOT END-OF-DATA CONT.
0400            POP     DE
```

PROGRAM NAME = SUBR. Z80

```
0401                POP     HL
0402                RET
0403    ;END SUBROUTINE
0404    ;SUBROUTINE DUMP MEMORY
0405    ;START ADDRESS IN HL
0406    ;END   ADDRESS IN DE
0407    ;BUFFER ADDRSS IN BC
0408    ;ADDRESS/LINE  IN ACC.
0409    ;
0410    DUMP    PUSH    HL
0411            PUSH    DE
0412            PUSH    BC
0413            EXX
0414            POP     BC          ;COPY CONTENT TO REG'
0415            POP     DE
0416            POP     HL
0417            EX      AF,AF'      ;SAVE BYTE COUNTER/LINE
0418            PUSH    HL          ;PRINT HEAD
0419            PUSH    DE
0420            LD      HL,HEAD
0421            LD      DE,EHEAD
0422            CALL    EPSON
0423            POP     DE
0424            POP     HL
0425    DP00    PUSH    HL          ;ADDRESS COUNTER NOW IN
0426            PUSH    DE          ;REG.  PAIR HL
0427            PUSH    BC          ;WRITE ADDRESS ROUTINE BEGIN
0428            EX      DE,HL
0429            PUSH    BC
0430            POP     HL          ;COPY BUFFER ADDRESS TO HL.
0431            LD      (HL),D      ;STORE ADDRESS COUNTER
0432            INC     HL
0433            LD      (HL),E
0434            INC     HL
0435            EX      DE,HL
0436            PUSH    BC
0437            POP     HL
0438            LD      BC,02       ;CONVERT 2 BYTE HEX TO ASCII
0439            CALL    CONV1
0440            EX      DE,HL
0441            PUSH    HL
0442            POP     DE
0443            INC     DE
0444            INC     DE
0445            INC     DE
0446            INC     DE
0447            CALL    EPSON
0448            LD      A,20H
0449            CALL    EPSEND
0450            POP     BC
```

PROGRAM NAME = SUBR.Z80 PAGE 0010

```
0451            POP     DE
0452            POP     HL
0453            PUSH    DE
0454            PUSH    BC
0455            PUSH    BC
0456            POP     DE
0457            EX      AF,AF'
0458            LD      C,A
0459            EX      AF,AF'
0460            LD      B,0
0461            CALL    CONV1
0462            ADD     HL,BC
0463    DP01    LD      A,(DE)
0464            INC     DE
0465            CALL    EPSEND
0466            LD      A,(DE)
0467            INC     DE
0468            CALL    EPSEND
0469            LD      A,20H
0470            CALL    EPSEND
0471            DEC     BC
0472            LD      A,C
0473            OR      B
0474            JR      NZ,DP01
0475            POP     BC
0476            POP     DE
0477            LD      A,0DH
0478            CALL    EPSEND
0479            AND     A
0480            PUSH    HL
0481            SBC     HL,DE
0482            POP     HL
0483            JR      C,DP00
0484            EXX
0485            LD      A,0AH
0486            CALL    EPSEND
0487            EX      AF,AF'
0488            RET
0489            DB      00H
0490    HEAD    DB      'MONITOR DUMP UTILITY'
0491            DB      '                    '
0492            DB      'VERSION 1.0   BY '
0493            DB      'SUPONG  PEKANAN'
0494            DB      0DH
0495    EHEAD   DB      0AH
0496            DB      00H
0497            DB      0DH,0AH
0498    ;END SUBROUTINE
0499    ;
0500    ;SUBROUTINE SCREEN DATA
```

```
0501    ;DATA FORMAT :RECORD LENGTH = 6 BYTE
0502    ;END OF FILE = FF
0503    ;SOURCE IN REG.HL
0504    ;DESTINATION IN DE
0505    ;RETURN END OF DESTINATION ADDRESS IN BC
0506    ;DESTINATION RECORD FORMAT 5 BYTE
0507    ;BYTE NO. 1 = TIME
0508    ;........ 2 = ENVELOP OF BPF-3
0509    ;........ 3 = ------------  2
0510    ;........ 4 = ------------  1
0511    ;........ 5 = EOF STATUS
0512    ;DATA WILL BE CLIPPING BY 8
0513    ;
0514    SCREEN  PUSH    HL
0515            PUSH    DE
0516            PUSH    HL
0517            INC     HL      ;READ TIME REF.
0518            LD      B,(HL)  ;STORE IN REG. B
0519            LD      C,0     ;CLEAR TIME COUNTER
0520            POP     HL
0521    SCR00   INC     HL      ;IGNORE Z-CROSSING
0522            LD      A,(HL)  ;READ TIME
0523            SUB     B       ;FIND REL.TIME DIFF.
0524            LD      B,(HL)  ;RESTORE TIME REF.
0525            SRL     A
0526            SRL     A
0527            SRL     A
0528            SRL     A       ;SHIFT RIGHT 4 BIT
0529            ADD     C       ;TIME COUNT=*+TIME DIFF.
0530            LD      C,A     ;KEEP TRACK TIME COUNT
0531            LD      (DE),A  ;STORE TIME COUNT
0532            PUSH    BC      ;
0533            LD      B,3     ;CLIPPING NEXT 3 (HL) REC.
0534    SCR01   INC     DE      ;INCREMENT STORE AREA
0535            INC     HL      ;INCREMENT SOURCE PTR.
0536            LD      A,(HL)
0537            SUB     08H     ;**** CLIPPING BY 8/256
0538            JR      NC,SCR02
0539            XOR     A       ;IF NEGATIVE THEN FORCE 0
0540    SCR02   LD      (DE),A
0541            DJNZ    SCR01   ;REPEAT UNTILL END
0542            POP     BC
0543            INC     HL
0544            INC     DE
0545            LD      A,(HL)  ;READ FILE STATUS
0546            LD      (DE),A  ;STORE
0547            INC     HL
0548            INC     DE
0549            CP      0FFH    ;CHECK EOF
0550            JR      NZ,SCR00        ;DO UNTILL EOF
```

```
0551            DEC     DE
0552            PUSH    DE
0553            POP     BC
0554            POP     DE
0555            POP     HL
0556            RET
0557    ;END SUBROUTINE
0558    ;subroutine 1st pass trace data
0559    ;function :find difference of
0560    ;           height of voice envelop
0561    ;           from each band-pass filter
0562    ;input record format
0563    ;+ from screen subprogram
0564    ;byte no. 1 = relative time
0565    ;-------- 2 = value of bpf. at the time
0566    ;-------- 3 = ------------2----------
0567    ;-------- 4 = ------------1----------
0568    ;-------- 5 = record status 00:continue;ff=end
0569    ;
0570    ;output record format :
0571    ;byte no. 1 = time difference of sampling point
0572    ;-------- 2 = diff. in bpf.1 = ( Xn+1 - Xn )
0573    ;-------- 3 = ------------2---------------
0574    ;-------- 4 = ------------3---------------
0575    ;-------- 5 = record status
0576    ;
0577    ;source data pointer assigned in reg. HL
0578    ;destination pointer ..... DE
0579    ;end address of destination return in BC
0580    ;
0581    VELOCTY PUSH    HL
0582            PUSH    DE      ;save register
0583            LD      BC,05H  ;set alternate pointer to next record
0584            PUSH    HL
0585            ADD     HL,BC
0586            PUSH    HL
0587            EXX             ;SET HL' AS ALTERNATE POINTER
0588            POP     HL
0589            EXX
0590            POP     HL      ;RETURN POINTER
0591    VEL00   LD      B,4     ;SET PROCESSING LOOP
0592    VEL01   EXX             ;USE ALTERNATE POINTER
0593            LD      A,(HL)  ;LOAD n+1 th DATA
0594            INC     HL      ;INCREMENT POINTER
0595            EXX             ;SET TO NORMAL POINTER
0596            SUB     (HL)    ;Dn+1 - Dn
0597            INC     HL
0598            LD      (DE),A  ;STORE RESULT
0599            INC     DE
0600            DJNZ    VEL01   ;DO WHILE NOT END OF RECORD
```

```
0601                INC     HL          ;SKIP RECORD STATUS OF Rn
0602                EXX                 ;EXAMINE Rn+1
0603                LD      A,(HL)      ;RETRIEVE RECORD STATUS
0604                INC     HL
0605                EXX
0606                LD      (DE),A      ;STORE RECORD STATUS
0607                INC     DE
0608                CP      0FFH        ;CHECK IF EOF
0609                JR      NZ,VEL00            ;DO WHILE NOT EOF
0610                DEC     DE          ;RETURN END DESTINATION ADDRESS
0611                PUSH    DE
0612                POP     BC
0613                POP     DE
0614                POP     HL
0615                RET                 ;END SUBROUTINE*********
0616    ;SUBROUTINE INSERT BYTE
0617    ;INSERT POINT SPECIFIED IN REG.HL
0618    ;END ADDRESS OF INSERT AREA IN DE
0619    ;NO. OF SPACE BYTE REQUIRED IN REG.B
0620    ;PADDING CHARACTOR IN A.
0621    ;
0622    ;
0623    INSERT  PUSH    HL
0624            PUSH    DE
0625            PUSH    BC
0626            PUSH    AF
0627            LD      C,B
0628            LD      B,00H
0629            DEC     BC
0630            ADD     HL,BC       ;SKIP ADDRESS
0631            PUSH    DE
0632            PUSH    DE          ;SAVE DE CONTENT
0633            EX      DE,HL
0634            AND     A           ;CLEAR CARRY FLAG
0635            SBC     HL,DE       ;FIND NO. OF ITERATION
0636            EX      DE,HL
0637            POP     HL          ;LOAD END ADDRESS TO HL
0638            INC     BC          ;FIND LAST BYTE
0639            AND     A
0640            SBC     HL,BC
0641            PUSH    DE
0642            POP     BC          ;LOAD BYTE COUNTER TO BC
0643            POP     DE
0644            LDDR                ;MOVE BYTE
0645            POP     AF
0646            POP     BC
0647            PUSH    BC
0648    INS01   LD      (DE),A      ;PADDING LOOP
0649            DEC     DE
0650            DJNZ    INS01
```

```
0651            POP     BC
0652            POP     DE
0653            POP     HL
0654            RET
0655    ;END SUBROUTINE
0656    ;
0657    ;SUBROUTINE DELETE BYTE
0658    ;DELETE POINT IN HL, END DELETE AREA IN DE
0659    ;NO. OF BYTE IN B
0660    ;PADDING CHARACTER IN A
0661    ;
0662    DELETE  PUSH    HL
0663            PUSH    DE
0664            PUSH    BC
0665            PUSH    AF
0666            LD      C,B
0667            LD      B,0
0668            PUSH    HL
0669            ADD     HL,BC   ;SET START LOCATION FOR LDIR
0670            EX      DE,HL   ;FIND NO. OF BYTE
0671            AND     A
0672            SBC     HL,DE
0673            PUSH    HL
0674            POP     BC
0675            INC     BC      ;SET NO. OF BYTE IN BC
0676            EX      DE,HL
0677            POP     DE
0678            LDIR            ;MOVE BYTE
0679            POP     AF
0680            POP     BC
0681            POP     DE
0682            PUSH    DE
0683            PUSH    BC
0684            PUSH    AF
0685    DELOO   LD      (DE),A  ;PADDING CHR. LOOP
0686            DEC     DE
0687            DJNZ    DELOO
0688            POP     AF
0689            POP     BC
0690            POP     DE
0691            POP     HL
0692            RET
0693    ;END SUBROUTINE
0694    ;SUBROUTINE SEARCH PARTITION
0695    ;VALUE TO BE SEARCH GIVE IN ACC.
0696    ;TABLE ADDRESS IN HL
0697    ;RESULT: RETURN ADDRESS OF TABLE RECORD IN HL
0698    ;AND RECORD COUNT IN B
0699    ;IF ASSIGNED VALUE FALL WITH IN ANY BOUNDARY
0700    ;THEN BIT 7 OF ACC. IS ON
```

```
0701    ;IF OVERLAP ON ONE BOUNDARY OCCURS
0702    ;BIT 6 OF ACC IS ON
0703    ;ELSE IF OVERLAP ON 2 PARTITION THEN
0704    ;BIT 7 TO BIT 5 IS ON
0705    ;ELSE ACC IS SET TO 0
0706    ;
0707    SPART   PUSH    BC
0708            PUSH    DE
0709            PUSH    AF
0710            ADD     HALFBND ;SET UPPER BOUND
0711            LD      C,A
0712            POP     AF
0713            SUB     HALFBND ;SET LOWER BOUND
0714    HALFBND EQU     0
0715            LD      DE,8
0716            ADD     HL,DE   ;TABLE-ADDR = *+8
0717            EXX             ;USE ALTERNATE REG.PAIR
0718            LD      B,1
0719            EXX             ;SET COUNT = 1
0720            EX      AF,AF'
0721            XOR     A       ;CLEAR FLAG
0722            EX      AF,AF'
0723            CALL    CKEOT+DF ;CHECK END OF TABLE
0724    SPAR1   JR      C,SPTEND ;IF EOT THEN END
0725            EX      AF,AF'
0726            AND     A
0727            JR      NZ,SPEOD ;IF ANY FLAG SET THEN END
0728            EX      AF,AF'
0729            PUSH    HL      ;SAVE RECORD ADDR.
0730            LD      A,C     ;CHECK UPPERBOUND : LB(TADDR)
0731            SUB     (HL)
0732            JR      NC,SPAR2 ;IF => THEN DO;
0733            EX      AF,AF'  ;SET DISABLE="TRUE"
0734            SET     0,A
0735            EX      AF,AF'
0736            AND     A
0737            POP     HL
0738            JR      SPAR1   ;RETURN TO DO WHILE LOOP
0739    SPAR2   JR      NZ,SPAR3 ;ELSE IF = THEN DO;
0740            EX      AF,AF'
0741            SET     7,A
0742            SET     6,A     ;FOUND,OVERLAP1="TRUE"
0743            EX      AF,AF'
0744            AND     A
0745            POP     HL
0746            JR      SPAR1
0747    SPAR3   INC     HL      ;IF > THEN DO;
0748            LD      A,(HL)  ;CHECK UPPER BOUND(TADDR),LB
0749            SUB     B
0750            JR      NC,SPAR4 ;IF < THEN DO;
```

```
0751            POP     HL
0752            INC     HL
0753            INC     HL
0754            EXX             ;ADDR=NEXT (ADDR)
0755            INC     B       ;COUNT=COUNT+1
0756            EXX
0757            CALL    CKEOT+DF ;CHECK END OF TABLE
0758            JR      SPAR1
0759    SPAR4   EX      AF,AF'
0760            SET     7,A     ;FOUND="TRUE"
0761            EX      AF,AF'
0762            LD      A,(HL)  ;CHECK UP,UP(TADDR)
0763            SUB     C
0764            JR      C,SPAR5 ;IF => THEN DO;
0765            POP     HL
0766            JR      SPAR1
0767    SPAR5   POP     HL      ;RETURN CURRENT ADDR
0768            PUSH    HL      ;SAVE ADDR
0769            INC     HL
0770            INC     HL      ;TADDR=NEXT(TADDR)
0771            LD      A,(HL)
0772            CP      0FFH    ;CHECK EOT
0773            JR      Z,SPAR6 ;IF NOT EOT THEN DO;
0774            LD      A,C
0775            SUB     (HL)    ;CHECK UP,UP(TADDR)
0776            JR      NC,SPAR7 ;IF => THEN DO;
0777    SPAR6   EX      AF,AF'
0778            SET     6,A     ;OVERLAP="TRUE"
0779            EX      AF,AF'
0780            POP     HL
0781            AND     A
0782            JR      SPAR1
0783    SPAR7   EX      AF,AF'
0784            SET     6,A     ;OVERLAP2="TRUE"
0785            EX      AF,AF'
0786            POP     HL
0787            AND     A
0788            JR      SPAR1
0789    SPEOD   EX      AF,AF'  ;RETURN TO NORMAL AF
0790    SPTEND  EXX             ;RETURN COUNT TO B
0791            LD      A,B
0792            EXX
0793            POP     DE
0794            POP     BC
0795            LD      B,A
0796            EX      AF,AF'
0797            PUSH    AF
0798            EX      AF,AF'
0799            POP     AF
0800            RET
```

PROGRAM NAME = SUBR.Z80 PAGE 0017

```
0801    CKEOT   AND     A
0802            LD      A,(HL)
0803            CP      0FFH
0804            JR      NZ,CKEOT1
0805            SCF
0806            RET
0807    CKEOT1  AND     A
0808            RET
0809    ;END SUBROUTINE
0810    ;
0811    FIN     ORG     08FFH
0812            DB      00H
0813            END     100H
```

PROGRAM NAME = RDVC.Z80 PAGE 0001

```
0001    ;SUBROUTINE SAMPLING VOICE INPUT
0002    ;WORK AREA START ADDRESS SET IN REG. HL
0003    ;END WORK AREA IN REG. DE
0004            ORG     05DAH
0005    C01     EQU     03      ;START ITERATION
0006    C02     EQU     16      ;END ITERATION
0007    S0100   PUSH    HL
0008            PUSH    DE
0009            EXX             ;REG TO REG'
0010            POP     DE
0011            POP     HL      ;COPY INFORMATION
0012            LD      B,C01   ;LOAD START ITERATION
0013            LD      C,C02   ;LOAD END ITERATION
0014            EXX             ;REG' TO REG
0015            EX      AF,AF'  ;AF TO AF'
0016            XOR     A       ;CLEAR ALL FLAG
0017    S0101   BIT     0,A     ;CHECK END OF VOICE
0018            JR      NZ,S01END       ;IF END THEN EXIT
0019            EX      AF,AF'  ;AF' TO AF
0020    S0101A  LD      A,01H
0021            LD      B,06H
0022            CALL    DSP1    ;SEND ZERO SAMPLING STATUS
0023    S0102   CALL    RDZERO  ;READ ZERO-CROSSING RATE
0024            JR      Z,S0102 ;IF RESULT IS 0 :RETRY
0025            XOR     A
0026            CALL    DSP1    ;SEND END OF READ
0027            INC     HL
0028            PUSH    HL      ;SAVE WORK AREA ADDR
0029            LD      HL,CLKARA       ;LOAD CLOCK AREA ADDRESS
0030            CALL    RDCLK   ;READ REALTIME CLOCK
0031            PUSH    AF      ;CHECK IF TIME NOT CHANGE
0032            AND     0F0H    ;B4-B7 IS SIGNIFICANT
0033            LD      B,A
0034            INC     HL
0035            LD      A,(HL)
0036            AND     0F0H
0037            XOR     B
0038            JR      NZ,S0102A       ;IF TIME DIFF. THEN CONT
0039            POP     AF
0040            POP     HL
0041            DEC     HL      ;RESET TO BEGIN OF RECORD
0042            JR      S0101A
0043    S0102A  POP     AF
0044            POP     HL      ;RETURN STORE ADDR.
0045            LD      (HL),A
0046            INC     HL
0047            CALL    RDADC   ;SAMPLING DATA
0048            LD      BC,03H
0049            ADD     HL,BC   ;INCREMENT STORE ADDR
0050            CALL    EXVC    ;CHECK IF VOICE
```

```
0051            EX      AF,AF'  ;AF TO AF'
0052            EXX             ;REG TO REG'
0053            BIT     7,B     ;CHECK IF START ITERATION<0
0054            JR      NZ,S0103
0055            DEC     B       ;IF NOT DECREMENT
0056    S0103   EX      AF,AF'  ;AF' TO AF
0057            JR      NZ,S010A        ;IF VOICE THEN SKIP
0058            EX      AF,AF'  ;AF TO AF'
0059            BIT     7,B     ;CHECK START ITER. COUNT
0060            JR      NZ,S0104        ;IF <0 THEN SKIP
0061            PUSH    HL
0062            EXX             ;ELSE RESTART STORE ADDR
0063            POP     HL
0064            JR      S0101   ;RESTART ROUTINE
0065    S0104   BIT     7,C     ;CHECK END ITER. COUNTER
0066            JR      NZ,S0105        ;IF<0 THEN END
0067            DEC     C
0068            JR      S0106   ;ELSE CONTINUE
0069    S0105   SET     0,A     ;SET END OF VOICE FLAG
0070            EX      AF,AF'  ;AF' TO AF
0071            LD      A,0FFH
0072            JR      S0108
0073    S0106   EX      AF,AF'  ;AF' TO AF
0074    S0107   XOR     A       ;SET TO CONTINUE
0075    S0108   EXX             ;REG' TO REG
0076            LD      (HL),A  ;STORE FLAG
0077            INC     HL
0078            PUSH    HL      ;SAVE STORE ADDR.
0079            SBC     HL,DE   ;CHECK ADDRESS
0080            POP     HL
0081            JP      NC,MON  ;IF EXCEED THEN TURN TO MON. ****
0082            EX      AF,AF'  ;AF TO AF'
0083            JR      S0101   ;CONTINUE SAMPLING
0084    S010A   LD      C,C02   ;RESTORE END ITERATION COUNT
0085            JR      S0107
0086    S01END  EX      AF,AF'  ;AF' TO AF
0087            LD      BC,RECLGN       ;LOAD RECORD LENGTH
0088            LD      A,C02   ;LOAD COUNTER
0089    S010B   SBC     HL,BC   ;SKIP BACK (C02) RECORDS
0090            DEC     A
0091            BIT     7,A
0092            JR      Z,S010B
0093            LD      A,0FFH
0094            LD      (HL),A  ;WRITE END OF DATA
0095            RET
0096    SET8    EQU     0354H
0097    PADD    EQU     03B8H
0098    DUMP    EQU     03EFH
0099    A1K     EQU     03FFH
0100    RECLGN  EQU     6
```

PROGRAM NAME = RDVC.Z80 PAGE 0003

```
0101    DSP1     EQU     220H    ;1 CHRS. DISPLAY ADDRESS
0102    RDZERO   EQU     0373H   ;ZERO CROSSING READ SUBR.
0103    RDCLK    EQU     0364H   ;CLOCK READ SUBR.
0104    RDADC    EQU     0335H   ;READ ADC SUBR.
0105    MON      EQU     0000H   ;MONITOR
0106    EXVC     EQU     0382H   ;EXAMINE VOICE ADDR
0107    DSPLY    EQU     0137H   ;MONITOR DISPLAY SUBROUTINE
0108    SCREEN   EQU     04AEH
0109    VELOCTY  EQU     04E4H
0110    FIN      ORG     08FFH
0111             DB      0FFH
0112    SPACE    ORG     319AH
0113    CLKARA   DS      4       ;SYSTEM CLOCK AREA
0114    WORKARA  DS      1024    ;INPUT DATA AREA 1 K
0115             END     100H
```

PROGRAM NAME = TSYS1.Z80 PAGE 0001

```
0001    LDC     MACRO   #X,#Y
0002    ;*******LDC MACRO CALL*****************************
0003            PUSH    AF
0004            LD      A,(#Y)  ;MOVE CHARACTER 1 BYTE FROM #Y TO #X
0005            LD      (#X),A
0006            POP     AF
0007    ;=============================================
0008            MEND
0009    ;
0010    MVC     MACRO   #X,#Y,#Z    ;MOVE CHARACTER MACRO
0011    ;*******MVC MACRO CALL**********************
0012            PUSH    HL      ;MOVE FROM #Y TO #X BY #Z BYTES
0013            PUSH    DE
0014            PUSH    BC
0015            LD      DE,#X
0016            LD      HL,#Y
0017            LD      BC,#Z
0018            LDIR
0019            POP     BC
0020            POP     DE
0021            POP     HL
0022    ;=========================================
0023            MEND
0024    ;
0025    AR      MACRO   #X,#Y   ;ADD REG #Y TO REG #X
0026    ;*******AR MACRO CALL*******************
0027            PUSH    AF
0028            LD      A,#X
0029            ADD     #Y
0030            LD      #X,A
0031            POP     AF
0032    ;=======================================
0033            MEND
0034    ;
0035    A       MACRO   #X,#Y   ;ADD STORAGE TO REGISTER
0036    ;*******A MACRO CALL********************
0037            PUSH    HL
0038            PUSH    AF
0039            LD      HL,#Y
0040            LD      A,#X
0041            ADD     (HL)
0042            LD      #X,A
0043            POP     AF
0044            POP     HL
0045    ;=====================================
0046            MEND
0047    ;
0048    AM      MACRO   #X,#Y   ;ADD STORAGE TO STORAGE
0049    ;*******AM MACRO CALL********************
0050            PUSH    HL
```

```
0051                PUSH    AF
0052                LD      HL,#Y
0053                LD      A,(#X)
0054                ADD     (HL)
0055                LD      (#X),A
0056                POP     AF
0057                POP     HL
0058    ;=====================================
0059                MEND
0060    ;
0061    AVRG        MACRO   #X,#Y    ;#X=INT((#X+#Y)/2)
0062    ;*******AVRG MACRO CALL*****************
0063                LD      A,(#X)
0064                LD      HL,#Y
0065                ADD     (HL)
0066                RRA
0067                LD      (#X),A
0068    ;=====================================
0069                MEND
0070    ;
0071    SUM         MACRO   #X       ;#X=#X+(HL) :#X LENGTH 1 BYTE
0072    ;*******SUM MACRO CALL**********************
0073                PUSH    AF
0074                PUSH    HL
0075                LD      A,(#X)
0076                ADD     (HL)
0077                LD      (#X),A
0078                LD      A,(#X+1)
0079                ADC     00H
0080                LD      (#X+1),A
0081                POP     HL
0082                POP     AF
0083    ;=======================================
0084                MEND
0085    ;
0086    ;
0087    ;SUBROUTINE ADJUST DATA AND FIND
0088    ;SUMMATION OF SAMPLING VALUE
0089    ;GROUPING INTO 4 PERIOD
0090    ;
0091    ;ADDRESS OF SOURCE IN REG.HL
0092    ;END OF DATA SPACE IN DE.
0093    ;RETURN END OF ADJUSTED DATA IN BC
0094    ;
0095                ORG     0667H
0096    CMPRS       PUSH    HL
0097                PUSH    DE       ;SAVE INFORMATION
0098                PUSH    HL       ;SAVE HL
0099                LD      HL,MAX   ;SET MAX=0
0100                LD      B,3
```

```
0101            XOR     A
0102    CMINIT  LD      (HL),A
0103            INC     HL
0104            DJNZ    CMINIT
0105            POP     HL
0106            CALL    GETDAT  ;GET DATA TO BUFFER AREA
0107            CALL    STMAX   ;FIND MAXIMUM VALUE
0108            CALL    MOVDAT  ;MOVE DATA TO STORE AREA
0109            CALL    GETNEXT ;GET NEXT RECORD
0110            CALL    STMAX
0111            JP      C,CMERR ;ERROR IF EOF
0112    CMPR01  LD      A,(TIME)
0113            PUSH    HL
0114            LD      HL,T1
0115            SUB     (HL)    ;DT=TIME-T1
0116            POP     HL
0117            CP      01H     ;COMPARE WITH 1
0118            JR      Z,CMPR02  ;IF = 1 THEN STORE AND GETNEXT
0119            JR      NC,CMPR04 ;IF > 1 THEN INSERT RECORD
0120            LD      A,(TIME)  ;CHECK IF TIME=0
0121            CP      00H
0122            JP      NZ,CMERR  ;IF NOT = 0 THEN ERROR
0123    CMPR02  LD      A,(FLAG)     ;CHECK IF EOF
0124            CP      0FFH
0125            JR      Z,CMPRNX     ;IF EOF THEN CONTINUE
0126            CALL    MOVDAT    ;ELSE GET DATA
0127    CMPR03  POP     DE
0128            PUSH    DE
0129            CALL    GETNEXT   ;GET NEXT DATA
0130            CALL    STMAX
0131            JR      CMPR01       ;DO WHILE NOT EOF
0132    CMPR04  POP     DE      ;RETURN END OF DATA AREA
0133            PUSH    DE      ;SAVE
0134            LD      B,5     ;INSERT 5 BYTE
0135            XOR     A
0136            CALL    INSERT
0137            PUSH    HL
0138            LD      HL,T1
0139            INC     (HL)    ;T1=T1+1
0140            AVRG    B1,BPF1 ;MACRO CALL *****
0141            AVRG    B2,BPF2 ;MACRO CALL *****
0142            AVRG    B3,BPF3 ;MACRO CALL *****
0143            POP     DE
0144            PUSH    DE
0145            LD      HL,T1   ;STORE AREA ADDRESS
0146            LD      BC,4
0147            LDIR
0148            POP     HL
0149            JR      CMPR03  ;CONTINUE PROCESSING
0150    CMPRNX  LD      BC,4    ;FIND NEW EOF ADDR
```

```
0151            ADD     HL,BC
0152            LD      (EOFAD),HL ;KEEP EOF ADDR
0153            LD      DE,(TIME)
0154            LD      D,0
0155            LD      HL,0000H
0156            LD      BC,0003H
0157            CALL    DVDE16
0158            LD      A,E
0159            LD      (TEND),A
0160            LD      A,(TIME)
0161            LD      (TEND+2),A
0162            SUB     E
0163            LD      (TEND+1),A
0164            POP     DE      ;RETURN END ADDRESS
0165            POP     HL      ;RETURN TABLE ADDRESS
0166            PUSH    DE
0167            PUSH    HL
0168            PUSH    HL
0169            POP     IX      ;STORE START ADDRESS
0170            LD      IY,A01H ;01H STORE ADDR
0171            LD      HL,05   ;RECORD LENGTH
0172            LD      DE,(EOFAD)  ;END OF DATA ADDR.
0173            LD      B,1     ;KEY LENGTH
0174            CALL    SEARCH  ;SEARCH RECORD TIME = 01H
0175            JP      NZ,CMERR  ;IF NOT FOUND THEN ERROR
0176            LD      (STADR),HL      ;STORE 1ST RECORD ADDRESS
0177            LD      B,3     ;ELSE DO I=1,3
0178            LD      DE,TEND ;END INTERVAL WORK AREA
0179            EXX
0180            LD      DE,SAVESUM+18  ;SAVESUM STACK AREA
0181            EXX
0182    CMPRLL  PUSH    HL
0183            LD      HL,0000H ;CLEAR SUM AREA
0184            LD      (SUMBF1),HL
0185            LD      (SUMBF2),HL
0186            LD      (SUMBF3),HL
0187            POP     HL
0188            LD      A,(DE)  ;KEEP TRACK END INT.
0189            LD      C,A
0190    CMPR05  PUSH    HL
0191            INC     HL
0192            SUM     SUMBF1  ;MACRO CALL *****
0193            INC     HL
0194            SUM     SUMBF2  ;MACRO CALL *****
0195            INC     HL
0196            SUM     SUMBF3  ;MACRO CALL *****
0197            POP     HL
0198            LD      A,(HL)
0199            CP      C       ;CHECK END OF INTERVAL
0200            JR      NC,CMPR06
```

```
0201            PUSH    BC
0202            LD      BC,5
0203            ADD     HL,BC   ;SKIP TO NEXT ADDR
0204            POP     BC
0205            JR      CMPR05  ;DO WHILE NOT END INTERVAL
0206    CMPR06  EXX
0207            LD      HL,0000H
0208            ADD     HL,SP   ;SAVE STACK POINTER
0209            EX      DE,HL
0210            LD      SP,HL
0211            LD      BC,(SUMBF1)
0212            PUSH    BC
0213            LD      BC,(SUMBF2)
0214            PUSH    BC
0215            LD      BC,(SUMBF3)
0216            PUSH    BC
0217            LD      HL,0000H
0218            ADD     HL,SP
0219            EX      DE,HL
0220            LD      SP,HL
0221            EXX
0222            INC     DE      ;LOAD NEXT END INTERVAL
0223            PUSH    BC
0224            LD      BC,05H
0225            ADD     HL,BC   ;SKIP RECORD POINTER
0226            POP     BC
0227            DJNZ    CMPRLL  ;DO WHILE NOT EOI
0228            EXX
0229            LD      HL,0000H
0230            ADD     HL,SP
0231            PUSH    HL
0232            POP     BC      ;SAVE CURRENT STACK POINTER IN BC
0233            LD      HL,RESULT+19  ;SET DATA STACK POINTER
0234            EX      DE,HL
0235            LD      SP,HL   ;SET STACK TO SAVE AREA
0236            EXX
0237            LD      B,9
0238    CMPRLP  POP     HL      ;TRANSFER DATA THRU STACK
0239            EXX
0240            LD      HL,0000H
0241            ADD     HL,SP
0242            EX      DE,HL
0243            LD      SP,HL
0244            EXX
0245            PUSH    HL
0246            EXX
0247            LD      HL,0000H
0248            ADD     HL,SP
0249            EX      DE,HL
0250            LD      SP,HL
```

```
0251            EXX
0252            DJNZ    CMPRLP
0253            EXX             ;RETURN TO CURRENT STACK
0254            LD      HL,0000H
0255            ADD     HL,BC
0256            LD      SP,HL
0257            EXX
0258            LD      A,(TIME)
0259            LD      (RESULT),A
0260            POP     HL
0261            POP     DE
0262            LD      BC,(EOFAD)
0263            RET
0264    GETDAT  PUSH    HL
0265            PUSH    DE
0266            PUSH    BC      ;GET DATA TO BUFFER
0267            LD      DE,TIME
0268            LD      BC,05H
0269            LDIR
0270            POP     BC
0271            POP     DE
0272            POP     HL
0273            RET
0274    GETNEXT PUSH    BC
0275            PUSH    DE
0276            PUSH    HL
0277            LD      BC,04H
0278            ADD     HL,BC
0279            LD      A,(HL)
0280            CP      0FFH
0281            POP     HL
0282            JR      NZ,GCONT
0283            SCF
0284            JR      GRET
0285    GCONT   LD      BC,05H
0286            ADD     HL,BC
0287            PUSH    HL
0288            AND     A
0289            SBC     HL,DE
0290            POP     HL
0291            PUSH    HL
0292            JR      C,GCON1
0293            JP      MON
0294    GCON1   LD      DE,TIME
0295            LDIR
0296            POP     HL
0297            AND     A
0298    GRET    POP     DE
0299            POP     BC
0300            RET
```

```
0301    MOVDAT  MVC     T1,TIME,5
0302            RET
0303    STMAX   PUSH    HL      ;FIND MAXIMUM
0304            PUSH    DE
0305            PUSH    BC
0306            PUSH    AF
0307            LD      DE,TIME+1   ;INPUT BUFFER ADDRESS
0308            LD      HL,MAX  ;MAXIMUM OF EACH BPF.
0309            LD      B,3     ;3 BPF EXAMINED
0310    STMAX1  LD      A,(DE)
0311            CP      (HL)
0312            JR      C,STMAX2    ;IF NEW<OLD THEN CONTINUE
0313            LD      (HL),A
0314    STMAX2  INC     HL
0315            INC     DE
0316            DJNZ    STMAX1
0317            POP     AF
0318            POP     BC
0319            POP     DE
0320            POP     HL
0321            RET
0322    A01H    DB      01H
0323    ;SUBROUTINE SET TABLE
0324    ;TABLE FORMAT
0325    ;BYTE NO. 1-2 = FLAGBYTE (1AH)
0326    ;BYTE NO. 3-4 = END OF TABLE ADDRESS
0327    ;BYTE NO. 5-6 = CURRENT TABLE BOTTOM
0328    ;
0329    SETTAB  PUSH    HL
0330            PUSH    BC
0331            PUSH    AF      ;SAVE INFORMATION
0332            LD      HL,TABLE        ;TABLE ADDRESS
0333            LD      BC,11FFH        ;6K SIZE DEFAULT
0334            LD      A,0FFH  ;PADDING CHARACTER
0335            CALL    PADD    ;CLEAR TABLE
0336            ADD     HL,BC
0337            DEC     HL
0338            LD      (TABLE+2),HL    ;STORE END OF TABLE
0339            LD      HL,TABLE+5      ;CURRENT BOTTOM ADDRESS
0340            LD      (TABLE+4),HL    ;STORE BOTTOM ADDRESS
0341            LD      BC,1A1AH
0342            LD      (TABLE),BC      ;STORE TABLE FLAG
0343            POP     AF
0344            POP     BC
0345            POP     HL
0346            RET
0347    ;END SUBROUTINE
0348    ;SUBROUTINE INSERT TABLE
0349    ;SOURCE DATA FROM (RESULT) AND (MEANING)
0350    ;EXIT TO ERROR WHEN TABLE IS FULL
```

```
0351    ;RECORD LENGTH = 7 BYTE
0352    ;
0353    INSTAB  PUSH    DE
0354            PUSH    BC
0355            PUSH    AF      ;SAVE INFORMATION
0356            PUSH    HL
0357            LD      HL,(TABLE+4)    ;CHECK AVAILABLE AREA
0358            LD      BC,5
0359            ADD     HL,BC   ;LOOK AHEAD
0360            LD      DE,(TABLE+2)    ;END OF TABLE
0361            PUSH    HL
0362            AND     A       ;CLEAR CARRY FLAG
0363            SBC     HL,DE
0364            JP      NC,TERROR       ;IF EXCEED THEN ERROR
0365            POP     HL
0366            LD      (TABLE+4),HL    ;STORE NEW BOTTOM
0367            POP     HL
0368            LD      B,5
0369            LD      A,0FFH
0370            CALL    INSERT  ;INSERT TABLE
0371            EX      DE,HL   ;TOP OF TABLE ADDRESS TO DE
0372            PUSH    DE
0373            INC     DE
0374            LD      HL,FINAL        ;SOURCE DATA
0375            LD      BC,4    ;19 BYTE MOVE
0376            LDIR            ;MOVE
0377            POP     DE
0378            LD      A,(MEAN)        ;STORE MEANING
0379            LD      (DE),A
0380            POP     AF
0381            POP     BC
0382            POP     DE
0383            RET
0384    ;END SUBROUTINE
0385    ;
0386    ;SUBROUTINE DELETE TABLE
0387    ;INFORMATION AT THE TOP OF TABLE
0388    ;IS DELETED
0389    ;
0390    DELTAB  PUSH    HL
0391            PUSH    DE
0392            PUSH    BC
0393            PUSH    AF
0394            LD      HL,(TABLE+4)
0395            LD      DE,TABLE+7
0396            AND     A       ;CHECK IF TOP OF TABLE
0397            SBC     HL,DE
0398            JR      C,DELTAR        ;IF TOP THEN RET
0399            LD      HL,(TABLE+4)
0400            LD      BC,8
```

```
0401            AND     A       ;CLEAR CARRY FLAG
0402            SBC     HL,BC
0403            LD      (TABLE+4),HL    ;STORE NEW BOTTOM
0404            LD      HL,TABLE+6
0405            LD      DE,(TABLE+2)
0406            LD      B,8
0407            LD      A,0FFH
0408            CALL    DELETE
0409    DELTAR  POP     AF
0410            POP     BC
0411            POP     DE
0412            POP     HL
0413            RET
0414    ;END SUBROUTINE
0415    ;
0416    ;MESSAGE DISPLAY SUBROUTINE
0417    ;ASSIGN MESSAGE BY SPECIFIED VALUE
0418    ;TO ACCUMULATOR
0419    ;
0420    MSGE    PUSH    HL
0421            PUSH    DE
0422            PUSH    BC
0423            PUSH    AF
0424            LD      HL,MSGTAB       ;MESSAGE TABLE ADDR
0425            LD      DE,ENDMSG-1
0426    FMSGE   CP      (HL)    ;FIND REQUIRED MESSAGE
0427            JR      Z,MDSP  ;IF FOUND THEN DISPLAY
0428            INC     HL      ;ELSE
0429            PUSH    HL      ;CHECK IF END OF TABLE
0430            AND     A
0431            SBC     HL,DE
0432            POP     HL
0433            JR      C,FMSGE ;DO WHILE NOT EOT
0434    MSGRTN  POP     AF
0435            POP     BC
0436            POP     DE
0437            POP     HL
0438            RET             ;RETURN**********
0439    MDSP    INC     HL      ;DISPLAY MESSAGE
0440            LD      B,(HL)  ;NO. OF DIGIT
0441            RES     7,B
0442            XOR     A
0443            INC     HL
0444            CALL    DSP2    ;DISPLAY
0445            JR      MSGRTN  ;RETURN
0446    MSGTAB  DB      00H,86H,0FFH,0FFH,0FFH
0447            DB      0FFH,0FFH,0FFH
0448    RUN     DB      01H,85H,0C7H,0AFH
0449            DB      0ABH,0FFH,0BFH
0450    READY   DB      02H,86H,0AFH,86H,88H
```

```
0451                DB      0A1H,91H,0FFH
0452    ERR         DB      03H,84H,86H,0AFH
0453                DB      0AFH,0BFH
0454    HYPHEN      DB      04H,85H,0FFH,0FFH,0FFH
0455                DB      0FFH,0BFH
0456    ENDMSG      DB      00H
0457    ;END SUBROUTINE
0458    ;
0459    CMERR       LD      A,3
0460                CALL    MSGE        ;DISPLAY ERR-
0461                LD      HL,M01
0462    CDSP        LD      B,2         ;CODE DISPLAY
0463                LD      A,4
0464                CALL    DSP2
0465                HALT
0466    OFERR       LD      A,3
0467                CALL    MSGE
0468                LD      HL,M00
0469                JR      CDSP
0470    TERROR      LD      A,3
0471                CALL    MSGE
0472                LD      HL,M02
0473                JR      CDSP
0474    M00         DB      0C0H,0C0H
0475    M01         DB      0C0H,0F9H
0476    M02         DB      0C0H,0A4H
0477    ;EXAMINE KEYBOARD
0478    ;DEFINE AS FOLLOW
0479    ;7   8   9   X       XXXXXX
0480    ;4   5   6   D       X    S      D=DELETE TABLE:S=SET TABLE
0481    ;1   2   3   E       X    X      E=ESCAPE
0482    ;0   L   R   I       ENTER.      I=IGNORE
0483    ;
0484    ;RETURN CONTENT IN (DATA)
0485    PANNEL      PUSH    HL
0486                PUSH    DE
0487                PUSH    BC
0488                PUSH    AF
0489                LD      A,0FFH
0490                LD      (DATA),A        ;CLEAR BUFFER
0491                CALL    SCAN    ;TRACE KEYBOARD
0492                JR      NZ,PAN01        ;IF KEY PRESS
0493    PANRET      POP     AF
0494                POP     BC
0495                POP     DE
0496                POP     HL      ;RETURN POINT
0497                RET
0498    PAN01       DEC     A
0499                LD      HL,PANTAB       ;KEY VAL. TABLE
0500                ADD     L
```

```
0501            LD      L,A
0502            JR      NC,PAN02
0503            INC     H
0504    PAN02   LD      A,(HL)  ;LOAD VALUE
0505            LD      (DATA),A        ;STORE
0506            CP      0AH     ;IF 0-9 THEN DISPLAY
0507            JR      NC,PANREL
0508            LD      HL,DTAB
0509            ADD     L
0510            LD      L,A
0511            JR      NC,PAN03
0512            INC     H
0513    PAN03   LD      A,(HL)  ;DISPLAY CHR.
0514            CPL
0515            LD      B,5     ;DIGIT 5
0516            CALL    DSP1
0517    PANREL  CALL    KDELAY  ;WAIT UNTIL RELEASE
0518            CALL    SCAN
0519            JR      NZ,PANREL
0520            JR      PANRET
0521    PANTAB  DB      07H,04H,01H,00H
0522            DB      08H,05H,02H,'L'
0523            DB      09H,06H,03H,'R'
0524            DB      0FFH,'D','E','I'
0525            DB      0FFH,0FFH,0FFH,0DH
0526            DB      0FFH,'S',0FFH,0FFH
0527            DB      00H
0528    DSP1    EQU     0220H
0529    DTAB    EQU     01EFH   ;DISPLAY TABLE
0530    KDELAY  EQU     01E2H   ;KEYBOARD DELAY
0531    DSP2    EQU     022BH
0532    PADD    EQU     03B8H
0533    SCAN    EQU     02ABH
0534    INSERT  EQU     050DH
0535    DELETE  EQU     0532H
0536    SEARCH  EQU     02FFH
0537    MULT16  EQU     0389H   ;MULTIPLY 16 BIT SUBR.
0538    DVDE16  EQU     03A1H   ;DIVIDE 32 BY 16 BIT SUBR.
0539    MON     EQU     0000H   ;MONITOR
0540    WKAREA  ORG     1900H
0541    T1      DS      1
0542    B1      DS      1
0543    B2      DS      1
0544    B3      DS      1
0545    TIME    DS      1
0546    BPF1    DS      1
0547    BPF2    DS      1
0548    BPF3    DS      1
0549    FLAG    DS      1
0550    RESULT  DS      19
```

PROGRAM NAME = TSYS1.Z80 PAGE 0012

```
0551    STADR   DS      2
0552    EOFAD   DS      2
0553    MAX     DS      3
0554    KEY     DS      6
0555    TEND    DS      4
0556    SAVESUM DS      18
0557    SUMBF1  DS      2
0558    SUMBF2  DS      2
0559    SUMBF3  DS      2
0560    SYSTACK DS      2       ;SYSTEM STACK SAVE AREA
0561    WKSTACK DS      2       ;DATA STACK POINTER SAVE AREA
0562    BASE    DS      2       ;BASE VALUE SAVE AREA
0563    MEAN    DS      1
0564    DATA    DS      1       ;KEYBOARD DATA BUFFER
0565    PEAK    DS      3
0566    DIR     DS      3
0567    ST      DS      3
0568    PT      DS      3
0569    LAST    DS      3
0570    DIFF    DS      3
0571    SLOPE   DS      3
0572    TYPE    DS      9
0573    MODE    DS      3
0574    DIFST   DS      6
0575    DIFLAST DS      6
0576    AATYPE  DS      2
0577    ADATA   DS      2
0578    ATEND   DS      2
0579    ATYPE   DS      2
0580    AMODE   DS      2
0581    APT     DS      2
0582    AST     DS      2
0583    ALAST   DS      2
0584    ADIFST  DS      2
0585    ADFLST  DS      2
0586    ADIFF   DS      2
0587    ADIR    DS      2
0588    APEAK   DS      2
0589    CAREA   DS      6
0590    TABLE   DS      1200H
0591    TDIF    DS      1
0592    VALUE   DS      1
0593    PAT     DS      1
0594    SCOUNT  DS      1
0595    SBF1    DS      2
0596    SBF2    DS      2
0597    SBF3    DS      2
0598    STIME   DS      2
0599    STMAXX  DS      1
0600    STMIN   DS      1
```

PROGRAM NAME = TSYS1.Z80 PAGE 0013

```
0601    STMEAN  DS      1
0602    SMAX1   DS      1
0603    SMIN1   DS      1
0604    SMEAN1  DS      1
0605    SMAX2   DS      1
0606    SMIN2   DS      1
0607    SMEAN2  DS      1
0608    SMAX3   DS      1
0609    SMIN3   DS      1
0610    SMEAN3  DS      1
0611    RES     DS      4
0612    INTRN   DS      3
0613    STAB    DS      120
0614    SCORE   DS      10
0615    FINAL   DS      4
0616    SMX     DS      2
0617    SMIN    DS      2
0618    UNUSE   DS      1367
0619    CLKARA  DS      4
0620    WORKARA DS      03FFH
0621            END     100H
```

PROGRAM NAME = TSYS2.Z80 PAGE 0001

```
0001    MVC     MACRO   #X,#Y,#Z
0002    ;=======MVC MACRO CALL=============
0003            PUSH    HL      ;MOVE DATA FROM #Y TO #X
0004            PUSH    DE      ;BY #Z BYTES
0005            PUSH    BC
0006            LD      DE,#X   ;LOAD DESTINATION
0007            LD      HL,#Y   ;LOAD SOURCE
0008            LD      BC,#Z   ;LOAD NO. OF BYTE
0009            LDIR
0010            POP     BC
0011            POP     DE
0012            POP     HL
0013    ;==================================
0014            MEND
0015    STORE   MACRO   #X,#Y   ;STORE POINTER
0016    ;========================================
0017            LD      HL,#Y   ;STORE ADDRESS OF #Y
0018            LD      (#X),HL ;TO LOCATION (#X)
0019    ;========================================
0020            MEND
0021    CMPTR   MACRO   #X,#Y   ;COMPARE
0022    ;========================================
0023            LD      HL,(#X) ;COMPARE VALUE POINT BY (#X)
0024            LD      A,(HL)  ;WITH VALUE #Y
0025            CP      #Y
0026            MEND
0027    LDREL   MACRO   #X,#Y   ;LOAD RELATIVE
0028            LD      DE,(#X) ;LOAD DATA FROM ADDRESS
0029            LD      HL,#Y   ;POINT BY ((#X)+#Y)
0030            ADD     HL,DE
0031            LD      A,(HL)
0032    ;========================================
0033            MEND
0034    STREL   MACRO   #X,#Y   ;STORE RELATIVE
0035    ;========================================
0036            LD      HL,(#X) ;STORE #Y TO LOCATION
0037            LD      A,#Y    ;POINT BY (#X)
0038            LD      (HL),A
0039    ;========================================
0040            MEND
0041    LDPTR   MACRO   #X,#Y   ;MOVE CONTENT
0042    ;========================================
0043            LD      HL,(#Y) ;MOVE CONTENT 1 BYTE
0044            LD      A,(HL)  ;POINT BY (#Y) TO (#X)
0045            LD      HL,(#X)
0046            LD      (HL),A
0047    ;========================================
0048            MEND
0049    ;ROUTINE TO BUILD KEY FOR ANALYSE
0050    ;VOICE SIGNAL
```

```
0051    ;KEY IS BUILD FROM
0052    ;1.   VOICE DURATION
0053    ;        IF DURATION< 10H THEN CLASS=1
0054    ;        ----------> 18H THEN CLASS=3
0055    ;        ELSE                  CLASS=2
0056    ;2.   AVERAGE OF VOICE DATA IN EACH PERIOD
0057    ;        TOTAL = 3 PERIOD
0058    ;        COMPARATION IS MADE AND THE RESULT
0059    ;        IS THAT MAX. OF TOTAL VALUE IS IN
0060    ;        WHICH PERIOD
0061    ;        I1 OR I2 OR I3
0062    ;3.   TYPE OF CURVE OF EACH BPF IN EACH PERIOD
0063    ;        FOUR TYPE IS CLASSIFIED
0064    ;        CLASS 1 : HAT SHAPE
0065    ;        CLASS 2 : V   SHAPE
0066    ;        CLASS 3 : SNAKE SHAPE
0067    ;        CLASS 4 : POSITIVE SLOPE INCLINE
0068    ;        CLASS 5 : NEGATIVE SLOPE INCLINE
0069    ;        CLASS 6 : NORMAL PLANE
0070    ;
0071    ;        KEY IS BUILT FROM THESE CLASSIFICATION
0072    ;        KEY LENGTH = 6 BYTE
0073    ;        INPUT DATA MUST BE SCREEN AND COMPRESS
0074    ;        BEFORE ENTER THIS ROUTINE
0075    ;
0076    ;
0077    ;++++++++++++++++++++++++++++++++++++++++++++++++
0078    ;SUBROUTINE COMPARE
0079    ;3 SET OF 2 BYTE DATA WILL BE COMPARE
0080    ;RETURN RESULT IN ACCUMULATOR
0081    ;LOGIC
0082    ;        IF A=>B AND A>C THEN RETURN (1)
0083    ;        IF B=>C AND B>A THEN RETURN (2)
0084    ;        IF C=>A AND C>B THEN RETURN (3)
0085    ;        ELSE RETURN (1)
0086    ;DATA MUST BE STORE IN CAREA TO CAREA+5
0087    ;
0088            ORG     09A6H
0089    CMPARE  PUSH    HL
0090            PUSH    DE
0091            PUSH    BC      ;SAVE REGISTER
0092            LD      HL,(CAREA)
0093            LD      DE,(CAREA+2)
0094            AND     A       ;CLEAR CARRY FLAG
0095            SBC     HL,DE   ;COMPARE A:B
0096            JR      C,CPRE1 ;IF B>A THEN CHECK B AND C
0097            LD      HL,(CAREA+4)  ;ELSE CHECK IF A>C
0098            LD      DE,(CAREA)
0099            AND     A
0100            SBC     HL,DE
```

```
0101            JR      NC,CPRE2        ;IF A NOT>C THEN CHECK B:C
0102            LD      A,1     ;ELSE A>C THEN RETURN (1)
0103            JR      CPRET
0104    CPRE1   LD      HL,(CAREA+4)
0105            AND     A
0106            EX      DE,HL
0107            SBC     HL,DE   ;COMPARE B AND C
0108            JR      C,CPRE3 ;IF C>B THEN RETURN(3)
0109            LD      A,2     ;ELSE RETURN (2)
0110            JR      CPRET
0111    CPRE2   LD      HL,(CAREA+2)
0112            LD      DE,(CAREA+4)
0113            AND     A
0114            SBC     HL,DE   ;COMPARE C AND B
0115            JR      NC,CPRE4  ;IF C NOT> B THEN RETURN (0)
0116    CPRE3   LD      A,3     ;ELSE RETURN (3)
0117            JR      CPRET
0118    CPRE4   XOR     A
0119    CPRET   POP     BC
0120            POP     DE
0121            POP     HL
0122            RET
0123    ;END SUBROUTINE
0124    ;
0125    ;SUBROUTINE NORMALIZE
0126    ;SEND DATA IN (A)
0127    ;POINTER TO BASE VALUE IN HL
0128    ;RETURN RESULT IN (A)
0129    ;
0130    PUNIT   PUSH    BC
0131            PUSH    DE
0132            PUSH    HL
0133            PUSH    AF
0134            XOR     A       ;CHECK IF BASE VALUE = 0
0135            CP      (HL)
0136            JR      Z,PURET ;RETURN IF ZERO
0137            POP     AF
0138            LD      E,A     ;MULTIPLY DATA TO 0FFH
0139            LD      D,0
0140            LD      HL,0000H
0141            LD      BC,0FFH
0142            CALL    MULT16
0143            PUSH    HL
0144            POP     BC      ;SAVE HL IN BC
0145            POP     HL
0146            LD      A,(HL)  ;LOAD BASE VALUE
0147            PUSH    HL
0148            PUSH    BC
0149            POP     HL
0150            EX      DE,HL   ;SET VALUE FOR DIVISION
```

```
0151            LD      C,A
0152            LD      B,0
0153            CALL    DVDE16  ;X=(A)*256/(BASE)
0154            LD      A,E
0155            JR      PURET1
0156    PURET   POP     AF
0157    PURET1  POP     HL
0158            POP     DE
0159            POP     BC
0160            RET
0161    ;END SUBROUTINE
0162    ;
0163    ;SUBROUTINE GET DATA AND NORMALIZE
0164    ;RECORD POINTER BY HL
0165    ;BASE VALUE FROM (MAX) TO (MAX+2)
0166    ;RETURN RESULT IN (PT)
0167    ;
0168    GETPT   PUSH    HL
0169            PUSH    DE
0170            PUSH    AF      ;SAVE REGISTER
0171            INC     HL      ;SKIP TIME
0172            LD      A,(HL)  ;LOAD DATA BPF1
0173            EX      DE,HL
0174            LD      HL,MAX
0175            CALL    PUNIT   ;NORMALIZE
0176            LD      (PT),A
0177            INC     HL
0178            INC     DE
0179            LD      A,(DE)
0180            CALL    PUNIT
0181            LD      (PT+1),A
0182            INC     HL
0183            INC     DE
0184            LD      A,(DE)
0185            CALL    PUNIT
0186            LD      (PT+2),A
0187            POP     AF
0188            POP     DE
0189            POP     HL
0190            RET
0191    ;END SUBROUTINE
0192    ;SUBROUTINE FIND DIFFERENCE
0193    ;X=A-B
0194    ;POINTER TO A DEFINE IN HL
0195    ;POINTER TO B DEFINE IN DE
0196    ;RESULT STORE IN (DIFF)
0197    DIFER   PUSH    HL
0198            PUSH    DE
0199            PUSH    BC
0200            PUSH    AF
```

```
0201            PUSH    IX
0202            PUSH    IY      ;SAVE REGISTER
0203            EX      DE,HL
0204            LD      IX,DIFF ;RESULT ADDRESS
0205            LD      IY,SLOPE ;SLOPE RESULT
0206            LD      B,3     ;PROCESS 3 FIELD
0207    DIFF01  LD      A,(DE)  ;FETCH A
0208            SUB     (HL)    ;A-B
0209            PUSH    AF      ;SAVE RESULT
0210            JR      NC,DIFF02  ;IF A>B THEN..
0211            LD      A,2     ;RETURN NEGATIVE SIGN
0212            LD      (IY),A
0213            POP     AF
0214            CPL             ;X=ABS(X)
0215            INC     A
0216            JR      DIFF04  ;STORE DATA
0217    DIFF02  JR      NZ,DIFF03  ;SKIP IF POSITIVE
0218            XOR     A
0219            LD      (IY),A
0220            POP     AF
0221            JR      DIFF04
0222    DIFF03  LD      A,1     ;STORE POSITIVE SIGN
0223            LD      (IY),A
0224            POP     AF
0225    DIFF04  LD      (IX),A  ;STORE ABS(A-B)
0226            INC     HL      ;SKIP RECORD ADDRESS
0227            INC     DE
0228            INC     IX
0229            INC     IY
0230            DJNZ    DIFF01
0231            POP     IY
0232            POP     IX
0233            POP     AF
0234            POP     BC
0235            POP     DE
0236            POP     HL
0237            RET
0238    ;END SUBROUTINE
0239    ;
0240    ;SUBROUTINE BUILD KEY
0241    ;*INDEPENDENT SUBROUTINE
0242    ;SOURCE DATA FROM SCREEN PROGRAM
0243    ;RESULT RETURN KEY
0244    BKEY    PUSH    HL
0245            PUSH    DE
0246            PUSH    BC
0247            PUSH    AF
0248            PUSH    IX
0249            PUSH    IY      ;SAVE REGISTER CONTENT
0250            STORE   AATYPE,TYPE
```

```
0251            STORE   ADATA,(STADR)
0252            STORE   ATEND,TEND
0253            LD      B,3       ;INTERVAL COUNTER = 3
0254    BLOP    PUSH    BC        ;DO WHILE NOT END OF INTERVAL
0255    MSET    LD      HL,(AATYPE)    ;SET TYPE =6
0256            LD      DE,MODE        ;SET MODE =0
0257            LD      IX,PEAK        ;SET PEAK =0
0258            LD      B,3       ;3 FIELD BPF1 TO BPF3
0259            LD      C,6
0260            XOR     A
0261    MSET1   LD      (DE),A    ;SET LOOP
0262            LD      (IX),A
0263            LD      (HL),C
0264            INC     DE
0265            INC     IX
0266            INC     HL
0267            DJNZ    MSET1     ;END SET LOOP
0268            LD      HL,(ADATA)       ;GET 1ST RECORD
0269            CALL    GETPT
0270            LD      BC,5      ;SKIP REC.PTR
0271            ADD     HL,BC
0272            LD      (ADATA),HL
0273            MVC     LAST,PT,3        ;LAST=PT
0274            MVC     ST,PT,3          ;ST=PT
0275    BLD0    STORE   ATYPE,(AATYPE)
0276            LD      HL,(ADATA)       ;GET NEXT DATA
0277            CALL    GETPT
0278            LD      B,3       ;EXAMINE IF ANY TYPE=6
0279            LD      A,6
0280            LD      IX,(ATYPE)
0281    BCHK1   CP      (IX)      ;CHECK LOOP
0282            JR      Z,BLD1    ;IF ANY THEN FIND (PT-ST)
0283            INC     IX
0284            DJNZ    BCHK1     ;END CHECK LOOP
0285            JR      BLD2      ;IF NON THEN EXIT
0286    BLD1    LD      HL,PT     ;FIND PT-ST
0287            LD      DE,ST
0288            CALL    DIFER
0289            MVC     DIFST,DIFF,6     ;KEEP RESULT
0290    BLD2    LD      HL,PT     ;FIND PT-LAST
0291            LD      DE,LAST
0292            CALL    DIFER
0293            MVC     DIFLAST,DIFF,6   ;SAVE RESULT
0294            LD      HL,PT
0295            LD      DE,PEAK ;FIND PT-PEAK
0296            CALL    DIFER ;RESULT REMAIN IN DIFF
0297            STORE   AMODE,MODE
0298            STORE   APT,PT
0299            STORE   AST,ST
0300            STORE   ALAST,LAST
```

```
0301            STORE   ADIFST,DIFST
0302            STORE   ADFLST,DIFLAST
0303            STORE   ADIFF,DIFF
0304            STORE   ADIR,DIR
0305            STORE   APEAK,PEAK
0306            LD      B,3     ;DO 3 BPF
0307    BLD3    PUSH    BC      ;SAVE ITERATION COUNTER
0308            CMPTR   ATYPE,6 ;CHECK IF TYPE(K)=6
0309            JR      NZ,BLD5 ;IF NOT THEN NEXT
0310            CMPTR   ADIFST,0CH      ;ELSE IF(PT-ST)>0CH
0311            JP      C,BLDX  ;IF < THEN NEXT K
0312            LDREL   ADIFST,3        ;ELSE A=SGN(PT-ST)
0313            STREL   ADIR,A  ;DIR=SGN(PT-ST)
0314            CP      02H     ;IF SGN IS -1 THEN
0315            JR      NZ,BLD4 ;DO .
0316            STREL   ATYPE,5 ;TYPE=5
0317            JP      BLDX    ;NEXT K
0318    BLD4    STREL   ATYPE,4 ;TYPE=4
0319            JP      BLDX    ;NEXT K
0320    BLD5    CMPTR   AMODE,0 ;ELSE IF MODE <> 6
0321            JR      NZ,BLD6
0322            LDREL   ADFLST,3        ;LOAD SLOPE(K)
0323            LD      HL,(ADIR)       ;COMPARE IF SLOPE=DIR
0324            CP      (HL)
0325            JR      Z,BLD6
0326            STREL   AMODE,01H  ;IF MODE=1 AND SLOPE<>DIR THEN
0327            LDPTR   APEAK,ALAST     ;MODE=2,PEAK=LAST
0328            LD      HL,PT
0329            LD      DE,PEAK ;FIND NEW PT-PEAK
0330            CALL    DIFER
0331    BLD6    CMPTR   AMODE,0 ;IF MODE = 1 THEN NEXT K
0332            JP      Z,BLDX
0333            LDREL   ADFLST,3 ;ELSE IF SLOPE=DIR THEN
0334            LD      HL,(ADIR)
0335            CP      (HL)
0336            JR      NZ,BLD7
0337            STREL   AMODE,0 ;MODE=1
0338            JP      BLDX    ;NEXT K
0339    BLD7    CMPTR   ADIFF,0CH       ;ELSE DO
0340            JP      C,BLDX  ;IF (PT-PEAK)<0CH THEN NEXT K
0341            CMPTR   ATYPE,3 ;ELSE IF TYPE=3 THEN NEXT K
0342            JP      Z,BLDX  ;ELSE DO
0343            CP      01H     ;IF TYPE = 1
0344            JR      Z,BLD8  ;OR TYPE = 2
0345            CP      02H     ;THEN
0346            JR      NZ,BLD9
0347    BLD8    STREL   ATYPE,3 ;TYPE=3
0348            JP      BLDX    ;NEXT K
0349    BLD9    CP      04H     ;ELSE
0350            JR      NZ,BLD10 ;IF TYPE = 4 THEN
```

```
0351            STREL   ATYPE,01H ;TYPE=1
0352            STREL   ADIR,02H  ;DIR=-1
0353            JR      BLD11    ;ST=PEAK
0354    BLD10   STREL   ATYPE,02H ;ELSE TYPE=2
0355            STREL   ADIR,01H  ;DIR=1
0356    BLD11   LDPTR   AST,APEAK
0357    BLDX    LDPTR   ALAST,APT       ;LAST=PT
0358            LD      HL,ATYPE        ;INCREMENT PTR FOR NEXT K
0359            LD      DE,APEAK+1
0360    BNEXT   PUSH    HL
0361            LD      C,(HL)
0362            INC     HL
0363            LD      B,(HL)
0364            POP     HL
0365            INC     BC      ;INCREMEANT POINTER
0366            LD      (HL),C  ;STORE POINTER
0367            INC     HL
0368            LD      (HL),B
0369            INC     HL
0370            PUSH    HL
0371            AND     A
0372            SBC     HL,DE   ;DO WHILE NOT END OF PTR
0373            POP     HL
0374            JR      C,BNEXT ;END DO
0375            POP     BC
0376            DEC     B
0377            JP      NZ,BLD3
0378            LD      HL,(ADATA)      ;CHECK IF END OF INTERVAL
0379            LD      A,(HL)
0380            LD      HL,(ATEND)
0381            CP      (HL)    ;NEXT IF NOT END OF INTERVAL
0382            JR      NC,BNXINT
0383            LD      BC,5    ;SKIP RECORD POINTER
0384            LD      HL,(ADATA)
0385            ADD     HL,BC
0386            LD      (ADATA),HL
0387            JP      BLD0
0388    BNXINT  LD      HL,(AATYPE)     ;NEXT I
0389            INC     HL
0390            INC     HL
0391            INC     HL
0392            LD      (AATYPE),HL
0393            LD      HL,(ATEND)      ;SKIP NEXT TEND
0394            INC     HL
0395            LD      (ATEND),HL
0396            POP     BC
0397            DEC     B
0398            JP      NZ,BLOP ;NEXT I
0399    BDKEY   LD      IX,KEY+1        ;BUILD KEY
0400            LD      HL,TYPE
```

```
0401            LD      B,5     ;DO I = 1 TO 9 SKIP 2
0402    BDKEY1  LD      C,0
0403            LD      D,2
0404    BDKEY2  SLA     C
0405            SLA     C
0406            SLA     C
0407            SLA     C       ;SHIFT LEFT 4 BIT
0408            LD      A,C
0409            OR      (HL)    ;OR WITH TYPE
0410            LD      C,A
0411            INC     HL
0412            DEC     D
0413            JR      NZ,BDKEY2
0414            LD      (IX),C  ;STORE TYPE BPF1,1
0415            INC     IX
0416            DJNZ    BDKEY1
0417            DEC     IX      ;SET B0-B3 OF LAST = 0
0418            LD      A,0F0H
0419            AND     C
0420            LD      (IX),A
0421            LD      A,(RESULT)      ;STORE TIME
0422            CP      10H     ;IF TMAX<10H THEN
0423            JR      NC,BDKEY3  ;TIME=1
0424            LD      A,40H
0425            JR      BDKEY5
0426    BDKEY3  CP      18H     ;ELSE IF TMAX=>18H
0427            JR      NC,BDKEY4 ;THEN TIME=3
0428            LD      A,80H   ;ELSE TIME=2
0429            JR      BDKEY5
0430    BDKEY4  LD      A,0C0H
0431    BDKEY5  LD      (KEY),A
0432            LD      HL,(RESULT+1)   ;ANALYSE DOMINATE INTERVAL
0433            LD      (CAREA),HL
0434            LD      HL,(RESULT+7)
0435            LD      (CAREA+2),HL
0436            LD      HL,(RESULT+13)
0437            LD      (CAREA+4),HL
0438            CALL    CMPARE
0439            SLA     A
0440            SLA     A
0441            SLA     A
0442            SLA     A
0443            LD      HL,KEY
0444            OR      (HL)    ;STORE IMAX
0445            LD      (HL),A
0446            LD      HL,(RESULT+3)
0447            LD      (CAREA),HL
0448            LD      HL,(RESULT+9)
0449            LD      (CAREA+2),HL
0450            LD      HL,(RESULT+15)
```

```
0451            LD      (CAREA+4),HL
0452            CALL    CMPARE
0453            SLA     A
0454            SLA     A
0455            LD      HL,KEY
0456            OR      (HL)
0457            LD      (HL),A  ;2ND BPF
0458            LD      HL,(RESULT+5)
0459            LD      (CAREA),HL
0460            LD      HL,(RESULT+11)
0461            LD      (CAREA+2),HL
0462            LD      HL,(RESULT+17)
0463            LD      (CAREA+4),HL
0464            CALL    CMPARE
0465            LD      HL,KEY
0466            OR      (HL)
0467            LD      (HL),A
0468            POP     IY      ;RETURN
0469            POP     IX
0470            POP     AF
0471            POP     BC
0472            POP     DE
0473            POP     HL
0474            RET
0475    ;END SUBROUTINE
0476    ;SUBROUTINE SUMMING DATA
0477    ;SUM BPF1,BPF2,BPF3
0478    ;
0479    TOTAL   PUSH    HL
0480            PUSH    DE
0481            PUSH    BC
0482            PUSH    AF
0483            LD      IX,RESULT+7
0484            LD      C,2
0485            LD      B,3
0486            LD      IY,RESULT+1
0487            PUSH    IY
0488    TLOOP   LD      L,(IY)
0489            INC     IY
0490            LD      H,(IY)
0491            INC     IY
0492            LD      E,(IX)
0493            INC     IX
0494            LD      D,(IX)
0495            INC     IX
0496            AND     A
0497            ADD     HL,DE
0498            JP      C,OVFLOW
0499            DEC     IY
0500            DEC     IY
```

```
0501                LD      (IY),L
0502                INC     IY
0503                LD      (IY),H
0504                INC     IY
0505                DJNZ    TLOOP
0506                POP     IY
0507                PUSH    IY
0508                LD      B,3
0509                DEC     C
0510                JR      NZ,TLOOP
0511                POP     IY
0512                POP     AF
0513                POP     BC
0514                POP     DE
0515                POP     HL
0516                RET
0517    ;END
0518    OVFLOW  EQU     03CFH+0567H
0519    MULT16  EQU     0389H
0520    DVDE16  EQU     03A1H
0521    WKAREA  ORG     1900H
0522    T1      DS      1
0523    B1      DS      1
0524    B2      DS      1
0525    B3      DS      1
0526    TIME    DS      1
0527    BPF1    DS      1
0528    BPF2    DS      1
0529    BPF3    DS      1
0530    FLAG    DS      1
0531    RESULT  DS      19
0532    STADR   DS      2
0533    EOFAD   DS      2
0534    MAX     DS      3
0535    KEY     DS      6
0536    TEND    DS      4
0537    SAVESUM DS      18
0538    SUBBF1  DS      2
0539    SUMBF2  DS      2
0540    SUMBF3  DS      2
0541    SYSTACK DS      2
0542    WKSTACK DS      2
0543    BASE    DS      2
0544    MEAN    DS      1
0545    DATA    DS      1
0546    PEAK    DS      3
0547    DIR     DS      3
0548    ST      DS      3
0549    PT      DS      3
0550    LAST    DS      3
```

PROGRAM NAME = TSYS2.Z80 PAGE 0012

```
0551    DIFF     DS      3
0552    SLOPE    DS      3
0553    TYPE     DS      9
0554    MODE     DS      3
0555    DIFST    DS      6
0556    DIFLAST  DS      6
0557    AATYPE   DS      2
0558    ADATA    DS      2
0559    ATEND    DS      2
0560    ATYPE    DS      2
0561    AMODE    DS      2
0562    APT      DS      2
0563    AST      DS      2
0564    ALAST    DS      2
0565    ADIFST   DS      2
0566    ADFLST   DS      2
0567    ADIFF    DS      2
0568    ADIR     DS      2
0569    APEAK    DS      2
0570    CAREA    DS      6
0571    TABLE    DS      1800H
0572    CLKARA   DS      4
0573    WORKARA  DS      03FFH
0574             END     100H
```

PROGRAM NAME = THS.Z80 PAGE 0001

```
0001    ;THESIS MAIN PROGRAM
0002    ;
0003            ORG     1000H
0004    MAIN    CALL    SET8    ;SET 8255
0005            LD      HL,TABLE
0006            LD      A,1AH   ;CHECK IF TABLE IS INSTALL
0007            CP      (HL)
0008            CALL    NZ,SETTAB       ;IF NOT THEN SET UP
0009            INC     HL
0010            CP      (HL)    ;NEXT CHECK
0011            CALL    NZ,SETTAB
0012    ;ENTER LEARN MODE
0013    MLRN    XOR     A       ;CLEAR DISPLAY
0014            CALL    MSGE
0015            LD      A,1     ;SEND "LRN -"
0016            CALL    MSGE
0017            LD      HL,WORKARA      ;CLEAR VOICE INPUT
0018            LD      BC,03FFH
0019            LD      A,0FFH  ;PADD WITH FF
0020            CALL    PADD
0021    WAIT1   CALL    EXVC    ;WAIT FOR ANY INPUT
0022            JR      NZ,SAMPLE       ;IF VOICE INPUT THEN SAMPLE
0023            CALL    PANNEL  ;ELSE CHECK PANNEL
0024            LD      A,(DATA)
0025            CP      'E'     ;IF ESCAPE MODE
0026            CALL    Z,ESCAPE
0027            CP      'R'     ;IF MODE CHANGE TO RUN
0028            JP      Z,MRUN
0029            JR      WAIT1   ;ELSE JUMP TO WAIT1
0030    ;END OF WAIT LOOP
0031    ;SAMPLING AND STORE PROTOTYPE TABLE
0032    SAMPLE  CALL    READVC  ;READ VOICE INPUT
0033            JR      C,MLRN  ;IF VOICE DURATION TOO SMALL
0034            LD      A,0FFH  ;CLEAR KEYBOARD INPUT
0035            LD      (MEAN),A
0036    WAIT2   CALL    PANNEL  ;WAIT FOR KEYBOARD
0037            LD      A,(DATA)
0038            CP      'I'     ;IF IGNORE THEN EXIT
0039            JR      Z,MLRN
0040            CP      'E'     ;IF ESCAPE
0041            CALL    Z,ESCAPE
0042            CP      0DH     ;IF CARRIAGE RETURN
0043            JR      Z,CHK
0044            CP      0AH     ;CHECK IF < 0-9
0045            JR      NC,WAIT2
0046            LD      (MEAN),A        ;IF 0-9 THEN STORE
0047            JR      WAIT2   ;WAIT FOR 0DH
0048    ;INSERT TABLE
0049    CHK     LD      A,(MEAN)
0050            CP      0FFH    ;IF NON KEY IS PRESSED
```

```
0051            JR      Z,WAIT2
0052            LD      IX,TABLE+6      ;ELSE STORE DATA
0053            LD      IY,MEAN ;SEARCH FOR SAME MEANING
0054            LD      HL,5
0055            LD      DE,(TABLE+4)
0056            LD      B,1
0057            CALL    SEARCH
0058            JR      Z,SINS  ;IF FOUND THEN INSERT
0059            LD      HL,TABLE+6      ;ELSE INSERT FRONT
0060    SINS    CALL    INSTAB
0061            JR      MLRN
0062    ;END OF LEARN STATE
0063    ;
0064    ;DECODE STATE:::::::::::::::::::::::::::::::::::::::
0065    MRUN    CALL    STAT    ;CALCULATE MEAN,MAX,MIN
0066            LD      B,120   ;CHECK IF TABLE = FF
0067            LD      HL,STAB
0068            LD      A,0FFH
0069    CKSTAB  CP      (HL)
0070            JR      NZ,RCONTO       ;IF NON FF TABLE THEN CONT.
0071            INC     HL
0072            DJNZ    CKSTAB  ;ELSE IF ALL FF THEN
0073            JP      MLRN    ;EXIT TO LEARN MODE
0074    RCONTO  XOR     A       ;SEND LEARN STATUS
0075            CALL    MSGE    ;CLEAR DSPLAY
0076            LD      A,4     ;DISPLAY "   -  "
0077            CALL    MSGE
0078            LD      HL,WORKARA
0079            LD      BC,03FFH  ;CLEAR VOICE INPUT AREA
0080            LD      A,0FFH
0081            CALL    PADD
0082    RCONT   CALL    EXVC    ;EXAMINE VOICE
0083            JR      NZ,RXAMPL       ;IF YES READ VOICE
0084            CALL    PANNEL  ;ELSE EXAMINE KEYBOARD
0085            LD      A,(DATA)
0086            CP      'E'     ;CHECK IF ESCAPE
0087            CALL    Z,ESCAPE
0088            CP      'L'     ;CHECK IF MODE CHANGE
0089            JP      Z,MLRN  ;IF YES JUMP TO LEARN
0090            JR      RCONT
0091    RXAMPL  CALL    READVC  ;READ AND ANALYSE VOICE
0092            JR      C,RCONTO        ;IF NOISE THEN RETRY
0093            LD      HL,SCORE        ;CLEAR SCORE
0094            LD      BC,9
0095            XOR     A
0096            CALL    PADD
0097            LD      IX,FINAL
0098            LD      IY,SCORE
0099    GETL1   LD      HL,STAB ;GET LIST POSSIBLE DURATION
0100            LD      A,(IX)  ;GET TIME
```

```
0101            LD      BC,12    ;STAB RECORD LENGTH
0102            LD      D,10     ;PATTERN COUNTER
0103    GETL2   PUSH    HL       ;SAVE STAB POINTER
0104            CP      (HL)     ;CHECK TIME WITH TMAX
0105            JR      Z,GETL3  ;IF = THEN SCORE
0106            JR      NC,GETL4 ;IF > THEN NEXT
0107            INC     HL       ;ELSE CHECK WITH TMIN
0108            CP      (HL)
0109            JR      C,GETL4  ;IF < THEN NEXT
0110    GETL3   INC     (IY)     ;SCORE
0111    GETL4   POP     HL       ;RETURN STAB POINTER
0112            ADD     HL,BC    ;SKIP TABLE POINTER
0113            INC     IY       ;NEXT PATTERN SCORE
0114            DEC     D
0115            JR      NZ,GETL2         ;END DURATION CHECK++++
0116            LD      IY,SCORE         ;CHECK IF ANY LIST
0117            LD      B,10
0118            XOR     A
0119    CHKL1   CP      (IY)     ;CHECK LIST
0120            JR      NZ,MR1   ;IF ANY LIST THEN CONTINUE
0121            INC     IY
0122            DJNZ    CHKL1
0123            LD      A,0FFH   ;ELSE IF NO LIST THEN
0124            LD      (TDIF),A ;FIND POSSIBLE DURATION
0125            LD      A,(IX)   ;LOAD TIME
0126            LD      HL,STAB+2 ;MEAN ADDRESS
0127            LD      BC,12
0128            LD      E,0
0129            LD      D,10
0130    GETL5   SUB     (HL)     ;TIME-MEAN
0131            JR      NC,GETL6 ;ABSOLUTE VALUE
0132            CPL
0133            INC     A
0134    GETL6   PUSH    HL       ;SAVE POINTER
0135            LD      HL,TDIF
0136            CP      (HL)     ;COMPARE WITH LAST DIFFERENCE
0137            JR      NC,GETL7 ;IF > THEN NEXT
0138            LD      (HL),A   ;ELSE STORE
0139            LD      A,E
0140            LD      (VALUE),A ;STORE POSSIBLE PATTERN
0141    GETL7   LD      A,(IX)   ;RELOAD TIME
0142            POP     HL
0143            ADD     HL,BC
0144            INC     E
0145            DEC     D
0146            JR      NZ,GETL5 ;CHECK NEXT
0147            LD      A,(VALUE)
0148            JP      DSPRST   ;DISPLAY RESULT
0149    MR1     INC     IX       ;CONTINUE CHECK++++
0150            LD      IY,SCORE
```

```
0151            LD      HL,STAB+3 ;CHECK FOR BPF1-BPF3 RANGE
0152            LD      BC,12     ;STAB RECORD LENGTH
0153            LD      D,10
0154    MR2     XOR     A         ;CHECK FOR LIST-MEMBER
0155            CP      (IY)      ;IF SCORE = 0 THEN IGNORE
0156            JR      NZ,MR3
0157    MR2A    ADD     HL,BC     ;SKIP TABLE POINTER
0158            INC     IY        ;SKIP SCORE
0159            DEC     D
0160            JR      NZ,MR2    ;DO WHILE NOT E-O-T
0161            JP      MR7       ;THEN CONTINUE
0162    MR3     PUSH    HL        ;SAVE REGISTER
0163            PUSH    IX
0164            PUSH    BC
0165            LD      B,3       ;3 BPF
0166    MR4     LD      A,(IX)    ;LOAD BPF..
0167            CP      (HL)      ;CHECK WITH UP-LIM
0168            PUSH    AF
0169            INC     HL        ;SKIP PTR TO LOW-LIM
0170            POP     AF
0171            JR      Z,MR5     ;IF = THEN SCORE
0172            JR      NC,MR6    ;IF > THEN NEXT
0173            CP      (HL)      ;CHECK WITH LOW-LIM.
0174            JR      C,MR6     ;IF < THEN NEXT
0175    MR5     INC     (IY)      ;INCREMENT SCORE
0176    MR6     INC     HL        ;SKIP MEAN
0177            INC     HL
0178            INC     IX        ;SKIP TO NEXT BPF
0179            DJNZ    MR4
0180            POP     BC
0181            POP     IX
0182            POP     HL        ;RESTORE REGISTER
0183            JR      MR2A      ;NEXT PATTERN
0184    MR7     LD      IY,SCORE ;CHECK IF ANY LIST
0185            LD      B,10
0186            LD      E,0
0187            LD      C,0
0188            LD      D,0       ;FIND MAXIMUM SCORE
0189    MR8     LD      A,(IY)
0190            CP      C
0191            JR      C,MR9     ;IF > LAST MAX THEN STORE
0192            LD      C,A       ;STORE MAX
0193    MR9     INC     IY
0194            DJNZ    MR8
0195            LD      IY,SCORE          ;SUPPRESS LOWER SCORE
0196            LD      B,10
0197    MR10    LD      A,(IY)
0198            CP      C         ;IF < MAX THEN SCORE=0
0199            JR      NZ,MR11 ;IF <> THEN SUPPRESS
0200            INC     D         ;COUNT MAX SCORE
```

```
0201            LD      A,E
0202            LD      (VALUE),A
0203            JR      MR12    ;NEXT PATTERN
0204    MR11    XOR     A       ;SCORE=0
0205            LD      (IY),A
0206    MR12    INC     IY      ;NEXT SCORE
0207            INC     E
0208            DJNZ    MR10
0209            LD      A,D     ;CHECK IF > 1 LIST
0210            CP      1
0211            LD      A,(VALUE)
0212            JP      Z,DSPRST        ;IF 1 THEN DISPLAY RESULT
0213            LD      IY,SCORE        ;ELSE COMPARE MEAN
0214            LD      IX,FINAL+1
0215            LD      HL,RES  ;SET RESIDENCE = FF
0216            LD      BC,3
0217            LD      A,0FFH
0218            CALL    PADD
0219            LD      HL,STAB+5 ;BPF MEAN ADDRESS
0220            LD      BC,12
0221            LD      E,0
0222            LD      D,10
0223    MR13    XOR     A       ;SKIP 0 SCORE
0224            CP      (IY)
0225            JR      NZ,MR15
0226    MR14    ADD     HL,BC   ;SKIP TABLE POINTER
0227            INC     IY      ;SKIP SCORE
0228            INC     E       ;PATTERN COUNTER
0229            DEC     D
0230            JR      NZ,MR13
0231            JP      MR21
0232    MR15    PUSH    HL      ;SAVE REGISTER
0233            PUSH    IY
0234            PUSH    IX
0235            PUSH    BC
0236            PUSH    DE
0237            LD      IY,RES
0238            LD      DE,INTRN ;NEWCOMER POINTER
0239            LD      B,3     ;3BPF
0240    MR16    LD      A,(IX)  ;LOAD BPF VALUE
0241            SUB     (HL)    ;ABS(BF-MEAN)
0242            JR      NC,MR17
0243            CPL
0244            INC     A
0245    MR17    LD      (DE),A  ;STORE DIFFERENCE
0246            INC     IX      ;SKIP NEXT BPF
0247            INC     HL      ;SKIP NEXT MEAN
0248            INC     HL
0249            INC     HL
0250            INC     DE      ;NEXT INTRN
```

```
0251            DJNZ    MR16    ;DO I = 1 TO 3
0252            CALL    DSPLC ;FIND DIFFERENCE DISPLACEMENT
0253            LD      HL,(RES) ;COMPARE LEFT DIGIT
0254            LD      DE,(DSUM)
0255            AND     A
0256            SBC     HL,DE
0257            JR      C,MR20  ;IF NEW>OLD THEN NEXT
0258            JR      NZ,MR19A ;IF LESS THEN THEN RECOGNIZED
0259            LD      HL,(RES+2) ;COMPARE RIGHT DIGIT
0260            LD      DE,(DSUM+2)
0261            AND     A
0262            SBC     HL,DE
0263            JR      C,MR20  ;IF NEW>OLD THEN NEXT
0264    MR19A   LD      HL,(DSUM)  ;RECOGNIZED
0265            LD      (RES),HL
0266            LD      HL,(DSUM+2)
0267            LD      (RES+2),HL
0268            POP     DE
0269            PUSH    DE      ;LOAD PATTERN COUNTER
0270            LD      A,E
0271            LD      (VALUE),A
0272    MR20    POP     DE      ;RESTORE REGISTER
0273            POP     BC
0274            POP     IX
0275            POP     IY
0276            POP     HL
0277            JP      MR14
0278    MR21    LD      A,(VALUE)       ;DISPLAY PATTERN
0279    DSPRST  LD      HL,DTAB
0280            ADD     L
0281            JR      NC,DSPNX
0282            INC     H
0283    DSPNX   LD      L,A
0284            LD      A,(HL)
0285            CPL             ;DISPLAY RESULT
0286            LD      B,5
0287            CALL    DSP1
0288    WAITEND CALL    SCAN    ;WAIT FOR HIT ANY KEY
0289            JR      Z,WAITEND
0290            JP      RCONTO
0291    ;END MAIN+++++++
0292    ;
0293    ESCAPE  PUSH    AF
0294            LD      A,2     ;DISPLAY 'READY '
0295            CALL    MSGE
0296    ESC1    CALL    PANNEL  ;WAIT FOR KEYBOARD COMMAND
0297            LD      A,(DATA)
0298            CP      0DH
0299            JR      Z,ESCRET        ;RETURN IF CARRIAGE RTN
0300            CP      'S'
```

```
0301            CALL    Z,SETTAB        ;CHECK IF SETTAB
0302            JR      ESC1
0303    ESCRET  POP     AF
0304            RET
0305    ;END
0306    ;
0307    ;SUBROUTINE
0308    READVC  LD      HL,WORKARA      ;CLEAR INPUT AREA
0309            LD      DE,WORKARA+1023
0310            CALL    S0100   ;READINPUT VOICE
0311            INC     HL
0312            EX      DE,HL   ;SEND NEXT AVAILABLE AREA PTR.
0313            LD      HL,WORKARA
0314            CALL    SCREEN  ;SCREEN DATA
0315            PUSH    HL      ;SAVE SOURCE ADDR.
0316            PUSH    DE      ;SAVE DESTINATION ADDR
0317            PUSH    BC      ;TRANSFER EOD TO HL
0318            POP     HL
0319            PUSH    BC      ;SAVE EOD PTR.
0320            LD      BC,4
0321            AND     A
0322            SBC     HL,BC   ;CHECK TIME
0323            LD      A,(HL)  ;READ TIME
0324            POP     BC
0325            POP     DE
0326            POP     HL      ;RETURN INF.
0327            SUB     10      ;IF TIME<10
0328            RET     C       ;RETURN ON CARRY
0329            EX      DE,HL   ;DESTINATION TO HL
0330            LD      DE,WORKARA+1023
0331            CALL    CMPRS   ;FIND SUM
0332            CALL    TOTAL   ;FIND TOTAL
0333            LD      A,(RESULT)
0334            LD      (FINAL),A       ;STORE TIME
0335            LD      C,A
0336            LD      B,0
0337            LD      DE,(RESULT+1)
0338            LD      HL,0000H
0339            CALL    DVDE16  ;TOTAL BPF3/TIME
0340            LD      A,E
0341            LD      (FINAL+1),A
0342            LD      DE,(RESULT+3)
0343            LD      HL,0000H
0344            CALL    DVDE16
0345            LD      A,E     ;TOTAL BPF2/TIME
0346            LD      (FINAL+2),A
0347            LD      DE,(RESULT+5)
0348            LD      HL,0000H
0349            CALL    DVDE16
0350            LD      A,E
```

```
0351            LD      (FINAL+3),A
0352            AND     A
0353            RET
0354    ;END READVC
0355    ;SUBROUTINE CALCULATE STATISTICAL
0356    ;
0357    STAT    LD      HL,TABLE+6      ;RAW DATA ADDRESS
0358            PUSH    HL      ;SAVE HL
0359            LD      HL,STAB ;STATISTICAL RESULT TABLE
0360            LD      BC,119  ;LENGTH = 120 BYTE
0361            LD      A,0FFH  ;CLEAR TABLE *****
0362            CALL    PADD
0363    STAT0   LD      HL,SCOUNT       ;INITIALIZE DATA COUNT
0364            LD      BC,20
0365            XOR     A
0366            CALL    PADD
0367            POP     HL
0368            PUSH    HL
0369            INC     HL      ;SKIP PATTERN TYPE
0370            LD      A,(HL)  ;LOAD TIME
0371            LD      (STMIN),A
0372            INC     HL
0373            LD      A,(HL)  ;LOAD BPF1
0374            LD      (SMIN1),A
0375            INC     HL
0376            LD      A,(HL)  ;LOAD BPF2
0377            LD      (SMIN2),A
0378            INC     HL
0379            LD      A,(HL)  ;LOAD BPF3
0380            LD      (SMIN3),A
0381            POP     HL
0382    STAT1   LD      A,(HL)  ;LOAD PATTERN
0383            LD      (PAT),A ;STORE PATTERN
0384            CP      0FFH    ;CHECK IF END OF TABLE
0385            RET     Z       ;RETURN IF END
0386    STAT1A  INC     HL      ;LOAD TIME
0387            LD      A,(HL)
0388            LD      IX,STMAX ;RECORD TMAX
0389            CP      (IX)
0390            JR      C,STAT2
0391            LD      (IX),A
0392            JR      STAT3
0393    STAT2   LD      IX,STMIN ;RECORD TMIN
0394            CP      (IX)
0395            JR      NC,STAT3
0396            LD      (IX),A
0397    STAT3   LD      DE,(STIME)      ;FIND SIGMA(TIME)
0398            ADD     E
0399            JR      NC,STAT4
0400            INC     D
```

```
0401    STAT4   LD      E,A
0402            LD      (STIME),DE      ;STORE SUM
0403            INC     HL      ;CHECK BF1
0404            LD      A,(HL)
0405            LD      IX,SMAX1
0406            CP      (IX)
0407            JR      C,STAT5
0408            LD      (IX),A  ;STORE BF1 MAX
0409            JR      STAT6
0410    STAT5   LD      IX,SMIN1 ;CHECK LOW-LIM.
0411            CP      (IX)
0412            JR      NC,STAT6 ;IF X>MIN THEN NEXT
0413            LD      (IX),A  ;MIN = DATA
0414    STAT6   LD      DE,(SBF1)       ;FIND SUM BF1
0415            ADD     E
0416            JR      NC,STAT7
0417            INC     D
0418    STAT7   LD      E,A
0419            LD      (SBF1),DE       ;STORE SUM
0420            INC     HL      ;NEXT BF.
0421            LD      A,(HL)  ;EXAMINE BF2
0422            LD      IX,SMAX2
0423            CP      (IX)
0424            JR      C,STAT8 ;IF < THEN CHECK WITH LOW-LIM.
0425            LD      (IX),A  ;STORE MAX
0426            JR      STAT9
0427    STAT8   LD      IX,SMIN2
0428            CP      (IX)
0429            JR      NC,STAT9 ;IF > THEN FIND SUM
0430            LD      (IX),A
0431    STAT9   LD      DE,(SBF2)
0432            ADD     E
0433            JR      NC,STAT10
0434            INC     D
0435    STAT10  LD      E,A
0436            LD      (SBF2),DE
0437            INC     HL      ;EXAMINE BF3
0438            LD      A,(HL)
0439            LD      IX,SMAX3
0440            CP      (IX)
0441            JR      C,STAT11
0442            LD      (IX),A
0443            JR      STAT12
0444    STAT11  LD      IX,SMIN3
0445            CP      (IX)
0446            JR      NC,STAT12
0447            LD      (IX),A
0448    STAT12  LD      DE,(SBF3)
0449            ADD     E
0450            JR      NC,STAT13
```

```
0451            INC     D
0452    STAT13  LD      E,A
0453            LD      (SBF3),DE
0454            LD      IX,SCOUNT
0455            INC     (IX)    ;SCOUNT=*+1
0456            INC     HL      ;CHECK IF NEXT IS SAME PATTERN
0457            LD      A,(HL)
0458            LD      IX,PAT
0459            CP      (IX)
0460            JP      Z,STAT1A        ;IF SAME PATTERN THEN NEXT
0461            PUSH    HL      ;SAVE TABLE POINTER
0462            LD      BC,(SCOUNT)
0463            LD      B,0     ;FIND DENSITY
0464            LD      DE,(STIME)
0465            LD      HL,0000H
0466            CALL    DVDE16  ;TIME-MEAN
0467            LD      A,E
0468            LD      (STMEAN),A
0469            LD      DE,(SBF1)       ;FIND BF1-MEAN
0470            LD      HL,0000H
0471            CALL    DVDE16
0472            LD      A,E
0473            LD      (SMEAN1),A      ;BF1-MEAN
0474            LD      DE,(SBF2)
0475            LD      HL,0000H
0476            CALL    DVDE16
0477            LD      A,E
0478            LD      (SMEAN2),A      ;BF2-MEAN
0479            LD      DE,(SBF3)
0480            LD      HL,0000H
0481            CALL    DVDE16
0482            LD      A,E
0483            LD      (SMEAN3),A      ;BF3-MEAN
0484            LD      A,(PAT)
0485            CP      10
0486            JR      C,STAT14        ;CHECK IF RIGHT PATTERN
0487            POP     HL
0488            POP     HL
0489            JP      MLRN    ;IF ERROR EXIT TO LEARN MODE
0490    STAT14  LD      HL,STAB ;STORE TO STAT TBL
0491            LD      DE,12
0492            AND     A
0493    STAT15  JR      Z,STAT16
0494            ADD     HL,DE   ;SKIP POINTER
0495            DEC     A
0496            JR      STAT15
0497    STAT16  EX      DE,HL   ;DESTINATION TO DE
0498            LD      HL,STMAX
0499            LD      BC,12
0500            LDIR
```

```
0501            JP      STAT0
0502    ;END
0503    DSPLC   LD      HL,0000H        ;FIND DIFFERENCE DISPLACEMENT
0504            LD      (DSUM),HL       ;CLEAR XUM
0505            LD      (DSUM+2),HL
0506            LD      BC,(INTRN)      ;FIND DX^2+DY^2+DZ^2
0507            LD      B,0
0508            PUSH    BC
0509            POP     DE
0510            LD      HL,0000H
0511            CALL    MULT16  ;FIND X^2
0512            LD      (DSUM),HL
0513            LD      BC,(INTRN+1)
0514            LD      B,0
0515            PUSH    BC
0516            POP     DE
0517            LD      HL,0000H
0518            CALL    MULT16
0519            LD      DE,(DSUM+2)
0520            AND     A
0521            ADD     HL,DE   ;X^2+Y^2
0522            LD      (DSUM+2),HL
0523            JR      NC,DPLC1
0524            LD      HL,(DSUM)
0525            INC     HL
0526            LD      (DSUM),HL
0527    DPLC1   LD      BC,(INTRN+2)
0528            LD      B,0
0529            PUSH    BC
0530            POP     DE
0531            LD      HL,0000H
0532            CALL    MULT16
0533            LD      DE,(DSUM+2)     ;X^2+Y^2+Z^2
0534            AND     A
0535            ADD     HL,DE
0536            LD      (DSUM+2),HL
0537            RET     NC
0538            LD      HL,(DSUM)
0539            INC     HL
0540            LD      (DSUM),HL
0541            RET
0542            JP      0
0543    EXVC    EQU     0382H
0544    S0100   EQU     05DAH
0545    PANNEL  EQU     03E9H+0567H
0546    SCREEN  EQU     04AEH
0547    DSP1    EQU     0220H
0548    DTAB    EQU     01EFH
0549    PADD    EQU     03B8H
0550    SCAN    EQU     02ABH
```

PROGRAM NAME = THS. Z80 PAGE 0012

```
0551    SEARCH  EQU     02FFH
0552    MSGE    EQU     0375H+0567H
0553    DELTAB  EQU     0347H+0567H
0554    INSTAB  EQU     0311H+0567H
0555    SETTAB  EQU     02EDH+0567H
0556    BKEY    EQU     01D6H+08A6H
0557    CMPRS   EQU     0667H
0558    SET8    EQU     0354H
0559    MULT16  EQU     0389H
0560    DVDE16  EQU     03A1H
0561    WKAREA  ORG     1900H
0562    TOTAL   EQU     0494H+08A6H
0563    DUMP    EQU     03EFH
0564    T1      DS      1
0565    B1      DS      1
0566    B2      DS      1
0567    B3      DS      1
0568    TIME    DS      1
0569    BPF1    DS      1
0570    BPF2    DS      1
0571    BPF3    DS      1
0572    FLAG    DS      1
0573    RESULT  DS      19
0574    STADR   DS      2
0575    EOFAD   DS      2
0576    MAX     DS      3
0577    KEY     DS      6
0578    TEND    DS      4
0579    SAVESUM DS      18
0580    SUMBF1  DS      2
0581    SUMBF2  DS      2
0582    SUMBF3  DS      2
0583    SYSTACK DS      2
0584    WKSTACK DS      2
0585    BASE    DS      2
0586    MEAN    DS      1
0587    DATA    DS      1
0588    PEAK    DS      3
0589    DIR     DS      3
0590    ST      DS      3
0591    PT      DS      3
0592    LAST    DS      3
0593    DIFF    DS      3
0594    SLOPE   DS      3
0595    TYPE    DS      9
0596    MODE    DS      3
0597    DIFST   DS      6
0598    DIFLAST DS      6
0599    AATYPE  DS      2
0600    ADATA   DS      2
```

```
0601    ATEND   DS      2
0602    ATYPE   DS      2
0603    AMODE   DS      2
0604    APT     DS      2
0605    AST     DS      2
0606    ALAST   DS      2
0607    ADIFST  DS      2
0608    ADFLST  DS      2
0609    ADIFF   DS      2
0610    ADIR    DS      2
0611    APEAK   DS      2
0612    CAREA   DS      6
0613    TABLE   DS      1200H
0614    TDIF    DS      1
0615    VALUE   DS      1
0616    PAT     DS      1
0617    SCOUNT  DS      1
0618    SBF1    DS      2
0619    SBF2    DS      2
0620    SBF3    DS      2
0621    STIME   DS      2
0622    STMAX   DS      1
0623    STMIN   DS      1
0624    STMEAN  DS      1
0625    SMAX1   DS      1
0626    SMIN1   DS      1
0627    SMEAN1  DS      1
0628    SMAX2   DS      1
0629    SMIN2   DS      1
0630    SMEAN2  DS      1
0631    SMAX3   DS      1
0632    SMIN3   DS      1
0633    SMEAN3  DS      1
0634    RES     DS      4
0635    INTRN   DS      3
0636    STAB    DS      120
0637    SCORE   DS      10
0638    FINAL   DS      4
0639    SMAX    DS      2
0640    SMIN    DS      2
0641    DSUM    DS      4
0642    UNUSE   DS      1363
0643    CLKARA  DS      4
0644    WORKARA DS      03FFH
0645            END     100H
```

ภาคผนวก ค.
ตัวอย่างข้อมูลเสียงและข้อมูลวิเคราะห์

VOCABULARY : 0

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	37	00	00	00	00	00
31A4	37	10	00	00	00	00
31AA	10	20	00	00	00	00
31B0	06	30	00	00	00	00
31B6	02	40	00	00	01	00
31BC	02	50	01	00	05	00
31C2	03	60	08	09	10	00
31C8	03	70	0E	26	15	00
31CE	03	80	11	2F	15	00
31D4	04	90	13	27	19	00
31DA	03	A0	14	1F	18	00
31E0	03	B0	12	1A	15	00
31E6	04	C0	0F	17	19	00
31EC	04	D0	0C	17	1C	00
31F2	04	E0	0A	19	1D	00
31F8	02	F0	08	1F	21	00
31FE	03	00	07	20	23	00
3204	03	10	06	18	2E	00
320A	03	20	05	12	3A	00
3210	03	30	05	0E	49	00
3216	03	40	03	0B	59	00
321C	03	50	05	0B	63	00
3222	03	60	06	0A	6F	00
3228	03	70	08	0B	6F	00
322E	01	80	06	06	69	00
3234	01	90	03	03	62	00
323A	01	A0	02	01	4D	00
3240	01	B0	02	01	41	00
3246	01	C0	02	01	31	00
324C	01	D0	03	01	23	00
3252	02	E0	05	01	19	00
3258	02	F0	05	02	18	00
325E	01	00	05	03	0F	00
3264	02	10	05	04	09	00
326A	02	20	04	02	09	00
3270	02	30	03	01	07	00
3276	02	40	02	01	08	00
327C	04	50	00	00	04	00
3282	02	60	00	00	03	00
3288	04	70	00	00	01	00
328E	01	80	00	00	01	00
3294	06	90	00	00	00	00
329A	01	A0	00	00	00	00
32A0	03	B0	00	00	01	00
32A6	01	C0	00	00	00	FF

VOCABULARY : 0

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	31	30	00	00	00	00
31A4	28	40	00	00	00	00
31AA	02	50	00	00	01	00
31B0	03	60	02	05	03	00
31B6	04	70	07	10	08	00
31BC	03	80	09	0F	09	00
31C2	03	90	0C	0D	0C	00
31C8	03	A0	0D	11	0D	00
31CE	04	B0	0B	18	12	00
31D4	03	C0	09	17	1B	00
31DA	03	D0	07	13	21	00
31E0	03	E0	06	10	23	00
31E6	03	F0	06	0F	1B	00
31EC	07	00	05	11	13	00
31F2	03	10	05	10	18	00
31F8	03	20	05	0E	15	00
31FE	03	30	04	0C	0F	00
3204	03	40	04	0A	13	00
320A	04	50	05	0A	12	00
3210	09	60	06	0A	11	00
3216	03	70	09	0A	12	00
321C	04	80	0F	09	0F	00
3222	02	90	0A	05	09	00
3228	02	A0	06	03	05	00
322E	01	B0	05	01	05	00
3234	01	C0	05	01	05	00
323A	01	D0	05	01	01	00
3240	08	E0	05	01	02	00
3246	06	F0	05	01	02	00
324C	01	00	04	01	03	00
3252	03	10	02	00	03	00
3258	03	20	02	01	03	00
325E	02	30	01	00	01	00
3264	02	40	01	00	03	00
326A	01	50	00	00	01	00
3270	07	60	00	00	00	00
3276	04	70	00	00	00	00
327C	07	80	00	00	00	FF

VOCABULARY : 1

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	11	B0	00	00	01	00
31A4	0E	C0	01	00	02	00
31AA	01	D0	01	00	03	00
31B0	02	E0	03	02	07	00
31B6	02	F0	07	03	06	00
31BC	06	00	0E	12	0B	00
31C2	03	10	14	2E	0F	00
31C8	06	20	1D	3C	0B	00
31CE	04	30	21	2E	0D	00
31D4	03	40	26	22	0F	00
31DA	03	50	2E	19	0F	00
31E0	0E	60	2E	14	11	00
31E6	04	70	32	10	13	00
31EC	06	80	32	0D	14	00
31F2	02	90	2F	0B	16	00
31F8	02	A0	2E	0A	19	00
31FE	02	B0	2B	0A	19	00
3204	06	C0	24	09	1D	00
320A	02	D0	1B	07	1D	00
3210	02	E0	15	05	1C	00
3216	02	F0	15	04	1C	00
321C	03	00	14	04	1D	00
3222	05	10	10	03	24	00
3228	05	20	0D	03	2E	00
322E	04	30	0E	02	3B	00
3234	02	40	10	02	4B	00
323A	03	50	0C	02	59	00
3240	06	60	0D	03	65	00
3246	04	70	0D	03	6C	00
324C	07	80	0B	02	59	00
3252	02	90	08	02	40	00
3258	10	A0	06	02	24	00
325E	06	B0	04	01	0F	00
3264	07	C0	02	01	06	00
326A	02	D0	01	00	03	00
3270	01	E0	00	00	03	00
3276	01	F0	00	00	01	FF

VOCABULARY : 1

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	01	B0	00	00	03	00
31A4	04	C0	00	00	04	00
31AA	02	D0	02	01	04	00
31B0	06	E0	07	04	09	00
31B6	04	F0	0C	0B	09	00
31BC	03	00	10	0F	09	00
31C2	03	10	13	10	0A	00
31C8	04	20	18	10	0B	00
31CE	03	30	1A	0F	0E	00
31D4	03	40	1F	0D	0F	00
31DA	05	50	21	0B	0F	00
31E0	04	60	22	0B	0F	00
31E6	04	70	22	0A	11	00
31EC	05	80	22	09	14	00
31F2	05	90	20	08	14	00
31F8	03	A0	20	08	16	00
31FE	02	B0	1F	08	17	00
3204	03	C0	17	06	1B	00
320A	01	D0	12	05	21	00
3210	02	E0	0E	03	23	00
3216	02	F0	0C	02	24	00
321C	05	00	0A	02	1D	00
3222	03	10	09	02	21	00
3228	06	20	0A	02	29	00
322E	01	30	0A	01	33	00
3234	02	40	08	01	40	00
323A	03	50	05	00	4B	00
3240	02	60	03	01	53	00
3246	01	70	01	01	59	00
324C	02	80	01	00	5B	00
3252	01	90	01	00	59	00
3258	01	A0	01	00	51	00
325E	01	B0	00	00	49	00
3264	03	C0	00	00	38	00
326A	02	D0	00	00	2B	00
3270	01	E0	00	00	1E	00
3276	09	F0	00	00	18	FF

VOCABULARY : 2

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	40	70	00	00	01	00
31A4	38	B0	00	00	00	00
31AA	36	90	00	00	00	00
31B0	21	A0	00	00	00	00
31B6	08	B0	00	00	00	00
31BC	05	C0	01	03	03	00
31C2	05	D0	03	0D	08	00
31C8	05	E0	06	17	0C	00
31CE	04	F0	08	21	09	00
31D4	04	00	0A	31	0D	00
31DA	04	10	0A	45	0C	00
31E0	04	20	0B	45	10	00
31E6	04	30	0B	3C	14	00
31EC	07	40	0B	33	13	00
31F2	05	50	0A	2C	13	00
31F8	04	60	0A	28	13	00
31FE	04	70	09	25	13	00
3204	04	80	08	25	11	00
320A	04	90	07	25	14	00
3210	04	A0	06	25	15	00
3216	04	B0	05	22	17	00
321C	03	C0	05	1E	1F	00
3222	05	D0	04	1D	26	00
3228	04	E0	04	1B	33	00
322E	04	F0	03	17	3B	00
3234	0F	00	03	15	45	00
323A	05	10	02	12	48	00
3240	06	20	01	0D	48	00
3246	05	30	01	0B	39	00
324C	02	40	01	0A	29	00
3252	02	50	01	0B	23	00
3258	02	60	03	0B	19	00
325E	02	70	03	07	14	00
3264	03	80	04	05	0E	00
326A	02	90	03	03	0B	00
3270	01	A0	03	03	09	00
3276	0A	B0	01	03	05	00
327C	0C	C0	01	02	05	00
3282	09	D0	00	01	05	00
3288	05	E0	00	00	03	00
328E	06	F0	00	00	00	00
3294	01	10	00	00	00	FF

VOCABULARY : 2

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	2E	E0	00	00	01	00
31A4	32	F0	00	00	01	00
31AA	3B	00	00	00	01	00
31B0	3B	10	00	00	01	00
31B6	3B	20	00	00	00	00
31BC	30	30	00	00	01	00
31C2	06	40	00	00	01	00
31C8	04	50	01	03	01	00
31CE	04	60	03	11	09	00
31D4	04	70	0B	1D	0C	00
31DA	04	80	0D	2C	0F	00
31E0	04	90	0C	41	11	00
31E6	04	A0	0B	40	17	00
31EC	08	B0	09	3C	1C	00
31F2	05	C0	08	37	26	00
31F8	04	D0	08	31	2B	00
31FE	05	E0	07	29	29	00
3204	04	F0	06	23	21	00
320A	04	00	06	23	13	00
3210	04	10	05	27	11	00
3216	04	20	05	26	12	00
321C	05	30	05	25	10	00
3222	04	40	05	25	0D	00
3228	04	50	05	22	0F	00
322E	05	60	04	21	10	00
3234	04	70	05	22	0F	00
323A	04	80	04	25	0F	00
3240	0C	90	03	1F	0B	00
3246	04	A0	02	18	0C	00
324C	08	B0	01	11	0B	00
3252	02	C0	01	0A	09	00
3258	01	D0	01	07	04	00
325E	01	E0	01	05	03	00
3264	01	F0	01	04	02	00
326A	01	00	01	03	01	00
3270	04	10	03	03	01	00
3276	02	20	03	03	01	00
327C	04	30	03	05	01	00
3282	01	40	03	07	04	00
3288	08	50	01	07	05	00
328E	01	60	01	04	06	00
3294	02	70	00	01	04	00
329A	03	80	00	00	01	00
32A0	01	90	00	00	00	00
32A6	02	A0	00	01	01	00
32AC	03	B0	00	00	01	FF

VOCABULARY : 3

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	04	F0	00	00	00	00
31A4	02	00	01	03	01	00
31AA	05	10	07	24	11	00
31B0	05	20	0E	36	12	00
31B6	05	30	10	51	0D	00
31BC	07	40	11	6D	13	00
31C2	05	50	13	7F	13	00
31C8	06	60	11	87	0F	00
31CE	05	70	12	81	13	00
31D4	05	80	13	74	1D	00
31DA	05	90	13	67	24	00
31E0	05	A0	12	63	2D	00
31E6	05	B0	11	62	34	00
31EC	05	C0	11	59	3C	00
31F2	05	D0	0E	55	41	00
31F8	10	E0	0A	55	47	00
31FE	05	F0	09	55	4B	00
3204	05	00	07	4F	51	00
320A	05	10	06	56	54	00
3210	09	20	05	61	55	00
3216	03	30	05	5C	4F	00
321C	03	40	04	3F	48	00
3222	08	50	03	35	41	00
3228	05	60	03	2B	3B	00
322E	05	70	01	1B	39	00
3234	02	80	01	0F	33	00
323A	02	90	00	07	2F	00
3240	01	A0	00	04	23	00
3246	03	B0	00	02	1D	00
324C	0A	C0	01	01	15	00
3252	05	D0	01	01	10	00
3258	02	E0	01	01	0C	00
325E	02	F0	01	00	08	00
3264	01	00	01	01	08	00
326A	06	10	01	01	03	00
3270	02	20	01	01	03	00
3276	01	30	00	00	01	00
327C	01	40	00	00	01	00
3282	04	50	00	00	00	00
3288	03	60	00	00	00	00
328E	02	70	00	00	00	FF

VOCABULARY : 3

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	3E	E0	00	00	01	00
31A4	38	F0	00	00	01	00
31AA	26	00	00	00	01	00
31B0	01	10	00	00	00	00
31B6	03	20	03	0D	01	00
31BC	07	30	09	22	04	00
31C2	04	40	0E	34	05	00
31C8	04	50	0F	50	05	00
31CE	05	60	10	6A	07	00
31D4	0B	70	0F	7B	07	00
31DA	05	80	0F	86	08	00
31E0	05	90	0F	8C	09	00
31E6	04	A0	0F	89	0B	00
31EC	04	B0	0D	82	08	00
31F2	04	C0	0C	7F	05	00
31F8	04	D0	0C	71	05	00
31FE	05	E0	0A	59	09	00
3204	05	F0	08	4B	0C	00
320A	05	00	09	40	12	00
3210	07	10	09	3D	1D	00
3216	05	20	09	3B	26	00
321C	0F	30	08	39	31	00
3222	04	40	08	39	38	00
3228	06	50	07	33	3D	00
322E	0B	60	05	25	41	00
3234	0B	70	04	19	3D	00
323A	08	80	02	0F	2D	00
3240	03	90	02	07	1E	00
3246	02	A0	01	05	1B	00
324C	05	B0	01	02	11	00
3252	01	C0	02	01	0F	00
3258	03	D0	02	01	0B	00
325E	02	E0	02	01	05	00
3264	02	F0	02	00	03	00
326A	02	00	02	00	05	00
3270	01	10	02	01	05	00
3276	11	20	01	00	01	00
327C	02	30	00	00	03	00
3282	01	40	00	00	01	00
3288	08	50	00	00	00	00
328E	0A	60	00	00	01	00
3294	01	70	00	00	01	FF

VOCABULARY : 4

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	36	B0	00	00	00	00
31A4	3F	90	00	00	01	00
31AA	33	A0	00	00	01	00
31B0	32	B0	00	00	00	00
31B6	1D	C0	00	00	00	00
31BC	11	D0	00	00	00	00
31C2	03	E0	01	00	00	00
31C8	03	F0	04	03	05	00
31CE	03	00	07	08	08	00
31D4	03	10	08	08	08	00
31DA	02	20	08	08	0B	00
31E0	02	30	08	08	0B	00
31E6	02	40	08	07	0F	00
31EC	02	50	07	06	11	00
31F2	02	60	06	06	13	00
31F8	03	70	05	05	18	00
31FE	03	80	05	04	1D	00
3204	02	90	06	04	20	00
320A	04	A0	06	02	1F	00
3210	09	B0	05	03	22	00
3216	0A	C0	06	03	2D	00
321C	08	D0	06	03	41	00
3222	06	E0	07	03	51	00
3228	06	F0	05	03	5F	00
322E	0D	00	04	02	69	00
3234	04	10	04	01	5B	00
323A	02	20	03	01	44	00
3240	08	30	02	01	39	00
3246	0A	40	01	01	2A	00
324C	03	50	01	01	20	00
3252	05	60	01	01	24	00
3258	02	70	01	01	23	00
325E	0C	80	00	00	1D	00
3264	03	90	00	00	11	00
326A	08	A0	00	00	05	00
3270	02	B0	00	00	04	00
3276	01	C0	00	00	02	00
327C	01	D0	00	00	03	00
3282	01	E0	00	00	01	FF

VOCABULARY : 4

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	39	00	00	00	00	00
31A4	33	10	00	00	00	00
31AA	34	20	01	00	01	00
31B0	38	30	01	00	00	00
31B6	35	40	00	00	01	00
31BC	16	50	00	00	00	00
31C2	02	60	02	00	01	00
31C8	03	70	09	04	09	00
31CE	06	80	0D	07	0D	00
31D4	04	90	10	08	14	00
31DA	03	A0	12	08	13	00
31E0	02	B0	13	09	10	00
31E6	02	C0	16	09	14	00
31EC	02	D0	17	08	13	00
31F2	02	E0	15	08	13	00
31F8	02	F0	15	08	15	00
31FE	03	00	17	08	19	00
3204	03	10	18	08	19	00
320A	04	20	19	07	1D	00
3210	03	30	18	06	20	00
3216	0C	40	17	06	21	00
321C	06	50	16	05	25	00
3222	03	60	19	06	2D	00
3228	03	70	17	05	43	00
322E	03	80	16	05	55	00
3234	03	90	13	04	69	00
323A	03	A0	12	04	6F	00
3240	04	B0	10	04	69	00
3246	04	C0	0F	03	4D	00
324C	03	D0	10	01	2E	00
3252	03	E0	0E	01	24	00
3258	03	F0	0B	01	1F	00
325E	06	00	0B	01	11	00
3264	08	10	07	01	0F	00
326A	03	20	05	00	0B	00
3270	01	30	03	00	09	00
3276	05	40	02	00	04	00
327C	0C	50	01	00	01	00
3282	03	60	00	00	00	00
3288	01	70	00	00	00	00
328E	03	80	00	00	01	FF

VOCABULARY : 5

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	02	80	01	02	00	00
31A4	02	90	03	0D	09	00
31AA	07	A0	10	19	0C	00
31B0	05	B0	1B	23	0D	00
31B6	06	C0	22	2D	0F	00
31BC	05	D0	25	34	0D	00
31C2	05	E0	26	3B	0B	00
31C8	07	F0	28	3D	0B	00
31CE	05	00	26	43	0D	00
31D4	07	10	26	46	0C	00
31DA	06	20	24	4B	0D	00
31E0	09	30	23	4C	0B	00
31E6	04	40	21	55	0C	00
31EC	05	50	1F	5E	0B	00
31F2	0E	60	1C	6D	0B	00
31F8	04	70	19	7D	09	00
31FE	07	80	19	83	05	00
3204	04	90	13	65	03	00
320A	05	A0	0F	45	04	00
3210	05	B0	0B	30	04	00
3216	05	C0	08	2A	05	00
321C	03	D0	07	1D	0C	00
3222	04	E0	06	18	11	00
3228	02	F0	06	19	1C	00
322E	05	00	06	1E	28	00
3234	01	10	04	1A	2B	00
323A	03	20	03	17	1D	00
3240	01	30	02	12	19	00
3246	0A	40	02	0E	11	00
324C	02	50	02	0D	09	00
3252	05	60	01	0D	05	00
3258	01	70	00	0A	03	00
325E	01	80	00	05	01	FF

VOCABULARY : 5

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	02	F0	00	01	01	00
31A4	02	00	01	05	09	00
31AA	02	10	04	0D	0D	00
31B0	06	20	0B	16	0B	00
31B6	04	30	0E	1D	0D	00
31BC	05	40	13	23	0B	00
31C2	03	50	18	2A	0D	00
31C8	03	60	1B	32	0D	00
31CE	03	70	1D	3D	0D	00
31D4	04	80	21	44	0D	00
31DA	03	90	25	4F	0D	00
31E0	05	A0	2A	56	0C	00
31E6	04	B0	2B	61	0B	00
31EC	04	C0	29	6F	0B	00
31F2	07	D0	24	77	0B	00
31F8	05	E0	1E	81	09	00
31FE	08	F0	1A	8B	09	00
3204	05	00	16	8F	09	00
320A	06	10	12	81	08	00
3210	04	20	0F	5A	06	00
3216	02	30	0D	3E	07	00
321C	01	40	0D	33	05	00
3222	02	50	0D	2C	09	00
3228	02	60	0B	21	0B	00
322E	02	70	09	18	13	00
3234	05	80	08	17	24	00
323A	03	90	07	13	25	00
3240	05	A0	05	0C	1C	00
3246	02	B0	05	08	19	00
324C	08	C0	03	06	11	00
3252	0B	D0	03	08	11	00
3258	0A	E0	03	06	0D	00
325E	01	F0	01	02	0B	00
3264	06	00	00	00	05	00
326A	01	10	00	00	05	00
3270	07	20	00	00	02	00
3276	0A	30	00	00	01	00
327C	04	40	00	00	00	00
3282	0B	50	00	00	00	00
3288	02	60	00	00	00	00
328E	01	70	00	00	00	00
3294	01	80	00	00	00	FF

VOCABULARY : 6

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	01	00	00	00	01	00
31A4	01	10	00	00	01	00
31AA	03	20	00	00	03	00
31B0	02	30	00	01	06	00
31B6	02	40	00	03	07	00
31BC	04	50	02	09	0A	00
31C2	04	60	07	13	0B	00
31C8	04	70	09	1A	0B	00
31CE	04	80	0A	1F	0B	00
31D4	03	90	0C	29	0B	00
31DA	04	A0	0E	44	0F	00
31E0	04	B0	0E	60	18	00
31E6	03	C0	0D	33	28	00
31EC	04	D0	0C	31	38	00
31F2	12	E0	0A	3A	50	00
31F8	04	F0	07	74	3F	00
31FE	06	00	04	7D	68	00
3204	06	10	01	63	58	00
320A	0F	20	00	43	48	00
3210	03	30	00	29	38	00
3216	01	40	00	17	29	FF

VOCABULARY : 6

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	02	80	00	00	01	00
31A4	01	90	00	00	01	00
31AA	01	A0	00	00	07	00
31B0	01	B0	00	01	0B	00
31B6	01	C0	00	02	0C	00
31BC	03	D0	02	16	0B	00
31C2	04	E0	08	21	0F	00
31C8	04	F0	0B	35	0B	00
31CE	06	00	0E	52	0C	00
31D4	04	10	0F	52	09	00
31DA	03	20	0E	45	0C	00
31E0	03	30	0E	49	09	00
31E6	0B	40	0E	5E	0C	00
31EC	04	50	0D	6C	0D	00
31F2	04	60	0B	6B	0F	00
31F8	03	70	08	4A	13	00
31FE	08	80	05	31	13	00
3204	0A	90	02	22	19	00
320A	04	A0	00	17	15	00
3210	02	B0	00	0E	14	00
3216	02	C0	00	07	0D	FF

VOCABULARY : 7

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	0E	70	00	00	01	00
31A4	04	80	04	01	03	00
31AA	04	90	0D	05	09	00
31B0	07	A0	18	07	08	00
31B6	05	B0	22	0A	09	00
31BC	03	C0	2A	11	09	00
31C2	05	D0	37	1B	09	00
31C8	05	E0	3C	25	0D	00
31CE	05	F0	3D	1F	11	00
31D4	04	00	32	18	19	00
31DA	04	10	29	13	20	00
31E0	19	20	1B	0E	2B	00
31E6	05	30	0D	08	29	00
31EC	01	40	06	03	24	00
31F2	0A	50	02	01	1B	FF

VOCABULARY : 7

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	2D	03	01	00	01	00
31A4	0E	10	01	01	01	00
31AA	04	20	0C	07	08	00
31B0	07	30	17	0A	09	00
31B6	07	40	23	0D	09	00
31BC	03	50	2E	14	07	00
31C2	07	60	37	18	0A	00
31C8	05	70	3B	1B	09	00
31CE	06	80	38	1C	0F	00
31D4	03	90	33	1E	11	00
31DA	04	A0	2C	1C	0F	00
31E0	05	B0	23	18	15	00
31E6	0E	C0	17	12	1D	00
31EC	04	D0	0B	0A	1D	00
31F2	04	E0	05	04	15	00
31F8	10	F0	01	01	11	00
31FE	01	00	00	00	0C	FF

VOCABULARY : 8

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	07	E0	03	08	01	00
31A4	05	F0	08	27	03	00
31AA	04	00	0B	3D	03	00
31B0	05	10	0F	4E	03	00
31B6	09	20	12	5B	05	00
31BC	05	30	13	65	03	00
31C2	05	40	18	71	03	00
31C8	05	50	1A	7D	03	00
31CE	05	60	1D	86	03	00
31D4	05	70	20	8C	03	00
31DA	06	80	20	91	06	00
31E0	06	90	21	95	07	00
31E6	08	A0	21	96	08	00
31EC	05	B0	21	97	09	00
31F2	05	C0	22	94	09	00
31F8	04	D0	22	7C	09	00
31FE	03	E0	22	60	09	00
3204	05	F0	1D	4C	09	00
320A	06	00	1A	3D	0D	00
3210	05	10	1A	31	14	00
3216	05	20	19	2A	19	00
321C	04	30	1A	2A	1E	00
3222	04	40	1C	2F	26	00
3228	06	50	19	37	2F	00
322E	05	60	12	3A	36	00
3234	01	70	09	2B	2F	00
323A	03	80	04	1B	29	00
3240	0C	90	01	0F	20	FF

VOCABULARY : 8

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	04	40	02	0E	04	00
31A4	05	50	0B	48	05	00
31AA	08	60	13	5D	09	00
31B0	06	70	1A	6A	09	00
31B6	06	80	23	75	0B	00
31BC	06	90	32	81	09	00
31C2	05	A0	3F	8A	09	00
31C8	0B	B0	45	90	09	00
31CE	06	C0	44	95	0D	00
31D4	06	D0	3D	99	0C	00
31DA	05	E0	3F	9B	0D	00
31E0	05	F0	41	9C	0F	00
31E6	05	00	39	9E	12	00
31EC	05	10	31	90	13	00
31F2	05	20	30	79	1B	00
31F8	04	30	2F	62	1F	00
31FE	05	40	2D	56	23	00
3204	04	50	2B	4A	2D	00
320A	06	60	2C	41	33	00
3210	04	70	2B	38	41	00
3216	05	80	24	41	4B	00
321C	05	90	20	40	55	00
3222	06	A0	1C	3D	58	00
3228	04	B0	16	36	59	00
322E	0A	C0	0D	26	51	00
3234	01	D0	05	16	43	00
323A	02	E0	01	0C	39	00
3240	01	F0	00	06	2D	FF

VOCABULARY : 9

Pattern no. 1

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	05	70	00	00	00	00
31A4	01	80	01	02	01	00
31AA	06	90	0B	09	08	00
31B0	06	A0	13	0F	08	00
31B6	05	B0	22	18	09	00
31BC	04	C0	30	23	09	00
31C2	06	D0	39	2D	08	00
31C8	05	E0	3E	39	08	00
31CE	07	F0	3F	3C	05	00
31D4	05	00	45	49	05	00
31DA	07	10	49	47	06	00
31E0	0C	20	4D	4E	07	00
31E6	05	30	50	4B	05	00
31EC	05	40	4D	51	08	00
31F2	04	50	41	58	05	00
31F8	07	60	36	64	05	00
31FE	05	70	2B	73	05	00
3204	04	80	25	7F	01	00
320A	05	90	24	8A	04	00
3210	0C	A0	22	8F	01	00
3216	05	B0	1F	88	03	00
321C	04	C0	20	61	02	00
3222	09	D0	20	49	02	00
3228	05	E0	20	47	03	00
322E	04	F0	1D	3F	07	00
3234	04	00	1A	61	09	00
323A	04	10	16	44	0F	00
3240	04	20	10	2E	1D	00
3246	04	30	0B	21	29	00
324C	05	40	06	23	31	00
3252	02	50	03	25	30	00
3258	01	60	01	19	21	00
325E	03	70	01	0F	19	00
3264	01	80	00	09	13	00
326A	05	90	00	03	0D	00
3270	01	A0	00	01	09	00
3276	02	B0	00	01	07	00
327C	07	C0	00	00	05	00
3282	03	D0	00	00	04	00
3288	0A	E0	00	00	04	00
328E	01	F0	00	00	01	00
3294	06	00	00	00	01	00
329A	01	10	00	00	01	00
32A0	02	20	00	00	01	00
32A6	01	30	00	00	00	00
32AC	01	40	00	00	01	00
32B2	01	50	00	00	01	FF

VOCABULARY : 9

Pattern no. 2

ADDR	Z-CR	Time	BPF3	BPF2	BPF1	FLAG
319E	12	81	00	00	00	00
31A4	07	90	01	02	00	00
31AA	05	A0	09	09	06	00
31B0	05	B0	17	1B	09	00
31B6	05	C0	29	43	09	00
31BC	04	D0	45	61	09	00
31C2	05	E0	53	77	08	00
31C8	06	F0	58	87	09	00
31CE	08	00	5B	8F	07	00
31D4	05	10	59	94	08	00
31DA	0A	20	54	96	03	00
31E0	06	30	4E	96	05	00
31E6	04	40	4B	93	05	00
31EC	06	50	46	93	05	00
31F2	05	60	3F	93	05	00
31F8	06	70	3C	98	06	00
31FE	05	80	33	98	05	00
3204	05	90	2E	99	05	00
320A	05	A0	2D	99	03	00
3210	06	B0	2C	9A	08	00
3216	04	C0	28	88	03	00
321C	04	D0	26	6D	03	00
3222	04	E0	26	56	05	00
3228	04	F0	28	61	07	00
322E	06	00	28	6C	0D	00
3234	05	10	29	55	19	00
323A	06	20	1F	3E	25	00
3240	05	30	16	36	33	00
3246	06	40	0F	2B	39	00
324C	07	50	0A	1D	29	00
3252	0B	60	06	16	21	00
3258	0F	70	04	10	18	00
325E	08	80	02	0C	13	00
3264	04	90	01	09	0F	00
326A	05	A0	01	09	09	00
3270	01	B0	00	08	07	00
3276	03	C0	00	05	03	00
327C	02	D0	00	02	02	00
3282	14	E0	00	01	01	00
3288	0C	F0	00	00	00	00
328E	03	00	00	00	00	00
3294	08	10	00	00	00	00
329A	01	30	00	01	01	00
32A0	01	40	00	00	00	FF

ภาคผนวก ง.
ข้อมูลจำเพาะของอุปกรณ์

Z80®-CPU
Z80A-CPU

Zilog®

Product Specification

MARCH 1978

The Zilog Z80 product line is a complete set of microcomputer components, development systems and support software. The Z80 microcomputer component set includes all of the circuits necessary to build high-performance microcomputer systems with virtually no other logic and a minimum number of low cost standard memory elements.

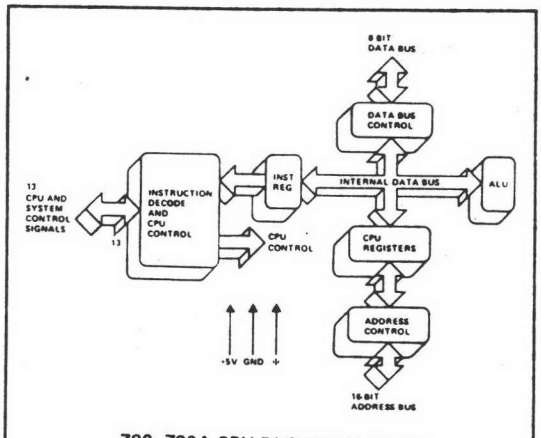
The Z80 and Z80A CPU's are third generation single chip microprocessors with unrivaled computational power. This increased computational power results in higher system through-put and more efficient memory utilization when compared to second generation microprocessors. In addition, the Z80 and Z80A CPU's are very easy to implement into a system because of their single voltage requirement plus all output signals are fully decoded and timed to control standard memory or peripheral circuits. The circuit is implemented using an N-channel, ion implanted, silicon gate MOS process.

Figure 1 is a block diagram of the CPU, Figure 2 details the internal register configuration which contains 208 bits of Read/Write memory that are accessible to the programmer. The registers include two sets of six general purpose registers that may be used individually as 8-bit registers or as 16-bit register pairs. There are also two sets of accumulator and flag registers. The programmer has access to either set of main or alternate registers through a group of exchange instructions. This alternate set allows foreground/background mode of operation or may be reserved for very fast Interrupt response. Each CPU also contains a 16-bit stack pointer which permits simple implementation of multiple level interrupts, unlimited subroutine nesting and simplification of many types of data handling.

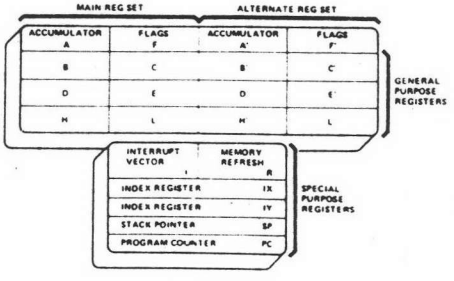
The two 16-bit index registers allow tabular data manipulation and easy implementation of relocatable code. The Refresh register provides for automatic, totally transparent refresh of external dynamic memories. The I register is used in a powerful interrupt response mode to form the upper 8 bits of a pointer to a interrupt service address table, while the interrupting device supplies the lower 8 bits of the pointer. An indirect call is then made to this service address.

FEATURES

- Single chip, N-channel Silicon Gate CPU.
- 158 instructions–includes all 78 of the 8080A instructions with total software compatibility. New instructions include 4-, 8- and 16-bit operations with more useful addressing modes such as indexed, bit and relative.
- 17 internal registers.
- Three modes of fast interrupt response plus a non-maskable interrupt.
- Directly interfaces standard speed static or dynamic memories with virtually no external logic.
- 1.0 μs instruction execution speed.
- Single 5 VDC supply and single-phase 5 volt Clock.
- Out-performs any other single chip microcomputer in 4-, 8-, or 16-bit applications.
- All pins TTL Compatible
- Built-in dynamic RAM refresh circuitry.

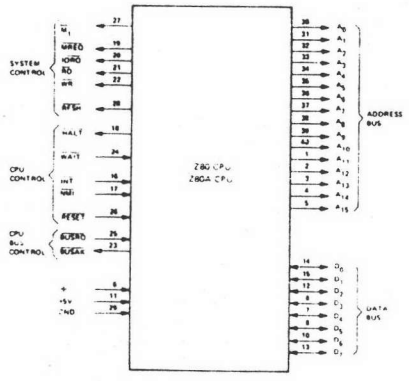


Z80, Z80A CPU BLOCK DIAGRAM



Z80, Z80A CPU REGISTERS

Z80, Z80A-CPU Pin Description



Z80, Z80A CPU PIN CONFIGURATION

A_0-A_{15} (Address Bus) — Tri-state output, active high. A_0-A_{15} constitute a 16-bit address bus. The address bus provides the address for memory (up to 64K bytes) data exchanges and for I/O device data exchanges.

D_0-D_7 (Data Bus) — Tri-state input/output, active high. D_0-D_7 constitute an 8-bit bidirectional data bus. The data bus is used for data exchanges with memory and I/O devices.

$\overline{M_1}$ (Machine Cycle one) — Output, active low. $\overline{M_1}$ indicates that the current machine cycle is the OP code fetch cycle of an instruction execution.

$\overline{MREQ}$ (Memory Request) — Tri-state output, active low. The memory request signal indicates that the address bus holds a valid address for a memory read or memory write operation.

$\overline{IORQ}$ (Input/ Output Request) — Tri-state output, active low. The $\overline{IORQ}$ signal indicates that the lower half of the address bus holds a valid I/O address for a I/O read or write operation. An $\overline{IORQ}$ signal is also generated when an interrupt is being acknowledged to indicate that an interrupt response vector can be placed on the data bus.

$\overline{RD}$ (Memory Read) — Tri-state output, active low. $\overline{RD}$ indicates that the CPU wants to read data from memory or an I/O device. The addressed I/O device or memory should use this signal to gate data onto the CPU data bus.

$\overline{WR}$ (Memory Write) — Tri-state output, active low. $\overline{WR}$ indicates that the CPU data bus holds valid data to be stored in the addressed memory or I/O device.

$\overline{RFSH}$ (Refresh) — Output, active low. $\overline{RFSH}$ indicates that the lower 7 bits of the address bus contain a refresh address for dynamic memories and the current $\overline{MREQ}$ signal should be used to do a refresh read to all dynamic memories.

$\overline{HALT}$ (Halt state) — Output, active low. $\overline{HALT}$ indicates that the CPU has executed a HALT software instruction and is awaiting either a non-maskable or a maskable interrupt (with the mask enabled) before operation can resume. While halted, the CPU executes NOP's to maintain memory refresh activity.

$\overline{WAIT}$ (Wait) — Input, active low. $\overline{WAIT}$ indicates to the Z-80 CPU that the addressed memory or I/O devices are not ready for a data transfer. The CPU continues to enter wait states for as long as this signal is active.

$\overline{INT}$ (Interrupt Request) — Input, active low. The Interrupt Request signal is generated by I/O devices. A request will be honored at the end of the current instruction if the internal software controlled interrupt enable flip-flop (IFF) is enabled.

$\overline{NMI}$ (Non Maskable Interrupt) — Input, active low. The non-maskable nterrupt request line has a higher priority than $\overline{INT}$ and is always recognized at the end of the current instruction, independent of the status of the interrupt enable flip-flop. $\overline{NMI}$ automatically forces the Z-80 CPU to restart to location 0066_H.

$\overline{RESET}$ — Input, active low. $\overline{RESET}$ initializes the CPU as follows: reset interrupt enable flip-flop, clear PC and registers I and R and set interrupt to 8080A mode. During reset time, the address and data bus go to a high impedance state and all control output signals go to the inactive state.

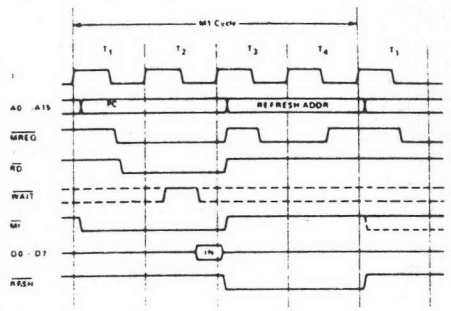
$\overline{BUSRQ}$ (Bus Request) — Input, active low. The bus request signal has a higher priority than $\overline{NMI}$ and is always recognized at the end of the current machine cycle and is used to request the CPU address bus, data bus and tri-state output control signals to go to a high impedance state so that other devices can control these busses.

$\overline{BUSAK}$ (Bus Acknowledge) — Output, active low. Bus acknowledge is used to indicate to the requesting device that the CPU address bus, data bus and tri-state control bus signals have been set to their high impedance state and the external device can now control these signals.

Timing Waveforms

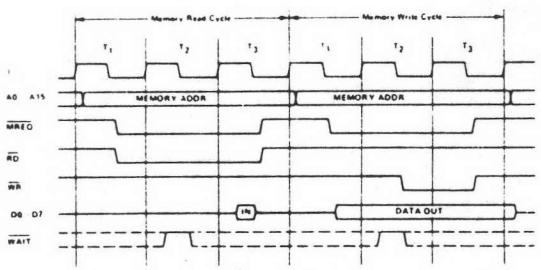
INSTRUCTION OP CODE FETCH

The program counter content (PC) is placed on the address bus immediately at the start of the cycle. One half clock time later $\overline{MREQ}$ goes active. The falling edge of $\overline{MREQ}$ can be used directly as a chip enable to dynamic memories. $\overline{RD}$ when active indicates that the memory data should be enabled onto the CPU data bus. The CPU samples data with the rising edge of the clock state T_3. Clock states T_3 and T_4 of a fetch cycle are used to refresh dynamic memories while the CPU is internally decoding and executing the instruction. The refresh control signal $\overline{RFSH}$ indicates that a refresh read of all dynamic memories should be accomplished.



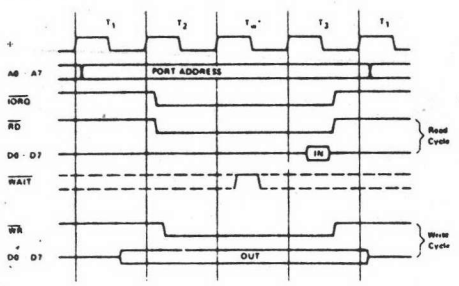
MEMORY READ OR WRITE CYCLES

Illustrated here is the timing of memory read or write cycles other than an OP code fetch (M_1 cycle). The $\overline{MREQ}$ and $\overline{RD}$ signals are used exactly as in the fetch cycle. In the case of a memory write cycle, the $\overline{MREQ}$ also becomes active when the address bus is stable so that it can be used directly as a chip enable for dynamic memories. The $\overline{WR}$ line is active when data on the data bus is stable so that it can be used directly as a R/W pulse to virtually any type of semiconductor memory.



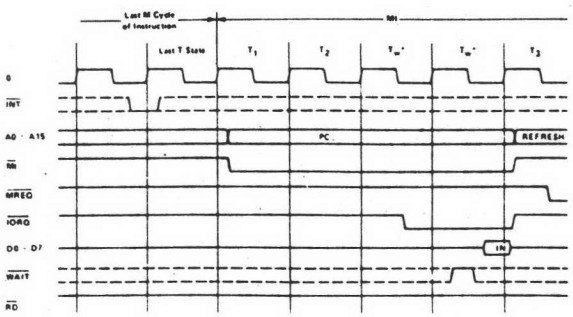
INPUT OR OUTPUT CYCLES

Illustrated here is the timing for an I/O read or I/O write operation. Notice that during I/O operations a single wait state is automatically inserted (Tw*). The reason for this is that during I/O operations this extra state allows sufficient time for an I/O port to decode its address and activate the $\overline{WAIT}$ line if a wait is required.



INTERRUPT REQUEST/ACKNOWLEDGE CYCLE

The interrupt signal is sampled by the CPU with the rising edge of the last clock at the end of any instruction. When an interrupt is accepted, a special M_1 cycle is generated. During this M_1 cycle, the $\overline{IORQ}$ signal becomes active (instead of $\overline{MREQ}$) to indicate that the interrupting device can place an 8-bit vector on the data bus. Two wait states (Tw*) are automatically added to this cycle so that a ripple priority interrupt scheme, such as the one used in the Z80 peripheral controllers, can be easily implemented.



SN54LS240/SN74LS240 • SN54LS241/SN74LS241 • SN54LS244/SN74LS244

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

DESCRIPTION—The 54LS/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

- HYSTERESIS AT INPUTS TO IMPROVE NOISE MARGINS
- 3-STATE OUTPUTS DRIVE BUS LINES OR BUFFER MEMORY ADDRESS REGISTERS
- OUTPUTS SINK 40 mA AT V_{OL} = 0.5 V
- 10 mA SOURCE CURRENT
- INPUT CLAMP DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS
- FULLY TTL AND CMOS COMPATIBLE

TRUTH TABLES

54LS/74LS240

INPUTS		OUTPUT
$\overline{E}_1, \overline{E}_2$	D	
L	L	H
L	H	L
H	X	(Z)

54LS/74LS244

INPUTS		OUTPUT
$\overline{E}_1, \overline{E}_2$	D	
L	L	L
L	H	H
H	X	(Z)
H	X	(Z)

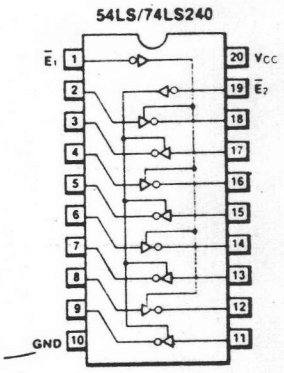
54LS/74LS241

INPUTS		OUTPUT	INPUTS		OUTPUT
$\overline{E}_1$	D		E_2	D	
L	L	L	H	L	L
L	H	H	H	H	H
H	X	(Z)	L	X	(Z)
H	X	(X)	L	X	(Z)

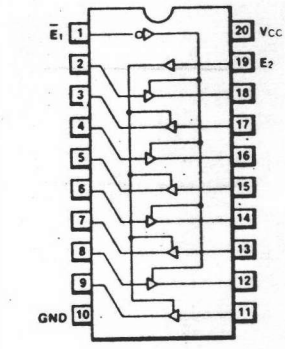
H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

LOGIC AND CONNECTION DIAGRAMS
DIP (TOP VIEW)

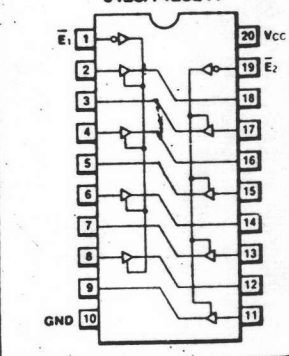
54LS/74LS240



54LS/74LS241



54LS/74LS244



Advance Information

SN54LS245/SN74LS245
OCTAL BUS TRANSCEIVER

DESCRIPTION—The 54LS/74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input ($\overline{E}$) can be used to isolate the buses.

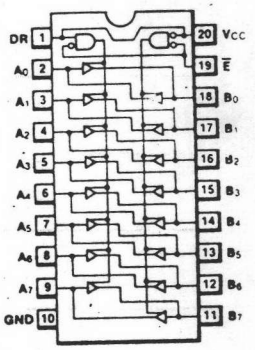
- HYSTERESIS INPUTS TO IMPROVE NOISE IMMUNITY
- 2-WAY ASYNCHRONOUS DATA BUS COMMUNICATION
- INPUT DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS
- FULLY TTL AND CMOS COMPATIBLE

TRUTH TABLE

INPUTS		OUTPUT
$\overline{E}$	DR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

LOGIC AND CONNECTION DIAGRAM
DIP (TOP VIEW)



GUARANTEED OPERATING RANGES

PART NUMBERS	SUPPLY VOLTAGE (V_{CC})			TEMPERATURE
	MIN	TYP	MAX	
SN54LS245X	4.5 V	5.0 V	5.5 V	−55°C to +125°C
SN74LS245X	4.75 V	5.0 V	5.25 V	0°C to +70°C

X = package type; W for Flatpak, J for Ceramic Dip, N for Plastic Dip. See Packaging Information Section for packages available on this product.

This is advance information and specifications are subject to change without notice.

SN54LS273/SN74LS273

8-BIT REGISTER WITH CLEAR

DESCRIPTION – The 54LS/74LS273 is a high-speed 8-Bit Register. The register consists of eight D-Type Flip-Flops with a Common Clock and an asynchronous active LOW Master Reset. This device is supplied in a 20-pin package featuring 0.3 inch lead spacing.

- **8-BIT HIGH SPEED REGISTER**
- **PARALLEL REGISTER**
- **COMMON CLOCK AND MASTER RESET**
- **INPUT CLAMP DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS**
- **FULLY TTL AND CMOS COMPATIBLE**

PIN NAMES

		LOADING (Note a) HIGH	LOW
CP	Clock (Active HIGH Going Edge) Input	0.5 U.L.	0.25 U.L.
D_0-D_7	Data Inputs	0.5 U.L.	0.25 U.L.
$\overline{MR}$	Master Reset (Active LOW) Input	0.5 U.L.	0.25 U.L.
Q_0-Q_7	Register Outputs (Note b)	10 U.L.	5 (2.5) U.L.

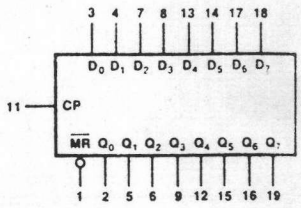
NOTES:
a. 1 TTL Unit Load (U.L.) = 40μA HIGH/1.6 mA LOW
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

TRUTH TABLE

$\overline{MR}$	CP	D_x	Q_x
L	X	X	L
H	⎍	H	H
H	⎍	L	L

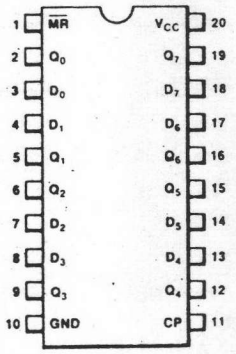
H = High Logic Level
L = Low Logic Level
X = Immaterial

LOGIC SYMBOL

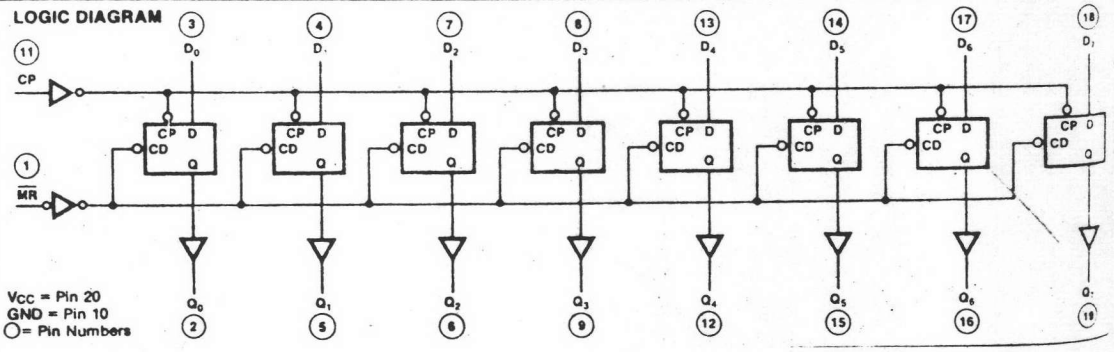


V_{CC} = Pin 20
GND = Pin 10

CONNECTION DIAGRAM DIP (TOP VIEW)



LOGIC DIAGRAM



VCC = Pin 20
GND = Pin 10
○= Pin Numbers

ประวัติผู้เขียน

นายสุพงศ์ เภกะนันทน์ เกิดวันที่ 26 มีนาคม 2502 สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (ไฟฟ้า) จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2522 เข้าศึกษาระดับปริญญามหาบัณฑิต สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ ในปี พ.ศ. 2524