EEE/INSTR/ECE/CS F241- MICROPROCESSOR PROGRAMMING AND INTERFACING

DESIGN - Voice Digitizer

Group no: 81
Group members:

1. ABHIVANDAN GANDHI (2018A8PS0831G)
2. AMIT VIGNESHWAR HEGDE (2018AAPS0421G)
3. AMIT SANJAY KSHIRSAGAR (2018AAPS0381G)
4. JHA SHIVANK SUDHIR (2018AAPS0298G)
5. NIKHIL RAVI (2018AAPS1032G)

Date:19/04/2020

Contents

Problem Statement

Voice signal is to be digitized and reproduced with certain modification by the microprocessor. Output from a microphone is sampled and digitized using an 8- bit ADC at the rate of 1000 samples per second. The output obtained from the microphone has been pre-processed inside the microphone to provide a signal varying in amplitude between 0 – 5 V. The digitized signal is to be stored in RAM. The signal for a period of 6 seconds has to be digitized. The voice stored has to be reproduced with delay when user closes a switch labelled sound replay. The delay to be entered by the user are numbers from 1-9, with the help of a key-pad. The keypad has digits 0-9, backspace and enter. A seven-segment display has to be provided with the keypad to display the delay value entered by the user. The delay is between samples – If value entered in is 5 then delay between two adjacent samples when reproduced is 5ms.

Assumptions

Here we are using the start, replay and the replay switch with a different delay only at specific times. Hence, we are using polling to check the switches instead of interrupts so that we can check the switch whenever it is required.

Specifications

1) Here we give the Analog input to the system through the audio generator which generates analog signals in the range 0 to 5 volts. This is given as input to ADC which converts analog input signal into digital at the rate of 1000 samples per second for 6 seconds. The rate is 1KHz, hence we take the input samples at a time difference of 1ms.
2) The sources of interrupts for 8259 are: 1 interrupt from the timer(1ms delay generated for taking the input samples) and 1 from the EOC of ADC. The ROM is used to store IVT of 8259. The vector address used is 80H onwards.
3) Once the timing interrupt is raised , the ISR is invoked and ALE and SOC are made high.
4) When the conversion is completed the EOC becomes high and an interrupt is raised, ISR is invoked OE and is made high. The data created after conversion is stored into the RAM.
5) The same process goes on for storing 6000 samples.
6) To indicate delay time between 2 consecutive output samples, a keypad is used. The user has to input a digit (1-9) and then press the enter button. The number that the user has pressed indicates the delay time between the 2 output samples. The keyboard is 4X3 matrix with buttons representing digits 0-9, backspace and enter.
7) When the user presses replay switch the sound is reproduced with the delay that the user has input.
8) The count of counter 2 of 8254 is initialized on the basis of the digit that the user has input. The ISR is invoked and digital values stored are given as input to DAC. The DAC converts the digital signal to analog. Then the sound is output through the speaker.

Components Used

Chip No	Chip name/ purpose	Quantity
8086	Micro Processor	1
74LS373	Octal Latch	3
74LS245	8 bit Bidirectional Buffer	2
2716	ROM	4
6116	RAM	4
8255	Programmable Peripheral Interface	2
ADC 0808	Analog to Digital Converter	1
DAC 0808	Digital to Analog Converter	1
8254	Programmable Interval Timer	1
8284	5Mhz Clock	1
8259	Programmable Interrupt Controller	1
7447	BCD to 7-Segment	1
74LS138	3:8 Decoder	1
74LS244	Unidirectional Buffer	1
74HC4075	3 Input Or Gate	12
7432	2 Input Or Gate	8
7404	Not gate	6

Other components

1. Speaker
2. Microphone/ or A audio generator
3. LEDs
4. 7 segment display
5. Resistors
6. Spdt switches
7. Operational amplifier
8. 4x3 keypad

Memory interfacing

Here we are using 8KB of RAM and ROM each, because we have to take 6000B input which is approximately 6KB and also since it is 8086 we use odd and even banks of memory. Both of them are made by using 4- 2KB chips.

RAM 1 03000H - 03FFFH

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

RAM 2 04000H - 04FFFH

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1

ROM 1 00000H - 00FFFH

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

ROM 2 FF000H - FFFFFH

19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

I/O Interfacing

Here we have used two 8255 programmable peripheral interface controllers for connecting the input/output devices.

We connect the A0 and A1 of both 8255A and 8255B to the A1 and A2 of the 8086-address bus respectively.

Port Address Maps

8255A 00H - 08H

		A7	A6	A5	A4	A3	A2	A1	A0
PORT A	00h	0	0	0	0	0	0	0	0
PORT B	02h	0	0	0	0	0	0	1	0
PORT C	04h	0	0	0	0	0	1	0	0
CONTROL REGISTER	06h	0	0	0	0	0	1	1	0

8255B 10H – 18H

		A7	A6	A5	A4	A3	A2	A1	A0
PORT A	10h	0	0	0	1	0	0	0	0
PORT B	12h	0	0	0	1	0	0	1	0
PORT C	14h	0	0	0	1	0	1	0	0
CONTROL REGISTER	16h	0	0	0	1	0	1	1	0

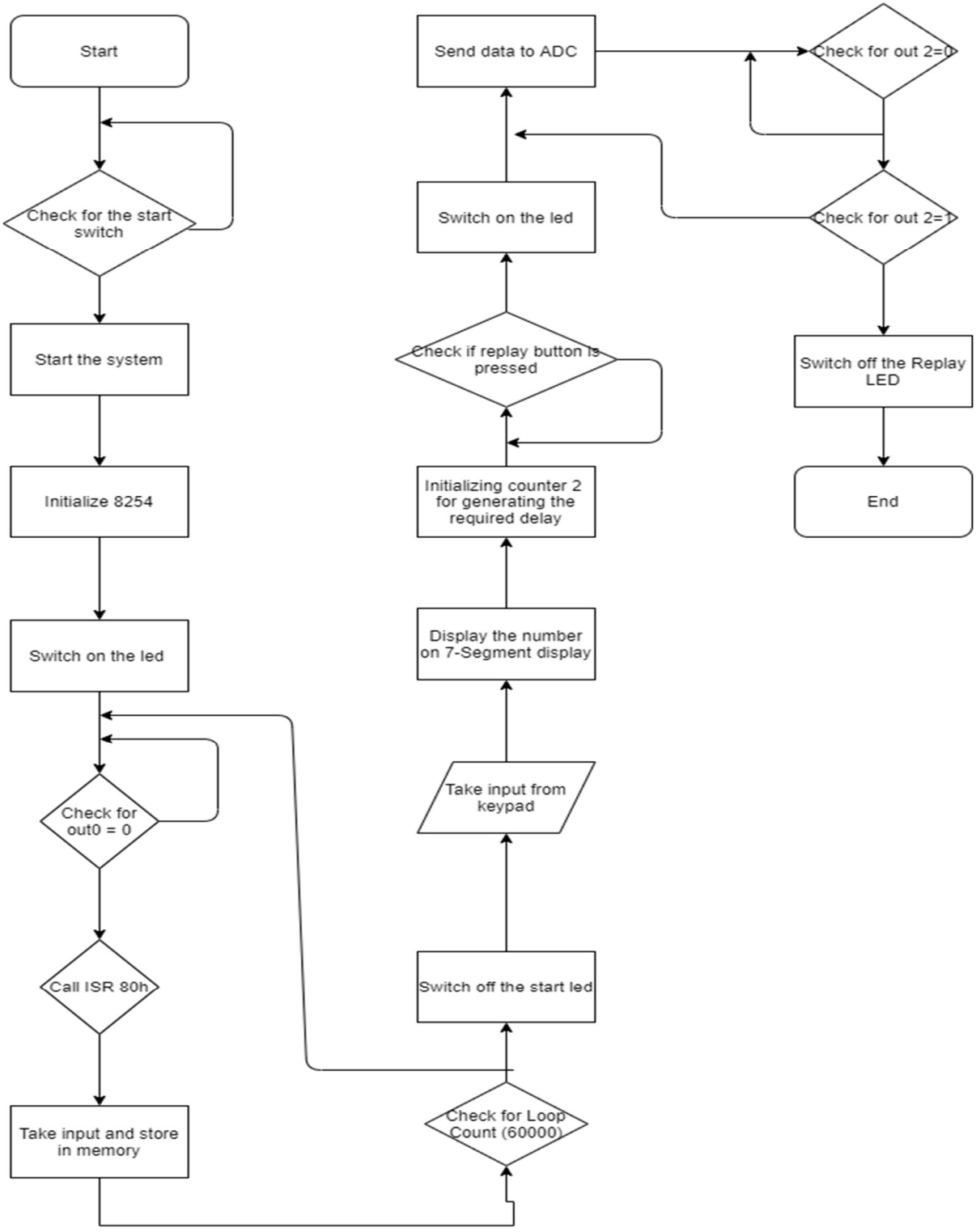
8253/4 20H – 28H

		A7	A6	A5	A4	A3	A2	A1	A0
COUNTER 0	20h	0	0	1	0	0	0	0	0
COUNTER 1	22h	0	0	1	0	0	0	1	0
COUNTER 2	24h	0	0	1	0	0	1	0	0
CONTROL REGISTER	26h	0	0	1	0	0	1	1	0

8259 B0H-B2H

1st Address - B0H
2nd Address- B2H

FLOWCHART


Start
Check for the start switch
Start the system
Initialize 8254
Switch on the led
Check for out0 = 0
Call ISR 80h
Take input and store in memory
Check for Loop Count (60000)
Switch off the start led
Take input from keypad
Display the number on 7-Segment display
Initializing counter 2 for generating the required delay
Check if replay button is pressed
Switch on the led
Send data to ADC
Check for out 2=0
Check for out 2=1
Switch off the Replay LED
End

Screenshot of code/firmware

Implemented using emu8086 attached.

```
#make_bin#

#LOAD_SEGMENT=FFFFh#
#LOAD_OFFSET=0000h#

#CS=0000h#
#IP=0000h#

#DS=0000h#
#ES=0000h#

#SS=0000h#
#SP=FFFEh#

#AX=0000h#
#BX=0000h#
#CX=0000h#
#DX=0000h#
#SI=0000h#
#DI=0000h#
#BP=0000h#

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;COMPONENTS USED AND THIER START ADDRESS
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;8255A  START ADDRESS IS 00H
;PORT A PA0 IS CONNECTED TO START SWITCH
;PORT A PA1 IS CONNECTED TO REPLAY SWITCH
;PORT A PA2 IS CONNECTED TO REPLAY SWITCH WITH DIFFERENT DELAY
;PORT A PA3 IS CONNECTED TO OUT FROM INTERVAL TIMER
;PORT B IS ADC
;PORT C PC0 IS CONNECTED TO START LED
;PORT C PC1 IS CONNECTED TO REPLAY LED
;PORT C PC2 IS CONNECTED TO GATE OF COUNTER 2
;PORT C PC4 IS CONNECTED TO SOC OF ADC
;PORT C PC5 IS CONNECTED TO OE OF ADC
;PORT C PC6 IS CONNECTED TO ALE OF ADC

;8255B  START ADDRESS IS 10H
;PORT A IS IS CONNECTED TO DAC
;PORT B PB4 TO PB7 IS CONNECTED TO  7SEG LED
;PORT CL IS OUTPUT -COLUMN I/PS
;PORT CU IS INPUT  -ROW O/PS

```

```
;8254   START ADDRESS IS 20H
;COUNTER0 FOR GENERATING THE CLOCK OF ADC
;COUNTER1 FOR GENERATING 2KHz
;COUNTER2 FOR 1MS DELAY GENERATION

;8259   START ADDRESS IS B0H
;HANDLING THE INTERRUPTS

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; DESCIPTION OF COMPONENTS FINISHED
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;       IVT ENTRY AT 80H i.e 128*4=512
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

jmp st1                         ;3BYTE
db  509 dup(0)                  ;509 BYTE
dw i_timer                      ;2BYTE
dw 0000                         ;2BYTE
dw i_eoc                        ;2BYTE
db 506 dup(0)                   ;506 BYTE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;   END OF IVT AS 1KB IS DONE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INDEX dw ?                      ; VARIABLE FOR STORING KEY PRESSED

;KEY PRESSED TABLE
TABLE_K     db      7DH,0EEH,0EDH,0EBH,0DEH,0DDH    ;0,1,2,3,4,5
            db      0DBH,0BEH,0BDH,0BBH             ;6,7,8,9
            db      07EH,07BH                       ;BACKSPACE AND ENTER

;DISPLAY TABLE
TABLE_D     db      00H,10H,20H,30H,40H,50H
            db      60H,70H,80H,90H
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                       MAIN CODE
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

st1:        cli
;INITIALIZE DS,ES,SS TO START OF RAM
mov ax,3000h            ;START OF THE RAM IS ENTERED HERE
mov ds,ax
```

```
mov es,ax
mov ss,ax
mov sp,0FFFEH
mov si,0000

;INITIALIZING 8255A
mov al,10010010b            ;8255A A-INPUT, B-INPUT,C-OUTPUT
out 06h,al                  ;CR OF 8255A
mov al,00000000b
out 04h,al                  ;OUTPUT OF PORT C IS 00

;CHECKING THE START SWITCH
start_switch:   mov al,10010010b        ;8255A A-INPUT B-INPUT C-OUTPUT
                out 06h,al              ;CR OF 8255A
                in al,00h               ;INPUT FROM PORT A(START SWITCH)
                and al,00000001b        ;MASKING BITS OTHER THAN PA0
                cmp al,00000001b        ;PA0=1 IF SWITCH IS PRESSED
                jnz start_switch        ;CHECK UNTIL START SWITCH IS PRESSED

;INITIALIZING 8254 TO GENERATE CLOCK FOR ADC
mov al,00010100b            ;8254 COUNTER-0 MODE 2-CLOCK FOR ADC
out 26h,al                  ;CR OF 8254
mov al,05h                  ;COUNT=5 FOR OUT =1MHz AND INPUT CLOCK 5MHz FROM 8284
out 20h,al                  ;COUNT STORED IN COUNTER 0

mov al,01110100b            ;8254 COUNTER-1 MODE 2, 16 BIT COUNT WRITTEN
out 26h,al                  ;CR OF 8254
mov al,0F4h                 ;COUNT=01F4H=500 for 2KHz SOUND DELAY, INPUT CLOCK IS 1MHz
out 22h,al                  ;LSB COUNT IN COUNTER1
mov al,01h
out 22h,al                  ;MSB COUNT IN COUNTER1

;INITIALIZING 8255A
mov al,10010010b            ;8255A A-INPUT, B-INPUT,C-OUTPUT
out 06h,al                  ;CR OF 8255A

mov si,00h                  ;DS:[SI] WILL POINT TO THE START LOC OF RAM WHERE INPUT DATA WILL BE STORED
mov cx,6000                 ;COUNT FOR 6000 SAMPLES

;INITIALIZING 8254
mov al,10010100b            ;8254 COUNTER-2 MODE-2,8BIT IN LSB,DELAY GENERATION COUNTER
out 26h,al                  ;CR OF 8254
mov al,02h                  ;COUNT=2 FOR 1ms, AND INPUT CLOCK IS 2KHz
out 24h,al                  ;STORING COUNT IN COUNTER 2
```

```
132
133 mov al,00000001b             ;THIS SETS PC0 TO 1 WHICH IS CONNECTED TO THE START LED
134 out 06h,al                   ;CR OF 8255A
135
136 mov al,00000101b             ;THIS SETS PC2 OF 8255A to 1 WHICH IS CONNECTED TO GATE OF COUNTER 2
137                               ;THIS COUNTER GENERATES 1ms DELAY WHICH IS REQUIRED FOR TAKING THE INPUT FROM ADC
138 out 06h,al
139
140 inputdata:
141 ;INITIALIZE 8259
142
143 mov al,00010011b             ;ICW1-D0-X86 AND ABOVE, D1-SINGLE 8259, D3-EDGE TRIGGERED, D4-ALWAYS 1
144 out 30h,al                   ;STORED IN STARTING ADDRESS
145 mov al,10000000b             ;ICW2- IVT START ADDRESS 80H MASK LOWER 3 BITS
146 out 32h,al                   ;STORED IN SECOND ADDRESS
147 mov al,00000001b             ;ICW4
148 out 32h,al                   ;STORED IN SECOND ADDRESS
149 mov al,11111100b             ;OCW1-IR0 IS ENABLED OTHERS ARE MASKED
150 out 32h,al                   ;STORED IN SECOND ADDRESS
151 sti
152
153 nop
154 nop
155 nop
156 nop
157
158 dec cx
159 jnz inputdata
160
161 mov al,00000000b             ;THIS SETS PC0 TO 0 WHICH IS CONNECTED TO THE START LED i.e DATA READING IS DONE
162 out 06h,al                   ;CR OF 8255A
163
164 mov al,00000100b             ;THIS SETS PC2 OF 8255A to 0 WHICH IS CONNECTED TO GATE OF COUNTER 2, TO DISABLE THE GATE
165 out 06h,al                   ;CR OF 8255A
166
167 ;CHECKING WHICH BUTTON IS PRESSED ON THE KEYPAD
168 ;INITIALIZING 8255B
169 mov al,10001000b             ;8255B A-OUTPUT, B-OUTPUT CL-OUTPUT CU-INPUT
170 out 16h,al                   ;CR OF 8255B
171 mov al,00000000b             ;00000000 IS DIPLAYED ON PORT B i.e 0 IS DISPLAYED ON THE 7SEG LED
172 out 12h,al                   ;SENT TO PORT B
173
174 x0: mov al,00h               ;00000000 IS SENT TO PORT C
175     out 14h,al               ;PORT C
```

```
176
177 x1: in al,14h                ;INPUT FROM PORT CU
178     and al,11110000b        ;MASK LOWER BIT
179     cmp al,11110000b        ;CHECK IF ALL INPUTS ARE HIGH
180     jnz x1                  ;CHECK UNTIL ALL INPUTS ARE HIGH i.e ALL THE KEYS ARE RELEASED/ NONE OF THEM IS PRESSED
181     call delay20ms          ;DELAY OF 20 MS
182     mov al,00h              ;00000000 IS DISPLAYED ON PORT C
183     out 14h,al              ;PORT C
184
185 x2: in al,14h                ;INPUT FROM PORT CU
186     and al,0f0h
187     cmp al,0f0h
188     jz x2                   ;CHECK IF ANY KEY IS PRESSED OR NOT
189     call delay20ms          ;CREATE A DELAY
190     mov al,00h              ;00 IS DIPLAYED ON PORT C
191     out 14h,al
192     in al,14h               ;THEN AGAIN CHECK IF THE KEY PRESSED IS A VALID KEY PRESSED OR NOT
193     and al,0f0h
194     cmp al, 0f0h
195     jz x2
196
197 mov al,00001110b            ;CHECK FOR KEY PRESSED IN COLUMN 1
198 mov bl,al
199 out 14h,al                  ;STORED IN PORT C
200 in al,14h                   ;INPUT FROM PORT C
201 and al,0f0h                 ;MASK LOWER BIT
202 cmp al,0f0h                 ;CHECK IF ALL INPUTS ARE HIGH
203 jnz x3
204
205 mov al,00001101b            ;CHECK FOR KEY PRESSED IN COLUMN 2
206 mov bl,al
207 out 14h,al                  ;STORED IN PORT C
208 in al,14h                   ;INPUT FROM PORT C
209 and al,0f0h                 ;MASK LOWER BIT
210 cmp al,0f0h                 ;CHECK IF ALL INPUTS ARE HIGH
211 jnz x3
212
213 mov al,00001011b            ;CHECK FOR KEY PRESSED IN COLUMN 3
214 mov bl,al
215 out 14h,al                  ;STORED IN PORT C
216 in al,14h                   ;INPUT FROM PORT C
217 and al,0f0h                 ;MASK LOWER BIT
218 cmp al,0f0h                 ;CHECK IF ALL INPUTS ARE HIGH
219 jnz x3
```

```
220
221 x3: or al,bl                 ;LOWER BIT IS STORED IN AL AND UPPER BITS IS 1111
222     cmp al,7bh              ;COMPARING WITH ENTER
223     jz x_enter
224     cmp al,7eh              ;COMPARING WITH BACKSPACE
225     jz backspace
226
227 x_check:     mov cx,09h      ;MOVE COUNT 9 IN CX
228             mov di,00h      ;INITIALIZE DI TO 0
229
230 x_keypressed:    cmp al,cs:TABLE_K[di]   ;COMPARE INPUT WITH TABLE_K
231                 mov INDEX,di            ;STORE DATA IN INDEX
232                 jz x_display            ;IF KEY IS MATCHED THEN GET THE VALUE FROM TABLE_D
233                 inc di
234                 loop x_keypressed
235
236 x_display:      lea bx,TABLE_D          ;INITIALIZE BX TO START ADDRESS OF TABLE_D
237                 mov al,cs:[bx+di]       ;GET THE VALUE FOR INDEX
238                 out 12h,al              ;DISPLAY ON PORT B 7SEG LED
239                 jmp x0
240
241 ;D20 MS FUNCTION
242 delay20ms:  mov cx,2220                ;DELAY GENERATED IS 0.02SEC i.e 2 ms
243     xn:     loop xn
244             ret
245 cli
246
247 backspace:  mov al,7dh                 ;DISPLAYING 0 AGAIN ON 7SEG LED ;BACKSPACE
248             jmp x_check
249
250 x_enter:                   ;IF ENTER IS PRESSED
251
252 in al,00h                  ;TAKE INPUT FROM PORT A OF 8255A
253 and al,00000010b           ;REPLAY SWITCH
254 cmp al,00000010b
255 jnz x_enter                ;CHECK IF PA1=1 IF ITS NOT THEN JUMP TO x_enter
256
257 mov si,00h                 ;BX HAS THE STARTING ADDRESS OF THE MEMORY LOCATION WHERE THE RECORDING IS STORED
258 mov cx,6000
259
260 mov al,10010100b           ;COUNTER 2 IS SELECTED,8BIT IN LSB, MODE 2, BINARY
261 out 26h,al                 ;CR FOR 8254
262 mov ax,INDEX               ;COUNT=INDEX
263 mov dl,02h
```

```
mul dl
out 24h,al                  ;STORING COUNT=INDEX IN COUNTER 2
                            ;i.e PRODUCING A DELAY OF THE NUMBER WHICH IS PRESSED

mov al,00000011b            ;THIS SETS PC1 of 8255A to 1. THIS IS ATTCHED TO REPLAY LED
out 06h,al                  ;CR OF 8255A SET PC1=0

mov al,00000101b            ;THIS SETS PC2 OF 8255A to 1 WHICH IS CONNECTED TO GATE OF COUNTER 2, TO ENABLE THE GATE
out 06h,al                  ;CR OF 8255A SET PC2=1

output:
delay1: in al,00h           ;TAKE INPUT FROM PORT A
        and al,00001000b    ;CHECK UNTIL 4TH INPUT IS HIGH i.e 1
        cmp al,00001000b    ;4th INPUT IS THE OUT FORM 8254
        jnz delay1


mov al,ds:[si]
out 10h,al                  ;SEND THE SIGNAL TO DAC
inc si
dec cx
jnz output                  ;COMPLETE  THIS UNTIL ALL THE 6000 SAMPLES ARE OUTPUTED THROUGH THE SPEAKER

mov al,00000000b            ;SENDING 0 AS THE LAST OUTPUT TO DAC
out 10h,al

mov al,00000010b            ;THIS SETS PC1 of 8255A to 0. THIS IS ATTCHED TO REPLAY LED
out 06h,al                  ;CR OF 8255A

mov al,00000100b            ;THIS SETS PC2 OF 8255A to 0 WHICH IS CONNECTED TO GATE OF COUNTER 2, TO DISABLE THE GATE
out 06h,al                  ;CR OF 8255A

in al,00h                   ;TAKE INPUT FROM PORT A OF 8255B
and al,00000100b            ;CHECK FOR REPLAY SWITCH WITH DIFFERENT DELAY
cmp al,00000100b
jnz x0                      ;CHECK IF PB2=1 IF ITS NOT THEN JUMP TO X0

;START OF INTERRUPT
i_timer:
or al,01000000b             ;MAKING ALE=1
out 04h,al

or al,00010000b             ;MAKING SOC=1
out 04h,al
```

```
281 mov al,ds:[si]
282 out 10h,al                  ;SEND THE SIGNAL TO DAC
283 inc si
284 dec cx
285 jnz output                  ;COMPLETE  THIS UNTIL ALL THE 6000 SAMPLES ARE OUTPUTED THROUGH THE SPEAKER
286
287 mov al,00000000b            ;SENDING 0 AS THE LAST OUTPUT TO DAC
288 out 10h,al
289
290 mov al,00000010b            ;THIS SETS PC1 of 8255A to 0. THIS IS ATTCHED TO REPLAY LED
291 out 06h,al                  ;CR OF 8255A
292
293 mov al,00000100b            ;THIS SETS PC2 OF 8255A to 0 WHICH IS CONNECTED TO GATE OF COUNTER 2, TO DISABLE THE GATE
294 out 06h,al                  ;CR OF 8255A
295
296 in al,00h                   ;TAKE INPUT FROM PORT A OF 8255B
297 and al,00000100b            ;CHECK FOR REPLAY SWITCH WITH DIFFERENT DELAY
298 cmp al,00000100b
299 jnz x0                      ;CHECK IF PB2=1 IF ITS NOT THEN JUMP TO X0
300
301 ;START OF INTERRUPT
302 i_timer:
303 or al,01000000b             ;MAKING ALE=1
304 out 04h,al
305
306 or al,00010000b             ;MAKING SOC=1
307 out 04h,al
308 mov al,01100000b            ;OCW2 EOI FOR EOC
309 out 30h,al
310 iret
311
312 i_eoc:
313 or al,00100000b             ;MAKING OE=1
314 out 04h,al
315 in al,02h                   ;TAKE INPUT FROM PORT B
316 mov ds:[si],al
317 inc si
318 mov al,01100001b            ;OCW2 EOI FOR EOC
319 out 30h,al
320 and al,11101111b            ;MAKING SOC =0
321 out 04h,al
322 and al,10111111b            ;MAKING ALE =0
323 out 04h,al
324 iret
```

Variations in Proteus Implementation with Justification

1. EOC is used as NMI and the Timer Int replaced by polling.
2. Using 8253 – as 8254 not available in Proteus.
3. Clock is at 2 MHz as the clock generated for 8086 requires a long rise and fall time of clock. So, ADC clock will be only 500KHz.
4. 2732 is used as 2716 – not available in Proteus.
5. Using a gate-based circuit for memory – does the same as LS 138 here
6. 8259 not there – justification is as per point 1.

List of Attachments

1. Complete Hardware Real World Design – HARDWARE DESIGN.pdf
2. Manuals
    a. ADC 0808
    b. DAC 0808
    c. 8255A
    d. 8254
    e. 8259
3. Proteus File – PROJECT.dsn
4. EMU8086 ASM File – PROJECT.asm
5. Binary File after assembly – PROJECT.bin