An-Najah National University
Faculty of Engineering and IT

جامعة النجاح الوطنية
كلية الهندسة وتكنولوجيا المعلومات

Computer Engineering Department
Course Name: Microprocessor Lab Number: 10636392
Lab Report Grading Sheet

Instructor: **Dr. Manar Qamhieh**	Experiment #: 3
Academic Year: **2021/2022**	Experiment Name: Musical Electrical Organ
Semester: **First semester**	

Students	
1- Fathi Fathallah Ali.	
Performed on: 23/9/2021	Submitted on: 30/9/2021

Report's Outcomes

ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %

Evaluation Criterion	Grade	Points
Abstract answers of the questions: "What did you do? How did you do it? What did you find?"	0.5	
Introduction and Theory Sufficient, clear and complete statement of objectives. In addition to Presents sufficiently the theoretical basis.	1.5	
Apparatus/ Procedure Apparatus sufficiently described to enable another experimenter to identify the equipment needed to conduct the experiment. Procedure sufficiently described.	2	
Experimental Results and Discussion (In-Lab Worksheet) Crisp explanation of experimental results. Comparison of theoretical predictions to experimental results, including discussion of accuracy and error analysis in some cases.	4	
Conclusions and Recommendations Conclusions summarize the major findings from the experimental results with adequate specificity. Recommendations appropriate in light of conclusions. Correct grammar.	1	
Appearance Title page is complete, page numbers applied, content is well organized, correct spelling, fonts are consistent, good visual appeal.	1	
Total	10	

Abstract:

As the name of the experiment implies, we will create a musical electrical organ (piano), which is a real-life example of one of the uses of the microprocessor in general. We were able to do so by combining both of the ICs we learned about previously, the 8255 IC and the 8253 IC, and as a result, we will be more familiar with the usage of both ICs, as well as see what we can accomplish by combining only the two simple ICs we learnt and making them work together to get our desired outcome, the Electrical Organ.

Introduction:

In this experiment, we will learn how to utilize the 8253 timer IC and how to use it with the microprocessor to produce music, which is one of the IC's uses and applications, and we are going to be able to do that by using the push buttons we have on the kit, the 8255 IC and 8253 IC combined and connected together to get our musical organ and produce music, and the whole idea behind that is pretty simple, we just put our wanted frequency based on what button is pressed into the 8253 timer IC, as a result this IC as we learn before gives as a signal as an output, and the output signal is connected to a speaker that generate an audio based on the signal and the frequency we have, and that's basically it, we have combined the 8253 timer with the 8255 I/O interfacing IC to generate our music.

Theory:

This experiment has a theoretical component, which is the assembly code that we must create in the dice8088, debug, and transfer to the MML8086K3 kit, and that theoretical component includes certain I/O Instructions, which in our instance are:

1. IN and OUT Instructions, and basically, they transfer data between the processor accumulator register (AL, AX, EAX) and an I/O device for which we know the address already.
2. CMP instruction, which allow us to compare values in different registers together.
3. JMP instruction, which involves a transfer of control to the address of an instruction that does not follow the currently executing instruction (Unconditional jump).
4. JE instruction, which involves a transfer of control to the address of an instruction that follow the currently executing instruction, and based on the result if it's equal then it will jump (Conditional jump).

Apparatus:

1. MML8086K3 Microprocessor Trainer Kit.
2. Computer with MML8086K3 software program which is dice8088.
3. Virtual machine (VM), if the computer we have has a newer version of Windows.

Procedure:

Before we get into the practical work, we need to understand the general circuit and how does it supposed to work, and we will be able to do that by comprehending the duty of each IC and understanding the flow of the data and the whole concept behind our circuit, with that being said the data will come from our push buttons (S1-S7) to Port A at the 8255 IC as an input, then we need to determine our wanted frequency based on which push button is pressed and put it to counter 1 in the 8253 IC, after that we need to enable the 8253 to generate our signal with the wanted frequency and we are going to do that by connecting Port B (Pb0) as an output to at the 8255 IC to gate-1 at 8253 timer IC and send value of 1 to it , then the signal will be generated and it will be connected the speaker that will generate the audio we want, and to be able to do that we are going to program our 8253 and 8255 IC by configure our Control Word register based on Figure 1 and 2 below:

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control:

SC-Select Counter:

SC1	SC0	
0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal

M-Mode

0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

RL-Read/Load:

RL1	RL0	
0	0	Counter latching operation (see read/write procedure section)
1	0	Read/Load most significant byte only
0	1	Read/Load Least significant byte only
1	1	Read/Load Least significant byte first, then most significant byte

BCD:

0	Binary counter 16-bits
1	Binary coded decimal (BCD) counter (4 decades)

Figure 1: 8253 Control Word Format. / 10636392 Manual EXP #3.

8255A Control Word Format:

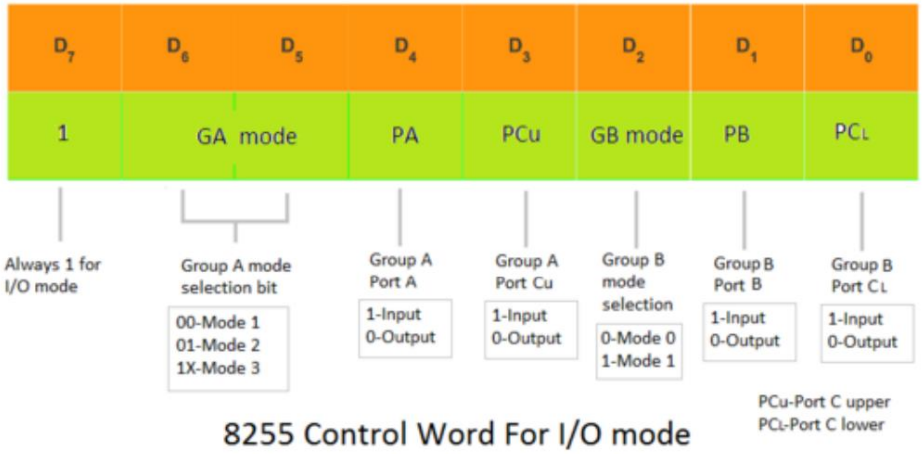

Figure 2: 8255 Control Word Format. / 10636392 Manual EXP #2.

Now we need to the required specification for the 8253 and the 8255 ICs, in 8253 timer IC we need to run counter-1, counting binary, mode 3 and we are going to use the whole 16-bit in the counter, so our configuration value for the Control word register is 01110110 which equals to 76 in hex, while in 8255 IC we need Port A only as input , GA mode 1 , GB mode 0 and Port B as an output, also we are not going to use Port C but we are going to but it as an output, as a result our configuration value for the control word register is 10010000 which equals to 90 in hex.

After calculating the configuration registers for each IC, we are going to send the configuration value to the wanted addresses based on Figure 3 and 4 below:

Selects	Address
Counter 0	0040H
Counter 1	0041H
Counter 2	0042H
Control word register	0043H

Figure 3: 8253 Counters and Config Register Addresses at MML8086K3 / 10636392 Manual EXP #3.

Port Address

Register	Address
Control word register	0FF2BH
PORT A	0FF28H
PORT B	0FF29H
PORT C	0FF2AH

Figure 4: The addresses of the 8255A's ports and configuration Register. / 10636392 Manual EXP #2.

After realizing all those valuable information which will help us do our assembly experimental program correctly, we will do the program on several steps:

1. Follow the assembly code template as shown in Figure 5 with changing the origin to 2000H, because this assembly code will be bigger than the previous codes.

```
CODE SEGMENT
ASSUME CS:CODE
ORG 2000H

START:

END START
CODE ENDS
```

Figure 5: the assembly template code we should follow. / 10636392 Manual EXP #2.

2. Send our computed configuration values to both configuration registers for the 8253 IC and the 8255 IC as Figure 6 shows.

```
MOV AL , 76H    ; Config register value => 76H in AL
MOV DX , 0043H  ; 8253 Control Word Register address stored in DX
OUT DX , AL     ; Transfer 76H to the 8253 Config Regester

MOV AL , 90H    ; Config register value => 90H in AL
MOV DX , 0FF2BH; 8255 Control Word Register address stored in DX
OUT DX , AL     ; Transfer 90H to the 8255 Config Regester
```

Figure 6: 76H to 8253 Config Reg & 90H to 8255 Config Reg.

3. Now we need to do an infinite loop by do the LOOP1 label that keeps disabling the 8253 IC, which means no signal will be generated until we enable it, and after that we read the data at Port A which is connected to the push buttons, not to mention that the push buttons work on active-low concept, so that means if the data equals 11111111 that means no button is pressed, then we jump again to label LOOP1, as Figure 7 shows.

```
LOOP1:
MOV AL , 0H     ; 0 as an output at Port B0 to disable the 8253 timer
MOV DX , 0FF29H; Port B Address at 8255 IC stored in DX
OUT DX , AL     ; Transfer 0 to Port B which is connected the GATE1 at 8253
MAIN:
MOV DX , 0FF28H; Port A address at 8255 IC stored in DX
IN AL , DX      ; Transfer the data at port-A which is connected to push buttons
CMP AL , 0FFH   ; compare it to 11111111 , that means nothing is pressed
JE LOOP1        ; jmp again to LOOP1 if nothing is pressed
```

Figure 7: Disable the 8253 timer and check for the push buttons.

4. Now, what if it the data at port A wasn't 0FFH, that means that there is some push button is 0, which means some button is pressed, and in that case what we need to do is determining what button is presses by comparing the value at port A we stored in AL register before, and jump on the specific label we want as Figure 8 shows.

```
CMP AL , 07FH ; button 1 is pressed (0111 1111)
JE S1
CMP AL , 0BFH ; button 2 is pressed (1011 1111)
JE S2
CMP AL , 0DFH ; button 3 is pressed (1101 1111)
JE S3
CMP AL , 0EFH ; button 4 is pressed (1110 1111)
JE S4
CMP AL , 0F7H ; button 5 is pressed (1111 0111)
JE S5
CMP AL , 0FBH ; button 6 is pressed (1111 1011)
JE S6
CMP AL , 0FDH ; button 7 is pressed (1111 1101)
JE S7
JMP LOOP1
```

Figure 8: specify the pressed push button.

5. Each push button has a required wanted frequency as shown in table 1, so what we need to do is calculate our wanted frequency by dividing each frequency at 1MHz, and then transfer the computed number to the 8253 timer IC.

Push Button	1	2	3	4	5	6	7
Frequency (Hz)	**440**	**493.88**	**554.37**	**587.33**	**659.26**	**739.99**	**830.61**

Table 1: The required frequency for each push button.

6. After we calculate the input we want for each button to get the required frequency, now at each label we need to enable the 8253 IC by Transferring 1 to Port B which is connected to GATE1 at 8253 which turn the IC on, also we need to transfer our calculated input to the 8253 IC to get the required frequency as Figure 9 shows, but to do that we need to transfer it twice, one for the least significant part and another one for the most significant part, because the data bus at the microprocessor we have at the kit equals to 8 bit, that means we can't transfer anything more than 8 bit at once. Let's take the first button as an example, 1MHz/440 = 2273 = 08E1 in hex, so we transfer the E1H then 08H to the counter-1 address at 8253 which is 0041H based on Figure 3, after that we jump again the MAIN label and see if there is any button pressed, if there isn't it will jump to LOOP1 and disable the 8253 timer IC, if there was it will repeat the process.

```
; 1MHz/440 = 2273 = 08E1H
S1:
MOV AL , 0E1H ;Store E1H in AL
MOV DX , 0041H;Address of counter-1 8253 stored in DX
OUT DX , AL   ;Transfer it the address 41H which is counter-1

MOV AL , 08H  ;Store 08H in AL
MOV DX , 0041H;Address of counter-1 at 8253 stored in DX
OUT DX , AL   ;Transfer it the address 41H which is counter-1

; (ENABLING THE 8253 TIMER TO GENERATE THE SIGNAL THEN GET THE AUDIO)
MOV AL , 01H  ;Store 01H in AL
MOV DX , 0FF29H;Store the Address of Port B at 8255 in DX
OUT DX , AL    ;Transfer 1 to the address 0FF29H which is Port B connected to GATE1

JMP MAIN       ; jump again to the main and determine if there is any push button pressed
```

Figure 9: Label S1 which is specified for the first button and has 440Hz as frequency.

The rest of the labels (S2-S7):

```
S2: ; 1MHz/493.88 = 2025 = 07E9H
MOV AL , 0E9H
MOV DX , 0041H
OUT DX , AL
MOV AL , 07H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
S3: ; 1MHz/554.37 = 1804 = 070CH
MOV AL , 0CH
MOV DX , 0041H
OUT DX , AL
MOV AL , 07H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
S4: ; 1MHz/587.33 = 1703 = 06A7H
MOV AL , 0A7H
MOV DX , 0041H
OUT DX , AL
MOV AL , 06H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
```

Figure 10: The Rest of the Labels. (S2-S4).

```
S5: ; 1MHz/659.26 = 1517 = 05EDH
MOV AL , 0EDH
MOV DX , 0041H
OUT DX , AL
MOV AL , 05H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
S6: ; 1MHz/739.99 = 1352 = 0548H
MOV AL , 48H
MOV DX , 0041H
OUT DX , AL
MOV AL , 05H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
S7: ; 1MHz/830.61 = 1207 = 04B4H
MOV AL , 0B4H
MOV DX , 0041H
OUT DX , AL
MOV AL , 04H
MOV DX , 0041H
OUT DX , AL
MOV AL , 01H
MOV DX , 0FF29H
OUT DX , AL
JMP MAIN
```

Figure 11: The Rest of the Labels. (S5-S7).

Experimental Results and Discussion:

After doing the experiment and debugging the assembly code the results was 100% correct and each push button matched the required frequency we needed and for each button from S1-S7 there was a different tone and a different kind of tune at the speaker, so the experiment results were perfectly matching the requirements.

Conclusions:

1. Now we realized that when we combine more than IC, we could have an amazing practical experiment just like we did in this Piano experiment.
2. Now we are more familiar with the 8255 interfacing I/O IC and the 8253 Timer IC, and after doing this experiment our knowledge has expanded, with the modes, config register, addressing the ports and the counters, all of that positively increases our knowledge in those ICs.
3. Practiced more with longer codes than the previous experiments, specially that this experiment has the Jump instructions in it and the comparison instruction in it and all that increases our knowledge with the practical work that involves the assembly language in it.