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: 2023/2024	Experiment Name: A Musical Electronic Organ
Semester:2nd	

Students				
1-Salsabeel Dwaikat	2-Doaa Yasin Jararaa			
3-Wala’ Essam Ashqar	4-			
Performed on: 20/2/2024	Submitted on:27/2/2024			
Report’s Outcomes				
ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %	ILO __ =( ) %

Evaluation Criterion	Grade	Points
Abstract answers to 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, it Presents sufficiently the theoretical basis.	1.5	
Apparatus/ Procedure Apparatus sufficiently described to enable another experimenter to identify the equipment needed to experiment. The procedure is 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 are appropriate in light of the conclusions. Correct grammar.	1	
Appearance The title page is complete, page numbers are applied, content is well organized, correct spelling, fonts are consistent, good visual appeal.	1	
Total	10	

Objectives:

in this Experiment, we have a couple of goals to achieve

- Further mastering of timer programming and its applications
- Understanding the basic method of using a microcomputer to produce music (every tone has a different frequency) using a timer (control to scale frequency).

introduction:

An electronic organ is a keyboard-based musical instrument that uses loudspeakers to radiate tone produced by electronic circuits.

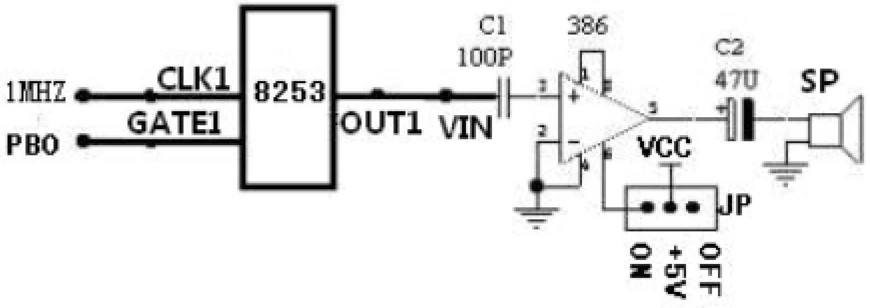
It is possible to construct this electronic organ with the 8253 PTI (programmable interval timer). The 8253 is used to produce various frequencies, or tones, that can then be amplified by an audio amplifier.

Audio Amplifier:

An electronic device that amplifies the power (amplitude) of audio signals passing through it is called an audio amplifier. Low-power audio signals are amplified by an audio amplifier to a volume appropriate for loudspeaker driving.

Experimental wiring diagram:

The wiring diagram of the 8253 is shown in the figure below. The PB0 (8255-PORTB) is used as the timer gating signal, and the OUT1 sends out audio signals to drive the speakers.

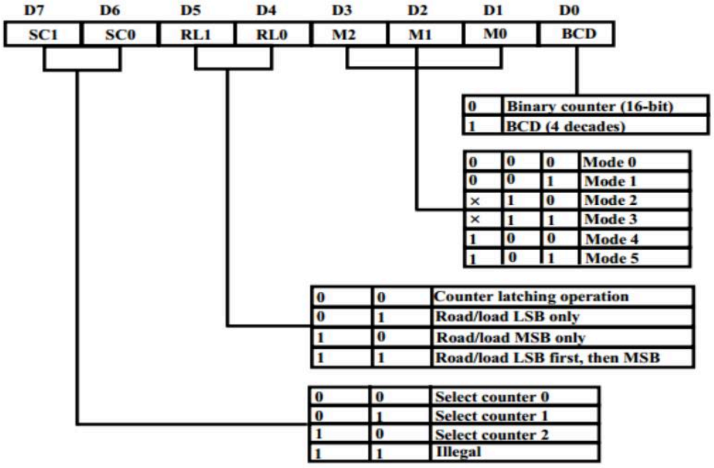


To program it:

Counter #	Address
Counter 0	40H
Counter 1	41H
Counter 2	42H
CWR	43H

Control word register:

8254 Control Word Format



D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

0	Binary counter (16-bit)
1	BCD (4 decades)

0	0	0	Mode 0
0	0	1	Mode 1
×	1	0	Mode 2
×	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

0	0	Counter latching operation
0	1	Road/load LSB only
1	0	Road/load MSB only
1	1	Road/load LSB first, then MSB

0	0	Select counter 0
0	1	Select counter 1
1	0	Select counter 2
1	1	Illegal

Fig. 14.9 Control Register

Register	Address
CWR	0FF2BH
PORT A	0FF28H
PORT B	0FF29H
PORT C	0FF2AH

Tools and equipment:

- MML8086K
  - INPUT: bush buttons
  - Output: buzzer
- 8253 IC
- 8255 IC
- Power supply

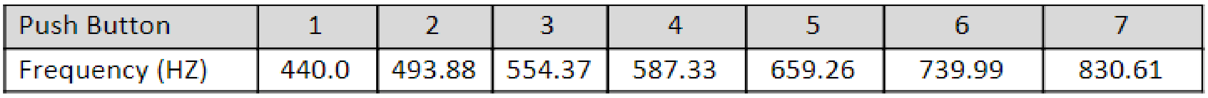
Procedure:

We were asked to write an assembly program to implement a simple electronic organ, The corresponding tones are selected by using the push-buttons [Btn1-Btn7] according to the scale frequency table:

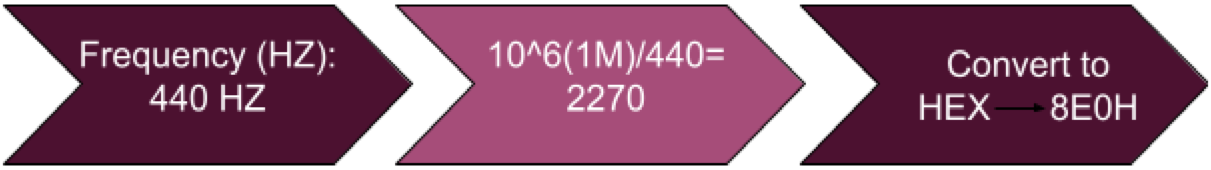
1. We initialized the 8255 to make port A an input and port B output. The result is 90h (1000 0000) Just like we reviewed in the introduction

2. We initialized counter 1 in order to work on mode 3, assuming the input frequency is 1 MHz, We divided the frequency by 2 bytes, and we used a binary system, so the result will be 76H.

3. We started writing the code to get the following frequencies using the 8253, and we measured n by the following equation n=F in/F out then we converted all the values to hexadecimal.



Push Button	1	2	3	4	5	6	7
Frequency (HZ)	440.0	493.88	554.37	587.33	659.26	739.99	830.61



Frequency (HZ): 440 HZ

10^6(1M)/440= 2270

Convert to HEX → 8E0H

Frequency Table:

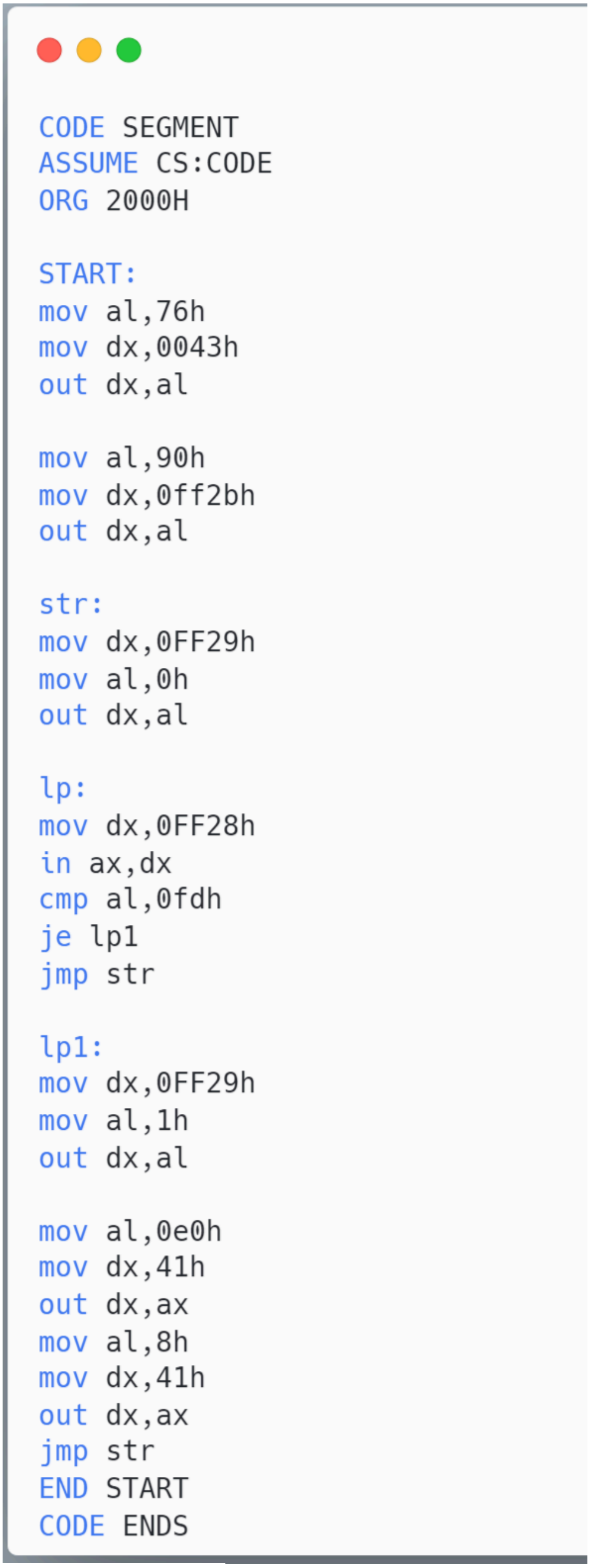
Frequency	1	2	3	4	5	6	7
CODE	8E0H	7E9H	70BH	6A6H	5ECH	547H	4B3H

To determine which button was pressed.
Because we have 16-bit mode 3 we divide it into 2 bytes (the least significant byte is 0E9H, and the most significant byte is 7H).

Code:

❖ First:

- ➔ we started by trying just one push button to make sure everything is okay and working.
- ➔ The program enters a loop (LP), where it continuously checks for button presses and generates corresponding tones based on the frequency table.
- ➔ Inside the loop, it reads the input from the push buttons (in ax, dx), compares it to a specific value (cmp al, 0fdh), and generates the corresponding frequency signal using the timer.
- ➔ The loop continues until a button is pressed.



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

START:
mov al,76h
mov dx,0043h
out dx,al

mov al,90h
mov dx,0ff2bh
out dx,al

str:
mov dx,0FF29h
mov al,0h
out dx,al

lp:
mov dx,0FF28h
in ax,dx
cmp al,0fdh
je lp1
jmp str

lp1:
mov dx,0FF29h
mov al,1h
out dx,al

mov al,0e0h
mov dx,41h
out dx,ax
mov al,8h
mov dx,41h
out dx,ax
jmp str
END START
CODE ENDS
CODE SEGMENT
ASSUME CS:CODE
ORG 2000H

START:
mov al,76h
mov dx,0043h
out dx,al

mov al,90h
mov dx,0ff2bh
out dx,al

str:
mov dx,0FF29h
mov al,0h
out dx,al

lp:
mov dx,0FF28h
in ax,dx
cmp al,0fdh
je lp1
cmp al,0fbh
je lp2
cmp al,0f7h
je lp3
cmp al,0efh
je lp4
cmp al,0dfh
je lp5
```

```
cmp al,0bfh

je lp6

cmp al,07fh

je lp7


jmp str


lp1:

mov dx,0FF29h

mov al,1h

out dx,al

mov al,0e0h

mov dx,41h

out dx,al

mov al,8h

mov dx,41h

out dx,al

jmp lp


lp2:

mov dx,0FF29h

mov al,1h

out dx,al

mov al,0e9h

mov dx,41h

out dx,al

mov al,7h

mov dx,41h

out dx,al

jmp lp
```

```
lp3:
mov dx,0FF29h
mov al,1h
out dx,al
mov al,0Bh
mov dx,41h
out dx,al
mov al,7h
mov dx,41h
out dx,al
jmp lp

lp4:
mov dx,0FF29h
mov al,1h
out dx,al
mov al,0a6h
mov dx,41h
out dx,al
mov al,6h
mov dx,41h
out dx,al
jmp lp

lp5:
mov dx,0FF29h
mov al,1h
out dx,al
mov al,0ech
mov dx,41h
out dx,al
```

```
mov al,5h

mov dx,41h

out dx,al

jmp lp


lp6:

mov dx,0FF29h

mov al,1h

out dx,al

mov al,47h

mov dx,41h

out dx,al

mov al,5h

mov dx,41h

out dx,al

jmp lp


lp7:

mov dx,0FF29h

mov al,1h

out dx,al

mov al,0b3h

mov dx,41h

out dx,al

mov al,4h

mov dx,41h

out dx,al

jmp lp

END START

CODE ENDS
```