Experiment # 7

Generating Music using the 8254 PIT on PC

Objective:

The objective is to understand, configure and test the 8253/4 Programmable Interval Timer (PIT) devices, on the personal computer.

Prelab

1. Review the PIT different modes and configuration from your microprocessor book and/or the 8253/4 datasheets.
2. Review the material below on the PITs in the personal computer.
3. Review the simple I/O mode of PPI (mode 0 of 8255).
4. Prepare all necessary code for parts A and B.

Introduction

In the PC there is a single clock used to synchronize activities of all peripheral chips connected to the CPU. The clock, which has the highest frequency in the system, belongs to the CPU. There are functions within the PC that require a clock with a lower frequency. The PIT (8253/54) is used to bring down the frequency to the desired level for various uses such as the beep sound in the PC. The 8254 PIT provides three independent channel timers that are programmed using the control (command) register of the PIT.

The Peripherals in Your PC

Fig. 1 shows the port address decoding in the PC. Fig. 2 shows the PPI connections with address lines while Fig. 3 shows the PIT connections with the address lines.

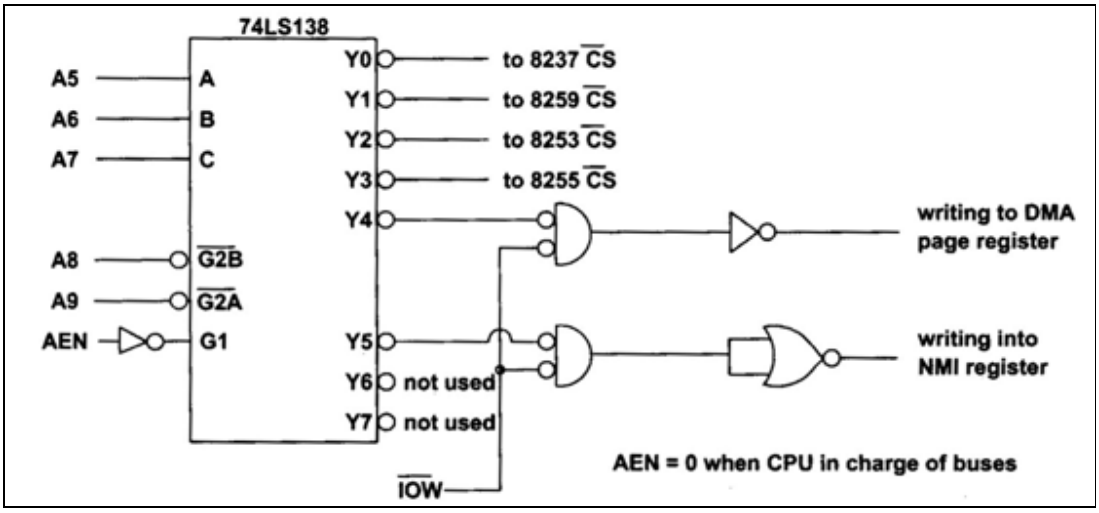


Figure 1

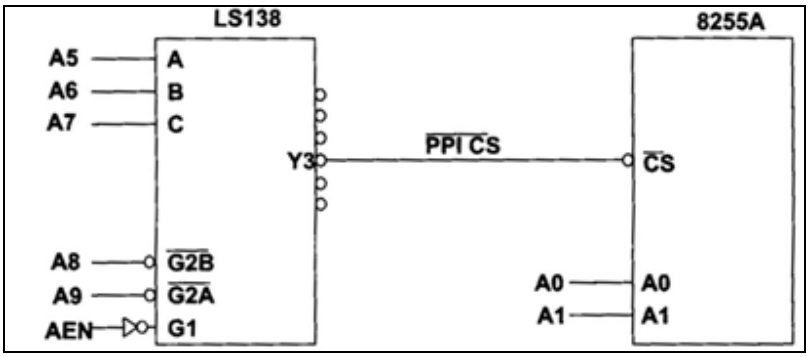


Figure 2

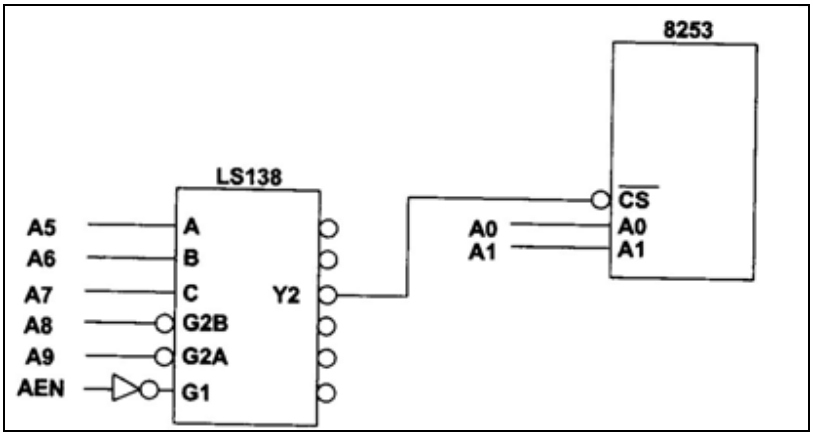


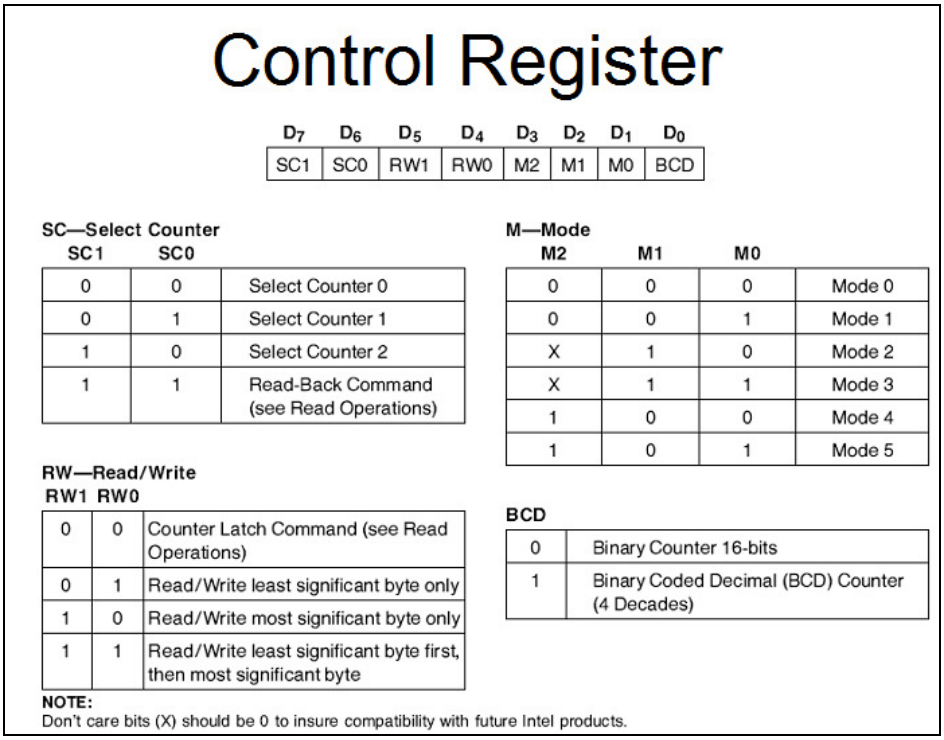
Figure 3

<u>Task 1:</u> Find the port addresses for the PPI and the PIT and fill them in the following table.

Port	Address in PC
PPI Port A	
PPI Port B	
PPI Port C	
PPI Command register	
PIT Counter 0	
PIT Counter 1	
PIT Counter 2	
PIT Command register	

Programming of the PIT

Fig. 4 shows the configuration of the control word to program the PIT counters. You can read more about this in the Microprocessor textbook.



Control Register

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RW1	RW0	M2	M1	M0	BCD

SC—Select Counter

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Read-Back Command (see Read Operations)

M—Mode

M2	M1	M0	
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

RW—Read/Write

RW1	RW0	
0	0	Counter Latch Command (see Read Operations)
0	1	Read/Write least significant byte only
1	0	Read/Write most significant byte only
1	1	Read/Write least significant byte first, then most significant byte

BCD

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

NOTE:
Don't care bits (X) should be 0 to insure compatibility with future Intel products.

Figure 4

System Timer Modes

The system timer has six modes as shown in the following table.

Mode	Name
0	Interrupt on Terminal Count
1	Hardware Re-triggerable One-Shot
2	Rate Generator
3	Square Wave Generator

4	Software Triggered Strobe
5	Hardware Re-triggerable Strobe

Common timer mode operations

All modes have the following operations in common:

- The counter logic is reset when control bytes are written to a counter.
- Counters do not stop when they reach zero.
- In modes 0, 1, 4 and 5, the counter wraps to the highest possible count, and continues to count.
- In modes 2 and 3, the counter reloads the initial count and continues to count.

PIT connections in the PC

Fig. 5 shows the PIT connections in your PC (it shows the connections of each counter of the PIT).

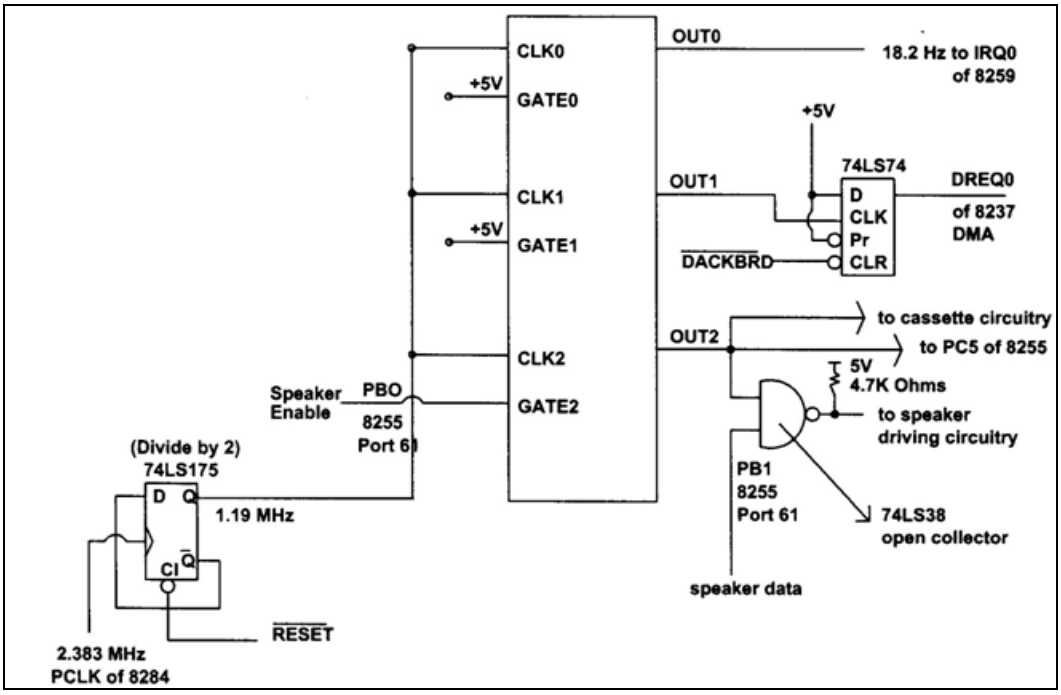


Figure 5

In this experiment, we will use the 8254 timer 2 which is connected with the speaker. We will use it to produce different beep sounds and to generate music. Note that counter 2 clock (i.e. clk2 is connected with 1.19 MHz) while Gate 2 is connected with PB0 of the PPI (i.e. bit 0 of port B in the PPI) and out2 is connected with speaker. For counter 2 to be enabled then gate 2 should be logic 1 otherwise it will be disabled. Also note that PB1 is connected with out2 to the speaker through NAND gate, thus PB1 should be 1 to enable out2 to go the speaker.

PROCEDURE

A. Generating Beep using debug

1. Start debug.

2. Program the command register such that counter 2 to generate a square wave (mode 3)
   - **O PIT_CTRL_Register , Control_Word**

3. Program counter 2 to produce a frequency of 10 KHz
   - **O _____ , _____**
   - **O _____ , _____**

4. Enable Gate2 and the speaker
   - **O _____ , _____**

5. Disable Gate2 and the speaker
   - **O _____ , _____**

6. Repeat the above steps to produce beep with frequency 1 KHz, 5 KHz and 15 KHz.

B. Using TASM to produce beep sounds

Copy the following code to a text editor to make an Assembly code. Use TASM and TLINK to produce the exe file.

```
.model small
.stack 1000h
.data
count  equ  200
t  equ  500
.code

start:

mov al,0b6h
out 43h,al

mov ax,count
out 42h,al
mov al,ah
out 42h,al

mov al, 00000011b
out 61h,al


mov cx,t
delay1:
push cx
mov cx,20000
```

```
delay2:
loop delay2
pop cx
loop delay1


mov al,00000000b
out 61h,al

mov ax,4c00h
int 21h

end start
```

1. What does the code above do?
2. Explain the task of each block.
3. Change the code to produce a beep sound of 3 KHz for 5 seconds (approximately)
4. Change the code to produce a beep sound of 12 KHz for 2 seconds (approximately).

C. Generate Music on your PC

Fig. 6 shows some piano notes and frequencies. The following table shows how to play "Happy Birthday" on the PC.

Lyrics	Notes	Freq. (Hz)	Duration
hap	C4	262	½
py	C4	262	½
birth	D4	294	1
day	C4	262	1
to	F4	349	1
you	E4	330	2
hap	C4	262	½
py	C4	262	½
birth	D4	294	1
day	C4	262	1
to	G4	392	1
you	F4	349	2
hap	C4	262	½
py	C4	262	½
birth	C5	523	1
day	A4	440	1
dear	F4	349	1
so	E4	330	1
so	D4	294	3
hap	B4b	466	½
py	B4b	466	½
birth	A4	440	1
day	F4	349	1
to	G4	392	1
you	F4	349	2

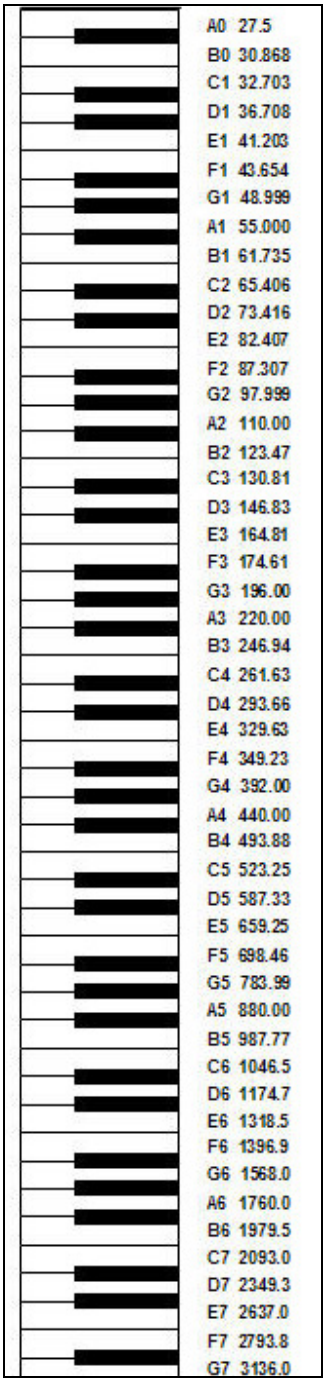


Figure 6

Complete the following Assembly code to play “Happy Birthday”.

```
.model small
.stack 1000h
.data
t equ 4000

.code

tone macro div,dur

mov al,0b6h
out 43h,al

mov ax,div
out 42h,al

mov al,ah
out 42h,al

mov al,00000011b
out 61h,al


mov cx,dur
call delay1

mov al,00000000b
out 61h,al

call delay2

endm

.startup
tone 4553,t     ; hap (c4)
tone 4553,t     ; py (c4)
tone 4057,2*t   ; birth (d4)

; Continue your code here……

mov ah,4ch
int 21h


delay1 proc near
d1:
push cx
mov cx,38000
d2:
loop d2
pop cx
loop d1
ret
delay1 endp

delay2 proc near
mov cx,65000
d3:
loop d3
ret
delay2 endp

end
```

D. Using Keyboard as Piano keys

In this task you will use the keyboard keys to generate different tones. Use the keys as in the following table

Keyboard Key	Tone	Duration (approximately)
A	A4	500 ms (½ t)
B	B4b	500 ms (½ t)
C	C4	500 ms (½ t)
D	D4	500 ms (½ t)
E	E4	500 ms (½ t)
F	F4	500 ms (½ t)
G	G4	500 ms (½ t)
H	C5	500 ms (½ t)

Use the keyboard to play “Happy birthday” music.