Programmable Interval Timer

MDA-8086 Kit–PPI Application

Birzeit University

Information Technology Faculty

Computer Systems Engineering Department

August, 2010

Abstract

This experiment aims at to understanding, configuring and testing the 8253/4 Programmable Interval Timer (PIT) devices, on the MDA 8086 kit and Personal Computers.

PART I Theoretical and Technical Introduction

1.1 The 8254A PIT in the personal computer

1.1.1 Introduction

The 8254A Programmable Interval Timer (PIT) is a counter and timer that provides three channel timers. All channels are driven by a 1.19MHz oscillator signal. Each "tick" of the PIT generates hardware interrupt request 0.

1.1.2 Timer channel differences

There are some differences between the three timer channels.

Counters 0 and 2:

- Are independent 16-bit counters.
- Can be preset.
- Can count in BCD (Binary Coded Decimal) or in binary.

Counter 3:

- Is only 8 bits.
- Can be preset.
- Counts in binary only.
- Can only count downward.

1.1.3 System Timer Modes

The system timer has six modes:

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

1.1.4 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.

1.1.5 Timer Channels

The following table describes the functions of the timer channels. The system timers are treated as a series of I/O ports. There are three counter registers, and two control registers.

Channel	I/O Port	Read/Write Status
0 System Timer	0040h	R/W
2 Tone Generator for Speaker	0042h	R/W
3 Watchdog Timer	0044h	R/W
Control Register 0, 2	0043h	W
Control Register 3	0047h	W

The speaker is controlled by the following I/O port

I/O Address	Read/Write Status	Description
0061h	W	System control port B, Where: Bit 7 = 1 Reset timer 0 output latch (IRQ 0) Bits 6-4 = Reserved Bit 3 = 0 Enable channel check Bit 2 = 0 Enable parity check Bit 1 = 1 Speaker data enable Bit 0 = 1 Enable timer 2 gate to speaker

* See the data sheets of the 8254A chip.

P.S. In this Experiment, we will use the 8254 timer 2 to generate different tones.

1.2 The 8253 PIT in the MDA-8086 Kit

Study the schematics shown in (Figure 1) for the 8253 PIT interface on the MDA-8086 kit.

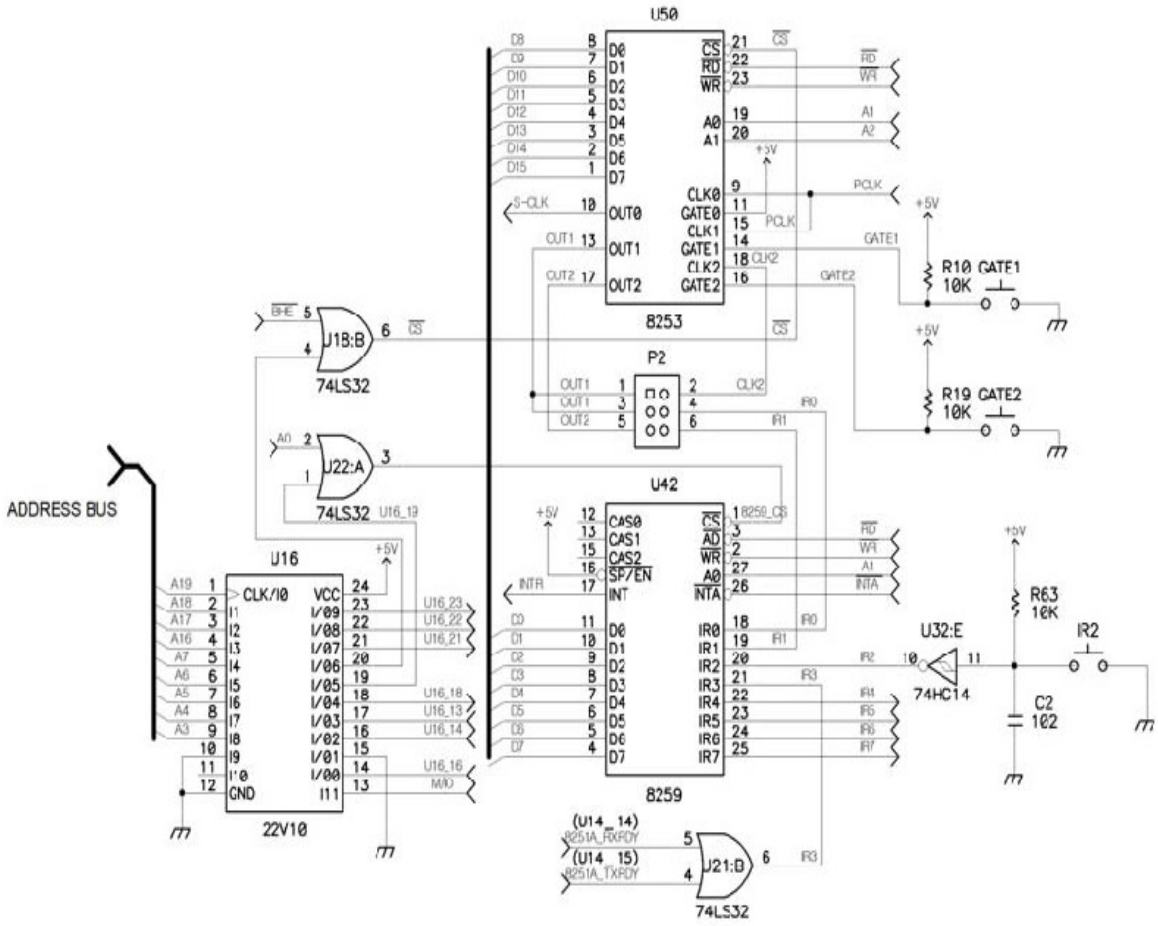

Figure 1 PIT Schematics on MDA-8086 Kit

Recall the I/O port addresses for the 8253 on the MDA-8086 kit:

09H : TIMER 0 REGISTER
0BH : TIMER 1 REGISTER
0DH : TIMER 2 REGISTER
0FH : CONTROL REGISTER

PART II Pre-Lab

(This part should be handed on to the teaching assistant in your Lab)

1. Review the PIT different modes and configuration from your microprocessor book and/or the 8253/4 datasheets.
2. Review the material about the PITs in the personal computer and the MDA 8086 kit.
3. Prepare all necessary code for the practices.

PART III Practices

3.1 PRACTICE I: Configuring PIT on PC

Step1: Using Dos Command prompt, start **debug**.

Step2: Type the following instructions.

O 42 11
O 42 11
O 61 33
O 61 32
O 42 55
O 42 3F
O 43 B0
O 42 11
O 42 11
O 42 FF
O 42 FF

What frequency dos drive the speaker?
What is the effect of this operation 3, 4 , 5?

TASKS:

1. Explain the above instructions one by one and show what is the result for each of them?
2. Find the control word at address 43 that enables the speakers with frequency of the count 1111.
3. What is the mode in which the timer operates?
4. What frequency dos drive the speaker
5. Write an assembly program that enables the speaker with the tone frequency about 5 KHz, 7 KHz, 12 KHz.
6. Modify the program so that the sequence of tones is repeated with a certain delay interruption between tones and about two seconds of interruption between the set of tones.

3.2 PRACTICE II: Configuring PIT on MDA-8086 Kit

Look-up the components on your kit guided by (Figure 1), and familiarize yourself with the hardware, especially U50, GATE1 and GATE2.

WARNING: Don't touch any exposed wiring or the pins of any of the ICs.

Step 1: Configure the P2 connector using a jumper cap as shown in (Figure 2).

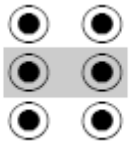

Figure 2 P2 Configuration

Step2: Write the following code and save it.

```
CODE    SEGMENT
    ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
    ;
PPIC_C  EQU 1FH
PPIC    EQU 1DH
PPIB    EQU 1BH
PPIA    EQU 19H
    ;
CTC1    EQU 0BH
CTCC    EQU 0FH
    ;
INTA    EQU 10H
INTA2   EQU INTA+2
    ;
INT_V   EQU 40H*4
    ;
    ORG 1000H
    ;
    XOR BX,BX
    MOV ES,BX
    ;
    MOV AX,OFFSET INT_SER
    MOV BX,INT_V
    MOV WORD PTR ES:[BX],AX
    ;
    XOR AX,AX
    MOV WORD PTR ES:[BX+2],AX
    ;
    CALL    INIT
    CALL    P_INIT
    ;
; 8255 Initialization
    MOV AL,10000000B
    OUT PPIC_C,AL
    ;
```

```
    MOV AL,11111111B
    OUT PPIA,AL
    ;
    MOV AL,00000000B
    OUT PPIC,AL
    ;
    MOV AH,11110001B
    MOV AL,AH
    OUT PPIB,AL
    STI
L2: NOP
    JMP L2
    ;
    INT 3
    ;
    ; The Interrupt Service Routine
INT_SER:
    SHL AH,1
    TEST    AH,00010000B
    JNZ L1
    OR  AH,11110000B
    JMP L3
    ; LED out
L1: MOV AH,11110001B
L3: MOV AL,AH
    OUT PPIB,AL
    ;
    PUSH    AX
    MOV AX,0FFFFH
    OUT CTC1,AL
    MOV AL,AH
    OUT CTC1,AL
    POP AX
    ;  EOI command
    MOV AL,00100000B
    OUT INTA,AL
    STI
    IRET
    ;
    ; 8253 Initialization
P_INIT  PROC    NEAR
    PUSH    AX
    MOV AL,01110000B
    OUT CTCC,AL
    ;
    MOV AX,0FFFFH
    OUT CTC1,AL
    MOV AL,AH
    OUT CTC1,AL
    POP AX
    RET
P_INIT  ENDP
    ;
    ; 8259 Initialization
INIT    PROC    NEAR
    ; ICW1
    MOV AL,00010011B
    OUT INTA,AL
    ;ICW2 interrupt vector
    MOV AL,40H
    OUT INTA2,AL
    ;ICW4
    MOV AL,00000001B
    OUT INTA2,AL
    ;interrupt mask
```

```
101     MOV     AL,11111110B
102     OUT INTA2,AL
103     RET
104 INIT    ENDP
105     ;
106 CODE    ENDS
107     END
```

Figure 3 Code 1

Step2: Compile and build this ASM file and execute it on MDA-8086 kit. (How? Review Exp#1 Intro. To MDA Kit)

P.S. The purpose of the code is to turn on a different LED after a certain amount of time, as follows:

TASKS:

1. Explain what does this code do?
2. Does it match the pattern described above?
3. What happens when you press the GATE1 button? Explain.
4. Which counter are we using in the 8253?
5. Under which mode is the counter working?
6. Knowing that the input clock frequency (PCLK) is 2.5MHz, what is the delay produced by the counter?
7. Try changing to the other modes.

P.S. You may press the RESET key of the MDA-8086 kit to stop the program.

8. Change the code (and the jumpers on connector P2) to increase the delay by a factor of 5 times. **(Hint: Use Timer 1 to divide the input clock frequency by 5 and feed it to Timer 2)**. Also, answer the questions from 1- 7 above, after completing this part.

Bibliography

Tech., MEDAS. 2008. *MDA 8086 Kit User Manual.* Korea : s.n., 2008.