Faculty of Engineering and Technology
Department of Electrical and Computer Engineering

ENCS 4110
COMPUTER DESIGN LABORATORY
Report #3
Experiment #10

"8259 Programmable Interrupt Controller Application"

Prepared by:

Nsreen Tawafsha – 1182319

Supervised by:

Dr. Ahmad Afaneh

Teacher Assistant:

Eng. Heba Qadah

Section: 4

Date: 18 May 2021

1. Abstract

The aim of this experiment is to examine Intel 8259 control, setup, and operational modes in order to better understand and extend 8086 Interrupt capabilities using Intel 8259 PIC. We'll also use PPI 8255 (programmable peripheral interface) and its various operating modes and PIT 8253/4 (Programmable Interval Timer) in this experiment.

Table of Content

List of Figures

2. Theory

2.1 PPI 8255A:-

PPI 8255 is a general purpose programmable I/O device designed to interface the CPU with its outside world such as ADC, DAC, keyboard etc. We can program it according to the given condition. It can be used with almost any microprocessor.

It consists of three 8-bit bidirectional I/O ports i.e. PORT A, PORT B and PORT C. We can assign different ports as input or output functions.

✓ Ports of 8255A:

8255A has three ports, i.e., PORT A, PORT B, and PORT C.

- **Port A** contains one 8-bit output latch/buffer and one 8-bit input buffer.
- **Port B** is similar to PORT A.
- **Port C** can be split into two parts, i.e. PORT C lower (PC0-PC3) and PORT C upper (PC7-PC4) by the control word.

These three ports are further divided into two groups, i.e. Group A includes PORT A and upper PORT C. Group B includes PORT B and lower PORT C. These two groups can be programmed in three different modes, i.e. the first mode is named as mode 0, the second mode is named as Mode 1 and the third mode is named as Mode 2.

✓ Operating Modes:

8255A has three different operating modes:

- **Mode 0:** In this mode, Port A and B is used as two 8-bit ports and Port C as two 4-bit ports. Each port can be programmed in either input mode or output mode where outputs are latched and inputs are not latched. Ports do not have interrupt capability.

- **Mode 1:** In this mode, Port A and B is used as 8-bit I/O ports. They can be configured as either input or output ports. Each port uses three lines from port C as handshake signals. Inputs and outputs are latched.

- **Mode 2:** In this mode, Port A can be configured as the bidirectional port and Port B either in Mode 0 or Mode 1. Port A uses five signals from Port C as handshake signals for data transfer. The remaining three signals from Port C can be used either as simple I/O or as handshake for port B.

✓ 8255 Architecture:

PPI block diagram is shown in **Figure 2.1.1** below.

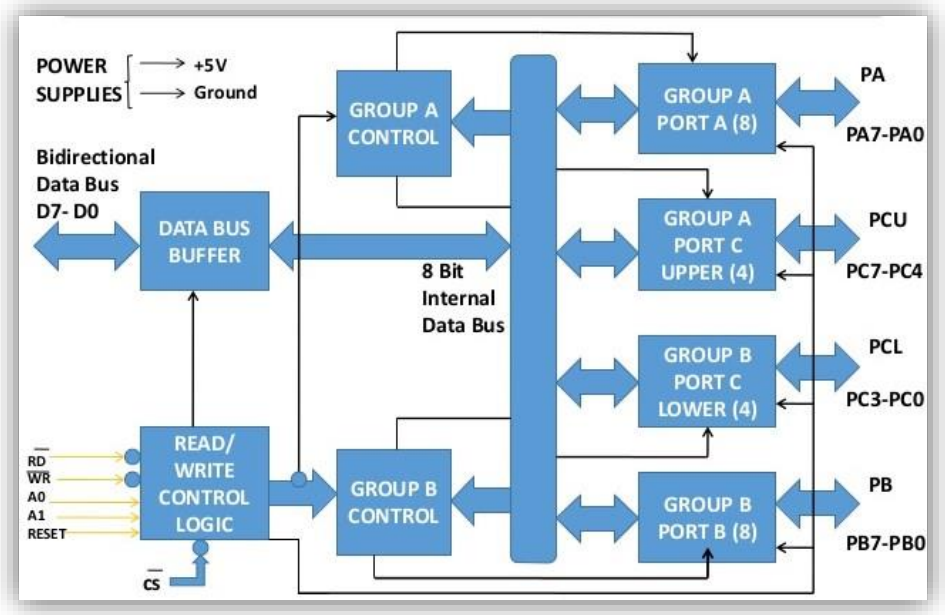

Figure 2.1.1: PPI 8255A block diagram

✓ 8255 Configuration using command byte:

Figure 2.1.1 shown PPI Configuration.

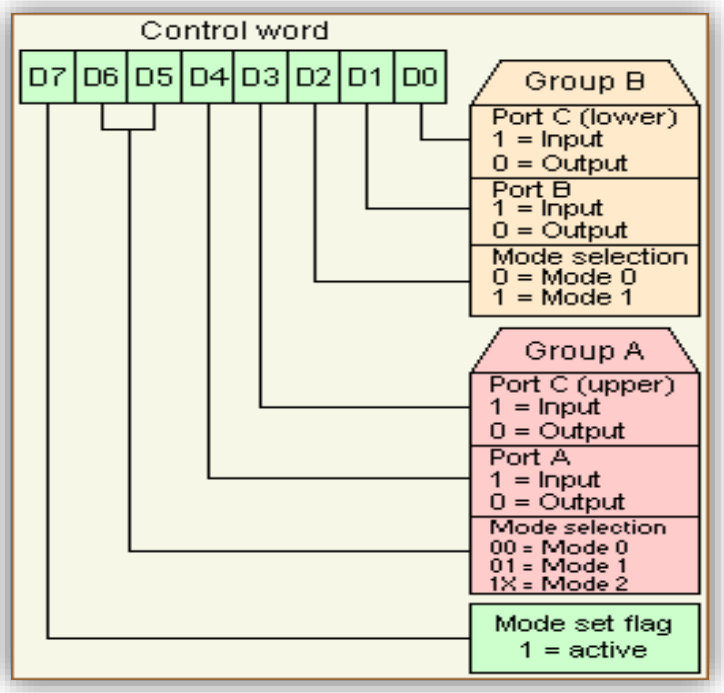

Figure 2.1.1: PPI Configuration

2.2 PIT 8253:-

The Intel 8253 Programmable Interval Timers (PTIs) designed for microprocessors to perform timing and counting functions using three 16-bit registers. Each counter has 2 input pins, i.e. Clock & Gate, and 1 pin for "OUT" output. To operate a counter, a 16-bit count is loaded in its register. On command, it begins to decrement the count until it reaches 0, then it generates a pulse that can be used to interrupt the CPU. **Figure 2.2.1 shows** 8253 Diagram, there are three counters, a data bus buffer, Read/Write control logic, and a control register. Each counter has two input signals - CLOCK & GATE, and one output signal - OUT.

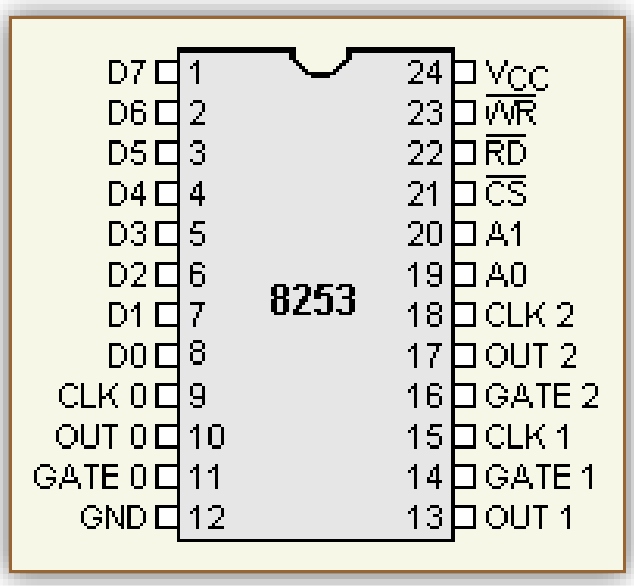

Figure 2.2.1: PIT 8253 Diagram

✓ Operating Modes:

8253 has six different operating modes:

- **Mode 0:** Interrupt on Terminal Count.
- **Mode 1:** Hardware Retriggerable One-Shot.
- **Mode 2:** Rate Generator.
- **Mode 3:** Square Wave Generator.
- **Mode 4:** Software Triggered Strobe.
- **Mode 5:** Hardware Retriggerable Strobe.

Table 2.2.2 shows the different uses of the 8253 gate input pin. Each mode of operation for the counter has a different use for the GATE input pin.

Table 2.2.2: PIT 8253 modes

Signal Status	Low or going low	Rising	High
Mode			
0	Disables counting	- -	Enables counting
1	- -	1) Initiates counting 2) Resets output after next clock	- -
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	- -	Enables counting
5	- -	Initiates counting	- -

✓ 8253 Architecture

Figure 2.2.3 shows the PIT 8253 Architecture.

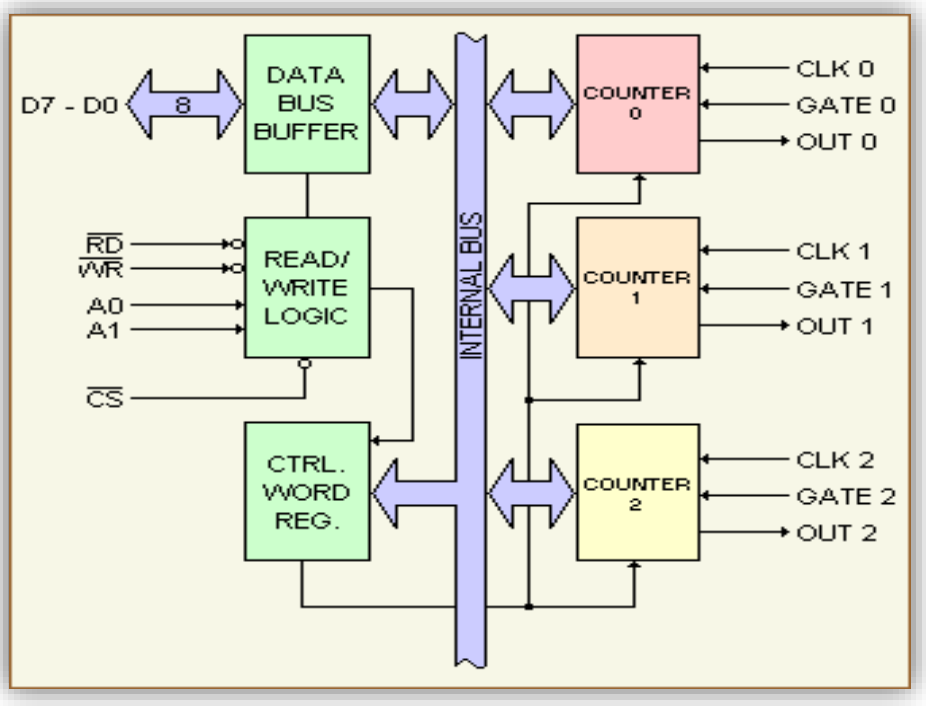

Figure 2.2.3: PIT 8253 Architecture

✓ 8253 Configuration using command byte:

Figure 2.2.4 shows the PIT 8253 Configuration.

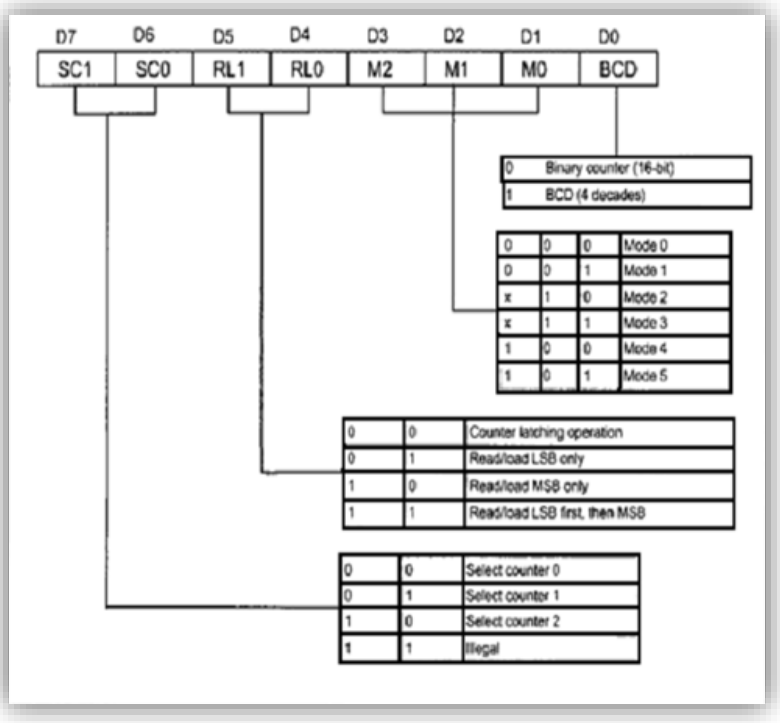

Figure 2.2.4: PIT 8253 Configuration

2.3 8086 Interrupts:-

In 8086 microprocessor following tasks are performed when microprocessor encounters an interrupt:

1) The value of flag register is pushed into the stack. It means that first the value of SP (Stack Pointer) is decremented by 2 then the value of flag register is pushed to the memory address of stack segment.
2) The value of starting memory address of CS (Code Segment) is pushed into the stack.
3) The value of IP (Instruction Pointer) is pushed into the stack.
4) IP is loaded from word location (Interrupt type) * 04.
5) CS is loaded from the next word location.
6) Interrupt and Trap flag are reset to 0.

The different types of interrupts present in 8086 microprocessor are given by:

- Hardware Interrupts:-

  Hardware interrupts are those interrupts which are caused by any peripheral device by sending a signal through a specified pin to the microprocessor. There are two hardware interrupts in 8086 microprocessor. They are:

  1) NMI (Non Maskable Interrupt):
     It is a single pin non maskable hardware interrupt which cannot be disabled. It is the highest priority interrupt in 8086 microprocessor. After its execution, this interrupt generates a TYPE 2 interrupt. IP is loaded from word location 00008 H and CS is loaded from the word location 0000A H.

  2) INTR (Interrupt Request):
     It provides a single interrupt request and is activated by I/O port. This interrupt can be masked or delayed. It is a level triggered interrupt. It can receive any interrupt type, so the value of IP and CS will change on the interrupt type received.

- Software Interrupts:-

  These are instructions that are inserted within the program to generate interrupts. There are 256 software interrupts in 8086 microprocessor. The instructions are of the format INT type where type ranges from 00 to FF. The starting address ranges from 00000 H to 003FF H. These are 2 byte instructions. IP is loaded from type * 04 H and CS is loaded from the next address given by (type * 04) + 02 H. Some important software interrupts are:

  1) TYPE 0 corresponds to division by zero(0).
  2) TYPE 1 is used for single step execution for debugging of program.
  3) TYPE 2 represents NMI and is used in power failure conditions.
  4) TYPE 3 represents a break-point interrupt.
  5) TYPE 4 is the overflow interrupt.

2.4 8259 PIC Microprocessor:-

8259 microprocessor is defined as Programmable Interrupt Controller (PIC) microprocessor. There are 5 hardware interrupts and 2 hardware interrupts in 8085 and 8086 respectively. But by connecting 8259 with CPU, we can increase the interrupt handling capability. 8259 combines the multi interrupt input sources into a single interrupt output. Interfacing of single PIC provides 8 interrupts inputs from IR0-IR7. **Figure 2.4.1 shows** 8259 Pin Diagram.

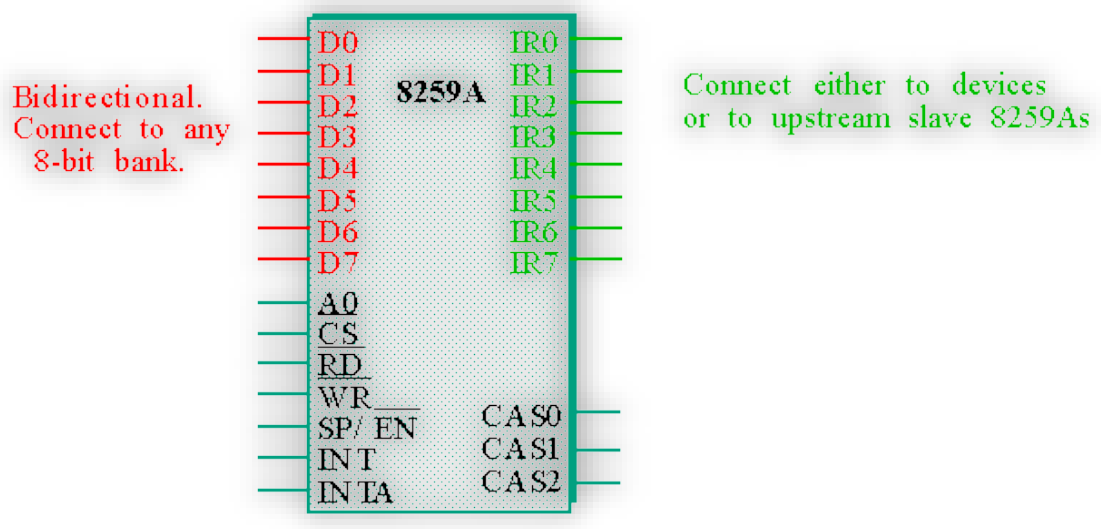

Figure 2.4.1: 8259 Pin Diagram

- INT Pin: Connects to the INTR pin on the microprocessor.
- INTA Pin: Connects to the INTA pin on the microprocessor.
- A0 Pin: Selects different Command WORDS
- CS Pin: Enables the chip for programming and control.
- SP/EN Pin: Slave program (SP) / Enable Buffer (EN).
  - ✓ Slave Program (1=Master, 0=Slave)
  - ✓ Enable Buffer (Controls data bus transceivers when in buffered mode)
- Interrupt request inputs (IR0-IR7): are used to request an interrupt and to connect to a slave in a system with multiple 8259As. An interrupt request is executed by raising an IR input (low to high), and holding it high until it is acknowledged (Edge Triggered Mode), or just by a high level on an IR input (Level Triggered Mode).
- D0 – D7 Pins: 8 bit Data connector pins.

The 8259A requires two types of control words:

1) Initialization Command Words (ICWs): are programmed before the 8259A is able to function in the system and dictate the basic operation of the 8259A.
2) Operational Command Words (OCWs): are programmed during the normal course of operation. The OCWs control the operation of the 8259A.

3. Procedure and Discussion

We must first specify the addresses for all Peripherals inside (PIC, PPI and PIT) before we can begin the experiment's tasks. We do that using 3-8 decoder shown in **Figure 3.3.1**

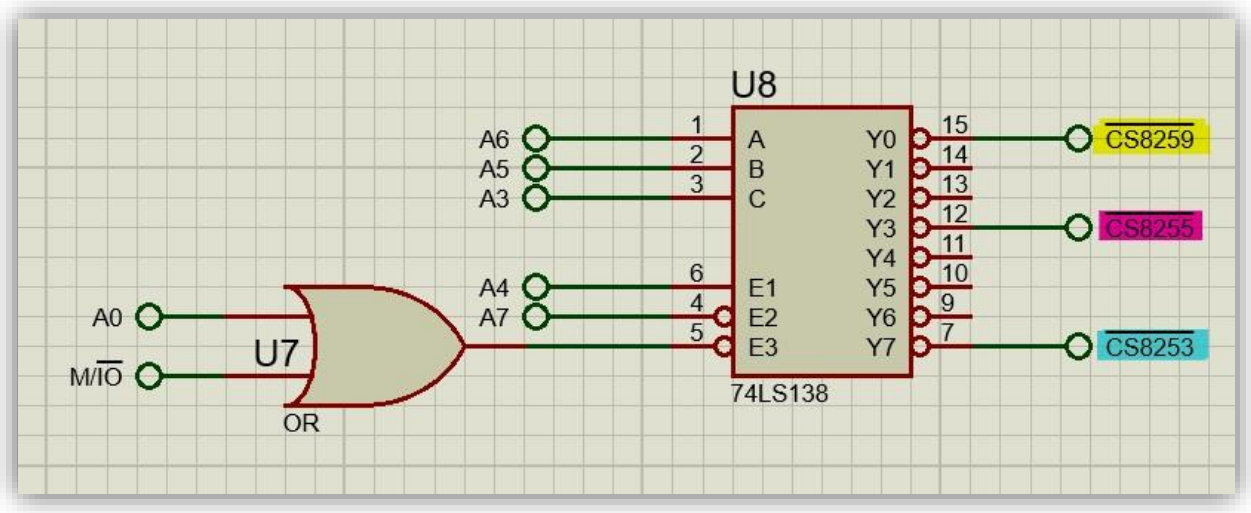

Figure 3.3.1: 3-8 Decoder

Figure 3.3.1 shows a 3-8 Decoder. Y0 is connected to the chip select of PIC, Y3 is connected to the chip select of PPI and Y7 is connected to the chip select of PIT. Then, A6, A5, A3 will determine the base address for each of them as follows:

A7	A6	A5	A4	A3	A2	A1	A0	
0	0	0	1	0	0	0	0	➔ 10h for PIC
0	1	1	1	0	0	0	0	➔ 70h for PPI
0	1	1	1	1	0	0	0	➔ 78h for PIT

So the port addresses for the PPI and the PIT is as follows:

PIC	**PPI**	**PIT**
Address1: 10h Address2: 12h	Port A: 70h Port B: 72h Port C: 74h Control register: 76h	Counter 0: 78h Counter 1: 7Ah Counter 2: 7Ch Command register: 7Eh

3.1 Part A: Automatic counter

Figure 3.1.1 shows the circuit for automatic counter.

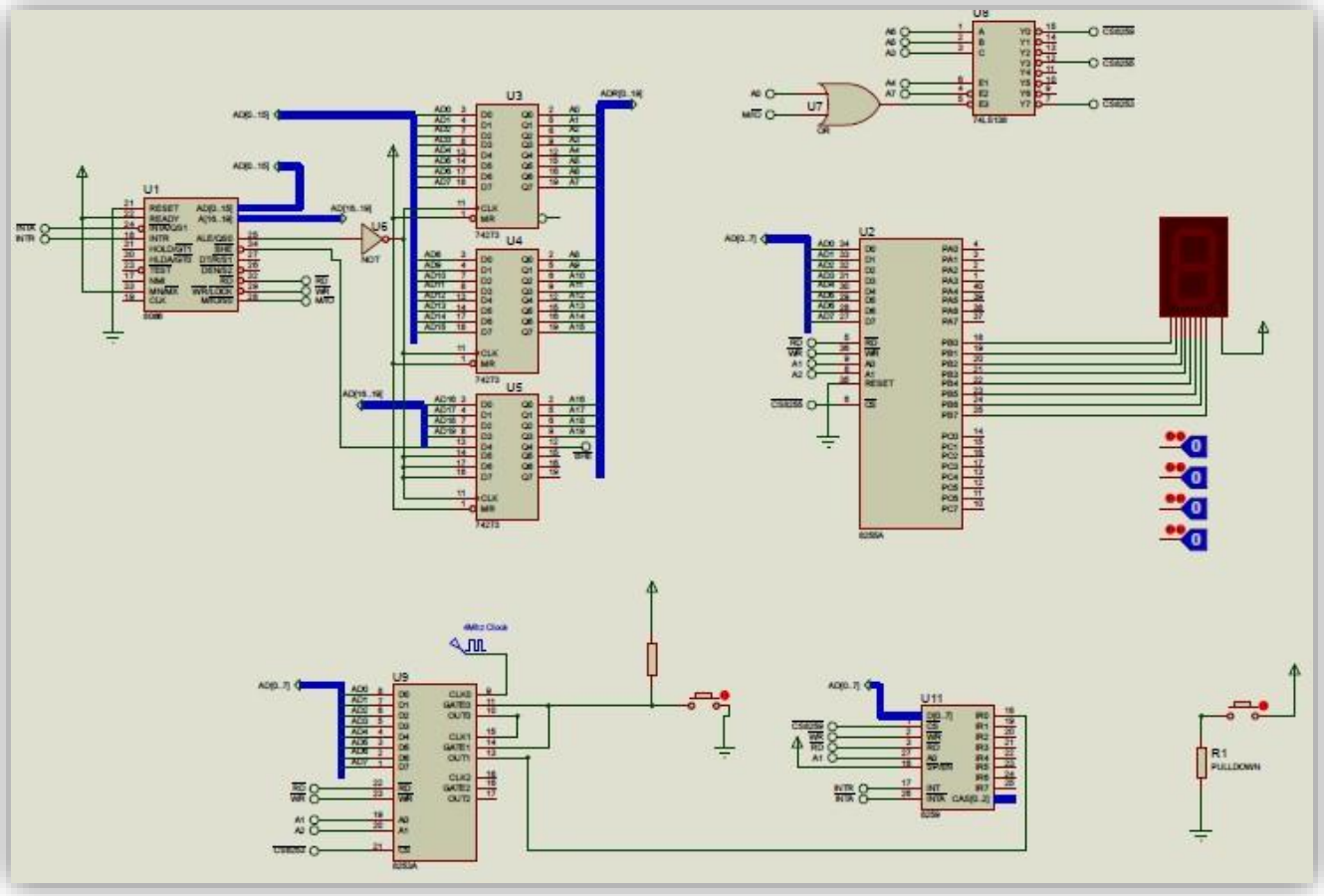

Figure 3.1.1: Automatic counter circuit

The code written at Appendix A used to generate an automatic counter from 0 to F printed in the 7-segment using the circuit in **Figure 3.1.1**.

Here in **Figure 3.1.2** the numbers representation was defined (C0 represents the representation if the number zero at the 7-segment and so on).

```
DIGITS  DB 0C0h, 0F9h, 0A4h, 0B0h, 99h, 92h, 82h, 0F8h, 80h, 90h, 088h, 083h, 0C6h, 0A1h, 86h, 8Eh    , 00 ; 0 to F
DATA    ENDS
```

Figure 3.1.2: Numbers representation

```
START PROC
  MOV AX, DATA
  MOV DS, AX

CALL INSERT_ISR_INTO_VECTOR_TABLE
    CALL SETUP_8259
    ;; Address of 8253: 78h
    ;; Addrss of  8255: 70h
    ;; Address of 8259: 10h
    MOV AL, 80h            ; 1000 0000b
    OUT 76h, AL            ; Setup 8255 as all ports mo0 output
    CALL SETUP_8253

    STI
    XOR AX,AX
ENDLESS:
    CALL SHOW_AX

    CMP AX, 10h
    JNZ ENDLESS
    MOV AX, 0h          ; MAke AX 0 after reaching 16

    JMP ENDLESS
RET
START ENDP
```

Figure 3.1.3: Re-count

Here the code in **Figure 3.1.3** represents that all Ports (Port A, Port B and Port C) are output ones and both Port A and Port B are in Mode 0. Also count from 0 to F which stored in AX, if it reaches the number 16 (F+1 = 10h) then clear AX register to store 0 and re-count.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;          Common Anode Seven Segmeent ; 0 for lighting, 1 otherwise
SHOW_AX PROC NEAR
  PUSH AX

  XOR BX, BX
  MOV BL, AL
  MOV AL, DIGITS[BX]
  OUT 72h, AL                    ; 8255's Port B at 72h

  POP AX
RET
SHOW_AX ENDP
```

Figure 3.1.4: prints the number

In the code above at **Figure 3.1.4** shows the part which make it possible to print the number we reached at the seven-segment screen, by sending the number to the seven-segment using Port B.

In this code at **Figure 3.1.5** all the code explained inside, using the comments.

```
SETUP_8253 PROC NEAR
    ;; SETUP 8253

    ; --> Used to Divide 4Mhz Clock to 2000 so that we get 2Khz Output
    MOV AL, 00110111b     ; 00 [counter 0], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
    OUT 7Eh, AL           ; configure the counter 0 of 8253

    ; --> Pulse Wave  :: MOD 2 of 8253
    MOV AL, 01110111b     ; 01 [counter 1], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
    OUT 7Eh, AL           ; configure the counter 1 of 8253

    MOV AL, 00h
    OUT 78h, AL           ; Send first byte of Counter 0
    MOV AL, 20h
    OUT 78h, AL           ; Send second byte of Counter 0
    ;;--> Counter0 : 2000h    ; send BCD 2000 to counter 0

    MOV AL, 00h
    OUT 7Ah, AL           ; Send first byte of Counter 1
    MOV AL, 5h
    OUT 7Ah, AL           ; Send second byte of Counter 1
    ;;--> Counter1 : 5h ; send BCD 5 to counter 1
RET
SETUP_8253 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8259 PROC NEAR

MOV AL, 00010011b         ; ICW1 : Edge Triggered - 1 Master - ICW4 Needed
OUT 10h, AL               ; SET InputControlWord1

MOV AL, 40h               ; ICW2 : 40h
OUT 12h, AL               ; SET InputControlWord2

MOV AL, 03h               ; AEOI(Automatic Interrupt) = 1, 8086= 1
OUT 12h, AL               ; SET InputControlWord4

RET
SETUP_8259 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

Figure 3.1.5: The rest of the code

The code in **Figure 3.1.6** below at shows the end of the code, and the part where each vector represents an interrupt.

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INSERT_ISR_INTO_VECTOR_TABLE PROC NEAR
  XOR AX, AX
  MOV ES, AX
  MOV AL, 40H
  MOV AH, 4
  MUL AH
  MOV BX, AX
  LEA AX, INC_AX_PER_15_SEC
  MOV WORD PTR ES:[BX], AX
  MOV AX, CS
  MOV WORD PTR ES:[BX+2], AX
RET
INSERT_ISR_INTO_VECTOR_TABLE ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Interrupt Routine
INC_AX_PER_15_SEC PROC FAR
  INC AX
IRET
INC_AX_PER_15_SEC ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

CODE    ENDS
      END START
```

Figure 3.1.6: End of the code

3.2 Part B: The task -> Manual counter

The code written at Appendix B used to generate the manual counter.

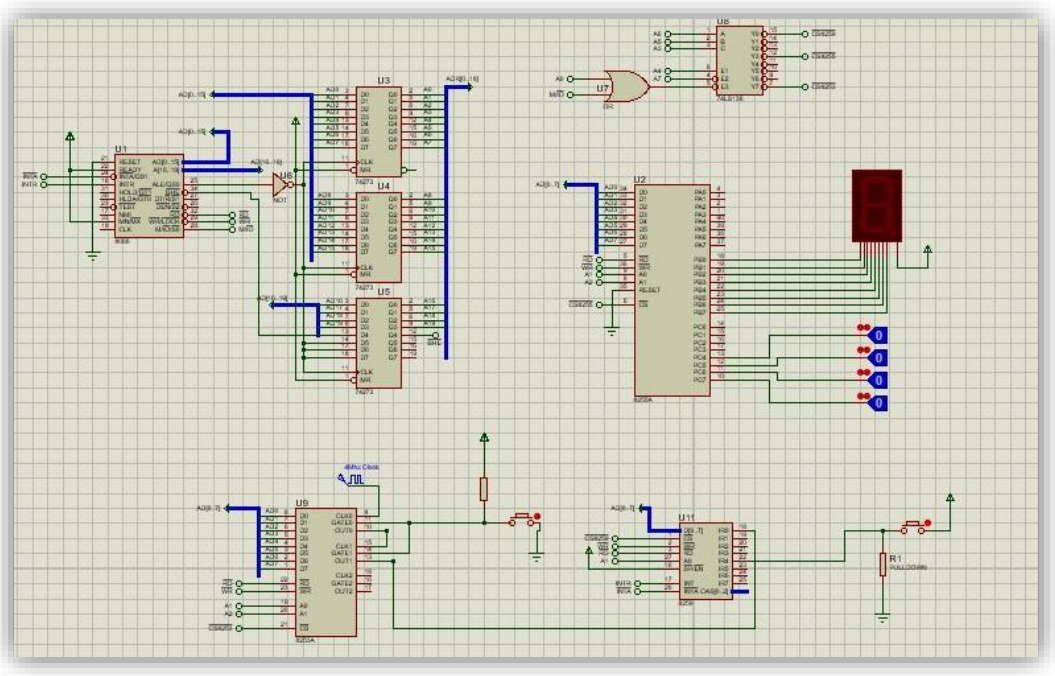

Figure 3.2.1: Manual counter circuit

4. Conclusion

In this experiment, we learned how to the PIC works and how we can program it.

5. References

[1] https://www.tutorialspoint.com/microprocessor/microprocessor_intel_8255a_programmable_peripheral_interface.htm Access Date: 18-5-2021 at 6:00PM.

[2] http://discipline.elcom.pub.ro/amp2/curs/8253.htm Access Date: 18-5-2021 at 6:10PM

[3] https://www.geeksforgeeks.org/interrupts-in-8086-microprocessor/ Access Date: 18-5-2021 at 7:10PM

[4] https://www.geeksforgeeks.org/8259-pic-microprocessor/ Access Date: 18-5-2021 at 8:30PM

[5] https://www.ecsdump.net/?tag=8259 Access Date: 18-5-2021 at 8:44PM

[7] Lab Manual.

6. Appendix

6.1 Appendix A:

```
CODE    SEGMENT PARA 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STAK

STAK    SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
STAK    ENDS

DATA    SEGMENT PARA 'DATA'
; 0 for lighting, 1 otherwise
DIGITS  DB 0C0h, 0F9h, 0A4h, 0B0h, 99h, 92h, 82h, 0F8h, 80h, 90h, 088h, 083h, 0C6h, 0A1h, 86h, 8Eh    , 00
              ; 0 to F
DATA    ENDS


START PROC
   MOV AX, DATA
   MOV DS, AX

CALL INSERT_ISR_INTO_VECTOR_TABLE
      CALL SETUP_8259
      ;; Address of 8253: 78h
      ;; Addrss of  8255: 70h
      ;; Address of 8259: 10h
      MOV AL, 80h                               ; 1000 0000b
      OUT 76h, AL                   ; Setup 8255 as all ports mo0 output
      CALL SETUP_8253

      STI
      XOR AX,AX
ENDLESS:
      CALL SHOW_AX

      CMP AX, 10h
      JNZ ENDLESS
      MOV AX, 0h            ; MAke AX 0 after reaching 16

      JMP ENDLESS
RET
START ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;                    Common Anode Seven Segmeent    ; 0 for lighting, 1 otherwise
SHOW_AX PROC NEAR
   PUSH AX

   XOR BX, BX
   MOV BL, AL
   MOV AL, DIGITS[BX]
   OUT 72h, AL                                 ; 8255's Port B at 72h
```

```
  POP AX
RET
SHOW_AX ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8253 PROC NEAR
        ;; SETUP 8253

        ; --> Used to Divide 4Mhz Clock to 2000 so that we get 2Khz Output
     MOV AL, 00110111b          ; 00 [counter 0], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL             ; configure the counter 0 of 8253

        ; --> Pulse Wave  :: MOD 2 of 8253
        MOV AL, 01110111b       ; 01 [counter 1], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL             ; configure the counter 1 of 8253

        MOV AL, 00h
        OUT 78h, AL             ; Send first byte of Counter 0
        MOV AL, 20h
        OUT 78h, AL             ; Send second byte of Counter 0
        ;;--> Counter0 : 2000h  ; send BCD 2000 to counter 0

        MOV AL, 00h
        OUT 7Ah, AL             ; Send first byte of Counter 1
        MOV AL, 5h
        OUT 7Ah, AL             ; Send second byte of Counter 1
        ;;--> Counter1   : 5h ; send BCD 5 to counter 1
RET
SETUP_8253 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8259 PROC NEAR

MOV AL, 00010011b               ; ICW1 : Edge Triggered - 1 Master - ICW4 Needed
OUT 10h, AL                     ; SET InputControlWord1

MOV AL, 40h                     ; ICW2 : 40h
OUT 12h, AL                     ; SET InputControlWord2

MOV AL, 03h                     ; AEOI(Automatic Interrupt) = 1, 8086= 1
OUT 12h, AL                     ; SET InputControlWord4

RET
SETUP_8259 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INSERT_ISR_INTO_VECTOR_TABLE PROC NEAR
  XOR AX, AX
  MOV ES, AX
  MOV AL, 40H
```

```
  MOV AH, 4
  MUL AH
  MOV BX, AX
  LEA AX, INC_AX_PER_15_SEC
  MOV WORD PTR ES:[BX], AX
  MOV AX, CS
  MOV WORD PTR ES:[BX+2], AX
RET
INSERT_ISR_INTO_VECTOR_TABLE ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Interrupt Routine
INC_AX_PER_15_SEC PROC FAR
  INC AX
IRET
INC_AX_PER_15_SEC ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

CODE    ENDS
      END START
```

6.2 Appendix B:

```
CODE    SEGMENT PARA 'CODE'
        ASSUME CS:CODE, DS:DATA, SS:STAK

STAK    SEGMENT PARA STACK 'STACK'
        DW 20 DUP(?)
STAK    ENDS

DATA    SEGMENT PARA 'DATA'
; 0 for lighting, 1 otherwise
DIGITS  DB 0C0h, 0F9h, 0A4h, 0B0h, 99h, 92h, 82h, 0F8h, 80h, 90h, 088h, 083h, 0C6h, 0A1h, 86h,
8Eh       , 00      ; 0 to F
DATA    ENDS



START PROC
  MOV AX, DATA
  MOV DS, AX

CALL INSERT_ISR_INTO_VECTOR_TABLE
    CALL SETUP_8259
    ;; Address of 8253: 78h
    ;; Addrss of  8255: 70h
    ;; Address of 8259: 10h
    MOV AL, 89h                     ; 1000 1001b
    OUT 76h, AL                     ; Setup 8255 as all ports mo0 output
    CALL SETUP_8253

    STI
    XOR AX,AX
ENDLESS:
    CALL SHOW_AX

    CMP AX, 10h
    JNZ ENDLESS
    MOV AX, 0h              ; MAke AX 0 after reaching 16

    JMP ENDLESS
RET
START ENDP

;;;;;;;;;;;;;;;;;;;;;;;;;;;;              Common Anode Seven Segmeent         ; 0 for lighting, 1 otherwise
SHOW_AX PROC NEAR
  PUSH AX

  XOR BX, BX
```

```
  MOV BL, AL
  MOV AL, DIGITS[BX]
  OUT 72h, AL                    ; 8255's Port B at 72h

  POP AX
RET
SHOW_AX ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8253 PROC NEAR
        ;; SETUP 8253

        ; --> Used to Divide 4Mhz Clock to 2000 so that we get 2Khz Output
      MOV AL, 00110111b       ; 00 [counter 0], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL           ; configure the counter 0 of 8253

        ; --> Pulse Wave  :: MOD 2 of 8253
        MOV AL, 01110111b     ; 01 [counter 1], 11[ 2 byte data], 010 [ with mod 3], 1 [BCD]
        OUT 7Eh, AL           ; configure the counter 1 of 8253

        MOV AL, 00h
        OUT 78h, AL           ; Send first byte of Counter 0
        MOV AL, 20h
        OUT 78h, AL           ; Send second byte of Counter 0
        ;;--> Counter0 : 2000h   ; send BCD 2000 to counter 0

        MOV AL, 00h
        OUT 7Ah, AL           ; Send first byte of Counter 1
        MOV AL, 5h
        OUT 7Ah, AL           ; Send second byte of Counter 1
        ;;--> Counter1   : 5h ; send BCD 5 to counter 1
RET
SETUP_8253 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
SETUP_8259 PROC NEAR

MOV AL, 00010011b             ; ICW1 : Edge Triggered - 1 Master - ICW4 Needed
OUT 10h, AL                   ; SET InputControlWord1

MOV AL, 40h                   ; ICW2 : 40h
OUT 12h, AL                   ; SET InputControlWord2

MOV AL, 03h                   ; AEOI(Automatic Interrupt) = 1, 8086= 1
OUT 12h, AL                   ; SET InputControlWord4
```

```
MOV AL,0EEh
OUT 12H,AL

RET
SETUP_8259 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INSERT_ISR_INTO_VECTOR_TABLE PROC NEAR
  XOR AX, AX
  MOV ES, AX
  MOV AL, 44H
  MOV AH, 4
  MUL AH
  MOV BX, AX
  LEA AX, INT_4
  MOV WORD PTR ES:[BX], AX
  MOV AX, CS
  MOV WORD PTR ES:[BX+2], AX

  XOR AX, AX
MOV ES, AX
MOV AL, 40H
MOV AH, 4
MUL AH
MOV BX, AX
LEA AX, INC_AX_PER_15_SEC
MOV WORD PTR ES:[BX], AX
MOV AX, CS
MOV WORD PTR ES:[BX+2], AX

RET
INSERT_ISR_INTO_VECTOR_TABLE ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;; Interrupt Routine
INC_AX_PER_15_SEC PROC FAR
  INC AX
IRET
INC_AX_PER_15_SEC ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
INT_4 PROC FAR
IN AL,74H
IRET
INT_4 ENDP
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

```
CODE    ENDS
        END START
```