**汇编与接口（英）试题**

# Final Examination of Assembly Programming and Interfacing (2022 Spr. )

There are altogether 3 projects in the exam. And all of them should be completed within 2 weeks.

Mail your solution to the email address notified in class, and text the teacher about it. So that to make sure your solution is received.

Your solution will be verified within EMU8086 software, or Proteus software. If the program works, 100% score will be assigned. If it fails, the source code will be reviewed, and a reasonable score will be assigned.

## Project1 Average Calculation (30 score)

Please write a program to calculate the average score of the following 3 score groups:

1. The scores is saved in an array that can be found in the Data segment.
2. The first item in the score array is the counts of score members in that array, and the last item in the array should be used to save the average score.
3. A template that includes data segment definition is provided. Build your program with that template or create your own.

;========================================================

;Description: Program of Project 1 calculating the average score.

; It is one of the assignment of the offline examination of

; assembly and interfacing class in spring semester 2022.

;Author:

;Student ID:

;Date:27th May 2022

;========================================================;===================

.MODEL SMALL

.STACK 64

.DATA

SCORETABLE1 DB 5, 90, 92, 96, 82, 94, 00

SCORETABLE2 DB 8, 70, 66, 83, 90, 92, 96, 82, 88, 00

SCORETABLE3 DB 6, 55, 90, 88, 73, 62, 92, 00

.CODE

MAIN PROC FAR ;this is the program entry point

MOV AX, @DATA ;load the data segment address

MOV DS, AX ;assign value to data segment register

;======================================================

; put your codes here

;=========================================================

MOV AH, 4CH ;set up to

INT 21H ;return to DOS

MAIN ENDP

END MAIN ;this is the program exit point

MAIN ENDP

END MAIN

## Project2 Basic IO Operation (30 Score)

Please try to write a program to acquire the states of a group of spin button switches, then use it to control the on/off state of a group led lights. The hard schematic design can be found in Figure 1 below.

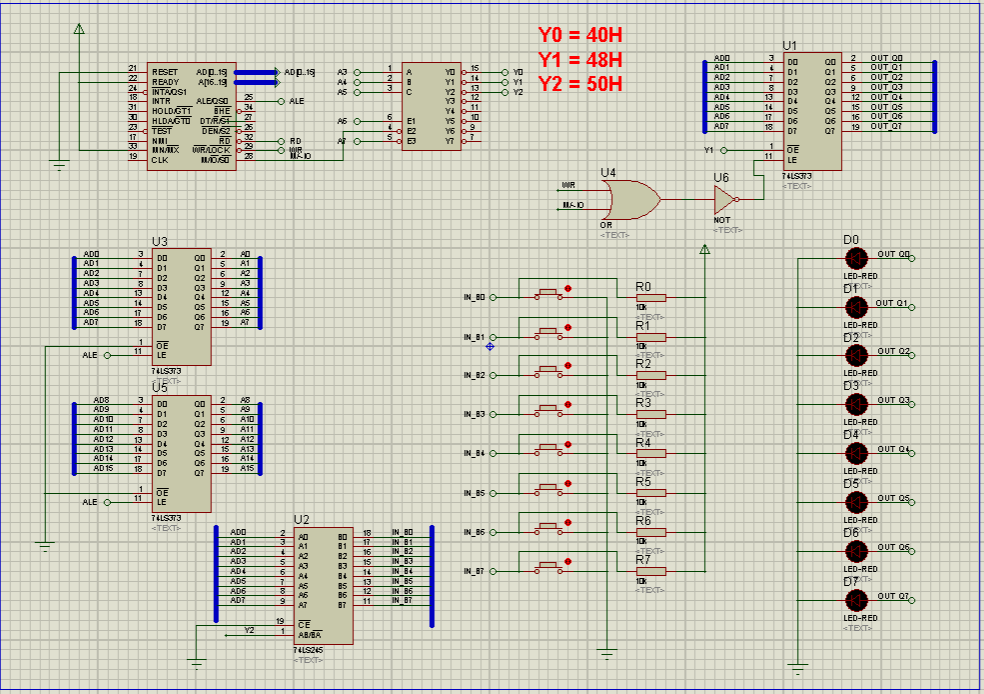


Figure 1. The full schematic of project02 direct IO

1. The spin button group is connected to the system data bus through a tri-state gate unit, as shown in Figure 2 below.

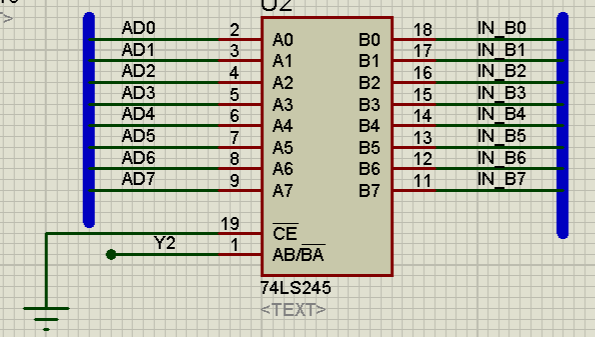


Figure2. The output port circuit

1. The led lights group is connected to the system data bus through a latched buffer unit, as shown in Figure3 below.

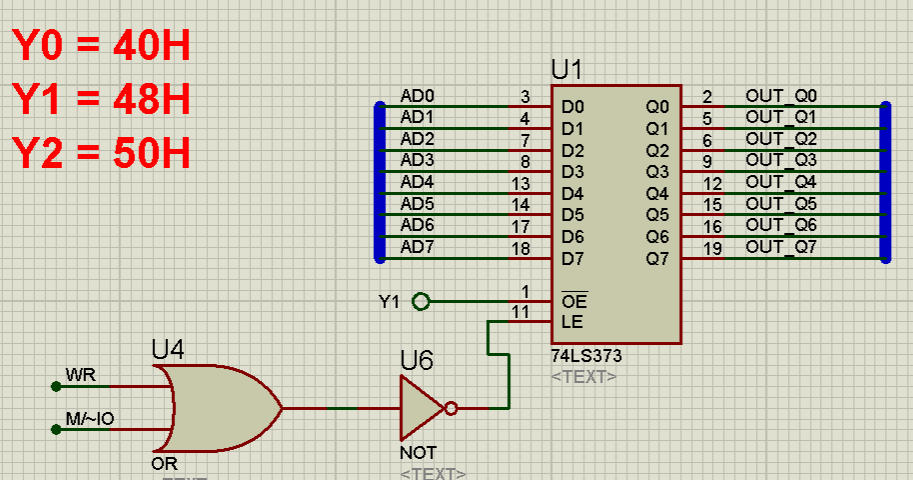


Figure3. The input port circuit

1. When your program is executed, each time we press the spin button , it will affect the on/off state of the led lights immediately.
2. A demo video can be found in the hardware schematic files director. And template .asm file is provided.

;========================================================

;Description: Program of Project 2 Direct I/O operation.

;Author:

;Student ID:

;Date:1st June 2022

;========================================================;===================

.MODEL SMALL

.STACK 32

.DATA

.CODE

MAIN PROC FAR

MOV AX, @DATA

MOV DS, AX

TODO1: ;TODO1: acquire switches

;TODO2: out put to PORT\_OUT

;TODO3: call the DELAY subprocedure before loop back to TODO1

JMP TODO1

;TODO3: quit to DOS

MOV AX, 4C00H

INT 21H

MAIN ENDP

;;==============================================================

;Subprocedure: DELAY

;delay for some millseconds

DELAY PROC NEAR

;codes to do delay

RET

DELAY ENDP

END MAIN

## Project3 Lantern Control (40 Score)

Please write a program to drive the lantern in project 03. The hardware schematic design can be found in Figure 4 below.

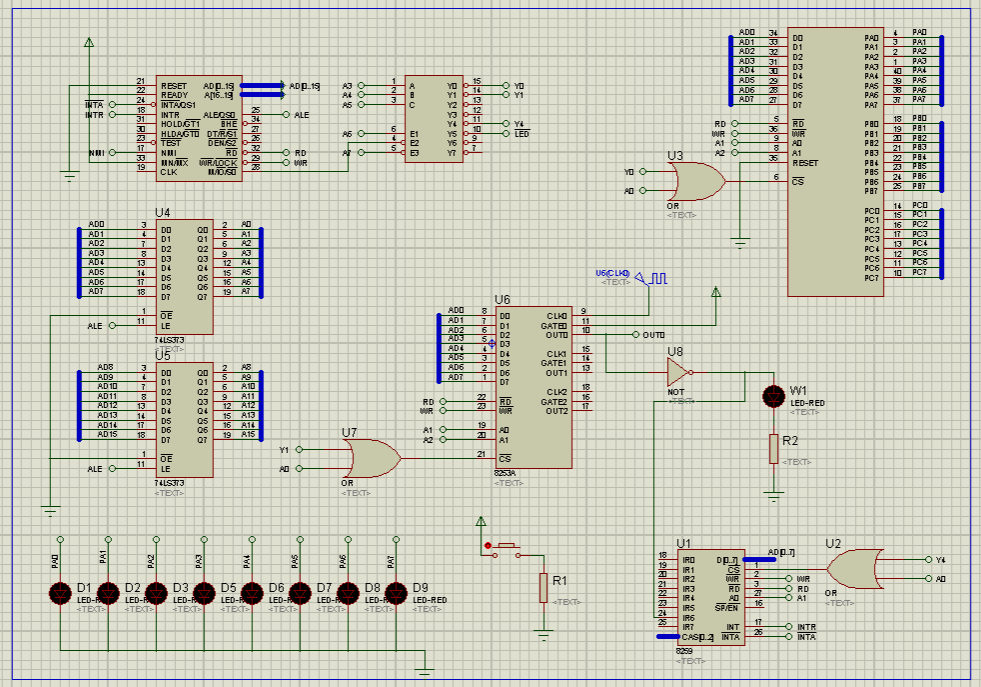


Figure 4. The full schematic of project03 Lantern Control

1. The lantern (8 led lights) is connected to the system data bus through the PortA of a piece of 8255, as shown in figure5 below

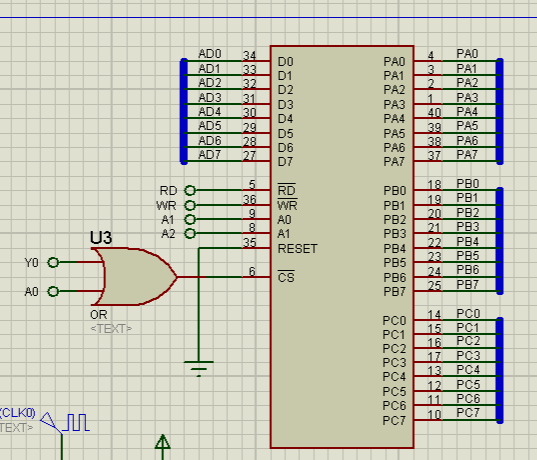


Figure 5. The lantern control port: 8255.PortA

1. A piece of 8253A programmable interval timer is used to generate periodical timing signal with its Timer0. The output of Timer0 is connected to the Interrupt Request pin IR6 of a piece of 8259 interrupt controller. The circuit can be found in figure 6 below.

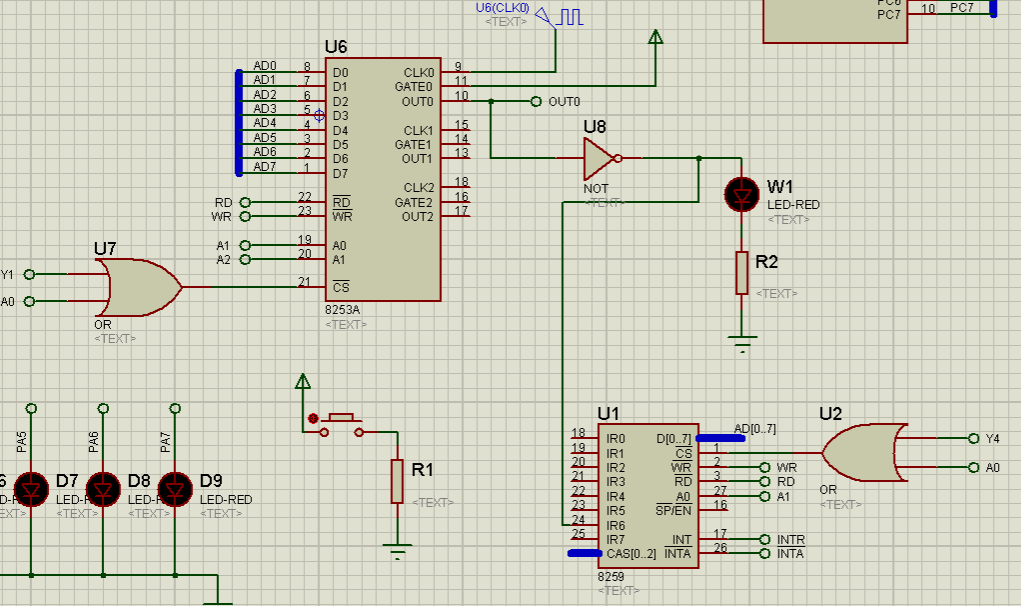


Figure 6. The Timing and Interrupt handling circuit

1. Each time the periodic timing output signal comes, a interrupt request is send to the 8086 processor. Within the Interrupt Service Routine, a pattern code should be send to the lantern to control its on/off states.
2. When your program is executed, the on-state light in the lantern moves, each time 8253.Timer0 timing is up.
3. A demo video can be found in the hardware schematic files director. And template .asm file is provided.

;========================================================

;Description: Program of Project 3 lantern control circuit.

;Author:

;Student ID:

;Date:3ed June 2022

;========================================================

.MODEL SMALL

.STACK 64

.DATA

; pattern code and control flags

pattern\_code DB 01H

.CODE

MAIN PROC FAR

MOV AX, @DATA

MOV DS, AX

CLI

;TODO1: regist \_ISR\_Timer

;TODO2: initiALize 8259, use IR6 as interrupt source input

;8259 mode: edge triggering, single piece, ICW4 needed;

; Interrupt Number of IR0 is 40H;

; Normal EOI, none buffered mode.

;TODO3: initiALize 8255: PortA mode0, output

;TODO4: OUTput the pattern code to the lantern

; and wait for the 1st interrupt request

;TODO5: initialize 8254: Timer0, mode2, initialize value = 20

;TODO6: open interrup handling, and output interrupt number of IR6

; to port 60H in the follwing FIX\_BUG section,

; in order to fix the bug within 8259 element.

STI

FIX\_BUG:

MOV DX,60H

MOV AL,(ICW2+6) ;dump 40H, which is the int number, to data bus

OUT DX,AL

JMP FIX\_BUG

;TODO7: quit to DOS

MOV AX, 4C00H

INT 21H

MAIN ENDP

;;==============================================================

;SubrOUTine: \_ISR\_Timer

\_ISR\_Timer PROC FAR

CLI

;===========================================

;put your codes here todo:

; load pattern\_code, and change it then output if through 8259.PortA

;===========================================

STI

IRET

\_ISR\_Timer ENDP

END MAIN ;this is the program exit point