|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | | | | |
| AIUB | **American International University- Bangladesh (AIUB)**  **Faculty of Engineering** | | | |
| **Course Name:** | Microprocessor and Embedded System | **Course Code:** | EEE 4103 | |
| **Semester:** | Fall 2022-23 | **Term:** | Mid | |
| **Total Marks:** | 20 | **Submission Date:** | 28-10-2022 | |
| **Faculty Name:** | Prof. Dr. Engr. Muhibul Haque Bhuyan | **Assignment #:** | | 02/02 |

Course Outcome Mapping with Questions

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Item** | **COs** | **POIs** | **K** | **P** | **A** | **Marks** | **Obtained Marks** |
| **Q1** | **CO2** | **P.a.3.C3** | **K3** | **P1, P2, P6** |  | **8** |  |
| **Q2** | **CO3** | **P.c.1.C4** | **K5** | **P1, P2, P7** |  | **3** |  |
| **Q3** | **CO2** | **P.c.1.C4** | **K5** | **P1, P2, P7** |  | **3** |  |
| **Q4** | **CO2** | **P.c.1.C4** | **K5** | **P1, P2, P7** |  | **3** |  |
| **Q5** | **CO1** | **P.a.4.C.3** | **K4** | **P1, P3, P7** |  | **3** |  |
| **Total:** | | | | | | **20** |  |

**Student Information:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name:** |  | **Section:** |  |
| **Student ID #:** |  | **Department:** |  |

**Marking Rubrics (to be filled by Faculty):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Problem #** | **Excellent**  **[5]** | **Proficient**  **[4]** | **Good**  **[3]** | **Acceptable**  **[2]** | **Unacceptable**  **[1]** | **No Response**  **[0]** | **Secured Marks** |
| Detailed unique response explaining the concept properly and answer is correct with all works clearly shown. | Response with no apparent errors and the answer is correct, but explanation is not adequate/unique. | Response shows understanding of the problem, but the final answer may not be correct | Partial problem is solved; response indicates part of the problem was not understood clearly. | Unable to clarify the understanding of the problem and method of the problem solving was not correct | No Response/ (Copied/identical submissions will be graded as 0 for all parties concerned) |
| **1** |  |  |  |  |  |  |  |
| **2** |  |  |  |  |  |  |  |
| **3** |  |  |  |  |  |  |  |
| **4** |  |  |  |  |  |  |  |
| **5** |  |  |  |  |  |  |  |
| **Comments** |  | | | | | **Total marks (20)** |  |

1. (a) Assume that a microprocessor has only two registers, R1 and R2, and that only the following instruction is available:

XOR Ri, Rj; Rj <- Ri ⊕ Rj ; i, j = 1, 2

Using this XOR instruction, find an instruction sequence to exchange the contents of registers R1 and R2.

Answer:

(a) Answer:

We need to exchange the data in register R1 and R2

that means:

R1  R2

R2  R1

MOV A, R1  Solving R2 value into A

XOR R1, R2  Cleaning (R1) register i.e., R1 = ow

XOR R1, R2  OH XOR with R2 = SH then R1 = SH

XOR R2, R2  Clarifying R2 i.e., R2 =OH

XOR R2, A  OH XOR with A contest i.e., 7H, R2 =7H

Example: R1 = 7H then output R1 = 5H

R2 = 5H R2 = 7H

A is the accumulator which is a general-purpose register. This register present in every microprocessor, microcontroller.

(b) Assume two 2’s-complement signed numbers are stored as M = 1111111lb and N = 11111100b. Write down the assembly code to perform the following signed multiplication: M = 5\*M - 2\*N

What would be the contents of the register, M?

Answer:

M DW 255

Q DW 252

MOV AX,5

IMUL M

MOV CX,M

MOV AX,2

IMUL Q

SUB CX,Q

(c) Find the logic operation for an 8-bit data to clear bits 2 and 4 of an 8-bit number, 7EH to 0’s without changing the other bits.

Answer:

Input data = (7E)16 = (0111 1110)2

Now we need to clear 2nd and 4th bit of the number.

Now the rightmost bit is bit 0 and leftmost bit is 7th bit. To clear 2nd, 4th take a number of 8 bits with all other bits 1 except 2nd and 4th (1110 1011)

Now we take product of this number with input data

So

* (0111 1011)2 and (1110 1011)2
* 0110 1010
* (6A)16

(d) Find the logic operation for an 8-bit data to set bits 0 and 7 of an 8-bit number, 3AH to 1’s without changing the other bits.

Answer:

Bitwise OR operation, this refers to the operation in which each bits of the operands are ORed. In case of OR operation, the output will be 1 if any of the bits is 1. So, this operation can be used to set bits. The Symbol for bitwise OR operation is |. The number is, 0x3A = 0011 1010

Here, bits 0 and 7 to be set without changing other bits. So the bits 7 and 0 must be 1 and remaining bits must be 0.

So, 8-bit data must be, 1000 0001 = 0x81

So operation is 0x3A | 0x81 = 0011 1010 | 1000 0001 = 1011 1011

(f) Using only MOV, ADD, SUB, INC, DEC, and NEG, translate the following high-level language assignment statements into assembly language. A, B, and C are word variables.

a. A = B - A

b. A = -(A + 1)

c. C = A + B

d. B = 3×B + 7

e. A = B – A – 1

Answer:

1. A=B-A, in Assembly language.

MOV AX, A

MOV BX, B

SUB BX, AX

MOV AX, BX

1. A= -(A+1), in assembly language

MOV AX, A

ADD AX, 1

NEG AX

MOV A, AX

1. C= A+B, in assembly language

MOV AX, A

MOV BX, B

ADD AX, BX

MOV C, AX

1. B= 3B+7, in assembly language
2. MOV Ax, 3
3. IMUL B
4. ADD Ax, 7
5. MOV B, Ax
6. A=B-A-1, in assembly language

MOV Bx, B

SUB Bx, A

SUB Bx, 1

MOV A, Bx

(g) Divide 4005h by 150 h, using 16-bit operands, and then show the concerned register contents in binary format. Run it on 8086 Emulator and show the simulation results and codes.

Answer:

The assembly code is:

MOV AX, 4005 H

MOV BX, 0150 H

DIV BX

So,

Ax is quotient is 0030H and in binary = 0000 0000 0011 0000b

= [48d] and

DX is reminder = 0105H and in binary = 0000 0001 0000 0101b = [261d]

Graphical user interface, application

Description automatically generated

(h) Fill in the blanks with appropriate mnemonics, data, registers, processor directives, etc. for an 8086 emulator to form the following Ladder (Arithmetic/Logic) Operation:

CX = 0AH \* 03H + 003H - 1BH / 02H [Consider, AX = 0AH, DX = 03H]

ORG 100H

XOR AX,

MOV AX,

MOV , 03H

MUL DX

CX, AX

CX, 0003H

MOV AX,

MOV DX,

DIV

SUB CX,

RET

Draw the flowchart of the completed program. Run it on 8086 Emulator and show the simulation results and codes.

Answer:

ORG 100H

XOR AX,AX

MOV AX,0AH

MOV DX,03H

MUL DX

MOV CX, AX

ADD CX, 0003H

MOV AX, 1BH

MOV DX, 02H

DIV DX

SUB CX, AX

RET

Graphical user interface, application

Description automatically generated

(i) Draw a flowchart to execute a function (A+B)^2 – 4\*A\*B, consider A = 3, B = 2. Write down the assembly language program and run it on 8086 Emulator. Show the simulation results and codes.

1. In a biscuit factory, a control system has been designed based on an Arduino Uno microcontroller. This system transfers the biscuit packets at a regular time interval to the conveyer belt from the biscuit-making machine. It was found that the Timer modules in the Arduino microcontroller are perfect candidates to count the number of biscuit packets being transferred to the conveyer belt correctly and appropriately. Compute the number of biscuit packets being transferred using the Timers within the following time intervals (Timer0 or Timer1 may be used based on need).
2. 100 ms
3. 5000 ms

Assume that the Arduino system clock has been set at 32 MHz and the available pre-scalers of the system are 1, 8, 64, 256, 512, and 1024. Comment on why and how the required counts are possible to implement using the Timer0 or Timer1. Write down the program codes for the Arduino board.

Answer:

a)

#define PIN\_USED 8

int milisec = 100;

int prescalar = 1024;

int clock\_frequency = 32000000;

float clock\_period = 1/(float)clock\_frequency;

int count = ((milisec\*.001/clock\_period)-1)

void setup ()

{

pinMode (PIN \_USED, OUTPUT)

TCCR1A = 0b00000000;

TCCR1B = 0b00000101;

TCNT1 = 0;

}

Void loop ()

{

If (TCNT1 >= count)

{

TCNT1 = 0;

digitalWrite (PIN\_USED, !digitalRead (PIN\_USED));

}

}

b)

#define PIN\_USED 8

int milisec = 5000;

int prescalar = 1024;

int clock\_frequency = 32000000;

float clock\_period = 1/(float)clock\_frequency;

int count = ((milisec\*.001/clock\_period)-1)

void setup ()

{

pinMode (PIN \_USED, OUTPUT)

TCCR1A = 0b00000000;

TCCR1B = 0b00000101;

TCNT1 = 0;

}

Void loop ()

{

If (TCNT1 >= count)

{

TCNT1 = 0;

digitalWrite (PIN\_USED, !digitalRead (PIN\_USED));

}

}

1. (a) A TV remote controller is to be designed for the selection of channels. The channel selection buttons are required to be debounced, that is, one press must cause the remote to change to the subsequent channel. It was observed that the switches exhibit bounce times well under 5 ms and the duty cycle is 50%. Design a circuit using the 74HC14 Schmitt trigger IC along with the resistance and capacitance. The worst-case *Vth* of 74HC14 for a signal going low is 2.1 V and that of when going high is 1.2 V. Also, consider that the CMOS device leakage current is 10 A and the gate’s best-case switching point is of the order of 0.5 V. What is the hysteresis voltage?

Answer:

Answer:

The worst-case Vth of 74HC14 for a signal going low is 2.1 V

5 ms and the duty cycle is 50%.

Rearranging the capacitor discharging formula to solve for R (the cost and size of capacitors vary widely so it is the best option to select a value for C and then compute R) yields:

The worst-case Vth of 74HC14 for a signal going low is 2.1 V. 0.1 μF capacitor, since this is small, cheap, and solve for the conditions in which the switch is closed. The capacitor discharges through R2 . If the power supply is 5 V (i.e., Vfinal = 5 V) then,

𝑅2 = = 57.63 kΩ. Since a resistor with this value is not available, so we use 47KΩ. But the analysis ignores the gate’s input leakage current. A CMOS device like the 74AHCT14 dribbles about 1 μA from the inputs. That 180 kΩ resistor will bias the input up to 0.047V ( 𝑉 = 𝐼𝑅 = = 0.047V), uncomfortably close to the gate’s best-case switching point of 0.5 V. Therefore, change the capacitor’s capacitance to 1 μF and use 4.7 kΩ for R

hysteresis voltage=0.047V

(b) Write a software code for the Arduino to read the status of a switch connected to pin number 7 by avoiding the bouncing problem of the switch and then display that switch status in the LED connected to pin number 10. If the switch status is HIGH/LOW the LED will be turned ON/OFF respectively. It was observed that the switches exhibit bounce times well under 10 ms and the duty cycle is 50%. Use the Timer0 or Timer1 function of the Arduino to compute this delay. Also, draw the flow chart of this program.

Answer:

For Timer 0

#define Led 10

#define switch 7

int milisec = 10;

int prescalar = 1024;

int Switch;

void setup ()

{

pinMode (Led, OUTPUT)

pinMode (switch, INPUT)

TCCR0A = 0b00000000;

TCCR0B = 0b00000101;

TCNT0 = 0;

}

Void loop ()

{

If (TCNT0>= count)

{

TCNT0 = 0;

Switch= digitalRead(switch);

If(Switch==1)

{

DigitalWrite(Led,HIGH);

}

else{

DigitalWrite(Led,Low);

}

}

}

For timer 1

#define Led 10

#define switch 7

int milisec = 10;

int prescalar = 1024;

int Switch;

void setup ()

{

pinMode (Led, OUTPUT)

pinMode (switch, INPUT)

TCCR1A = 0b00000000;

TCCR1B = 0b00000101;

TCNT1 = 0;

}

Void loop ()

{

If (TCNT1>= count)

{

TCNT1 = 0;

Switch= digitalRead(switch);

If(Switch==1)

{

DigitalWrite(Led,HIGH);

}

else{

DigitalWrite(Led,Low);

}

}

}

1. A biomedical engineer is to design a new laboratory where biologically hazardous materials are being used for experimental research works. As such, during the experimental works are ongoing, no one is allowed to remain inside the laboratory. To display the message that hazardous materials are being used in the laboratory, 2 lights with green and red colors are installed at the entrance door of the laboratory. When the green light is turned on, it indicates that anybody can enter the laboratory but when the red light is turned on then none can enter the laboratory. It is to be ensured that both lights can’t remain turned on at the same time and these 2 lights should be controlled using a single mechanical switch inside the laboratory and they can only light up alternatively when the switch is pressed or depressed. The switch has already been debounced using a hardware circuit. Draw a flowchart to show the logical flow of the program and then write a program for the Arduino Uno microcontroller to operate the door lights based on the switch conditions as per the given constraints.

Answer:

#define red 10

#define green 9

#define switch 7

Int red=1000;

int prescalar = 1024;

int Switch;

int delay\_timer (int milliseconds)

{

int count = 0;

while(1)

{

if(TCNT0 >= 16) // Checking if 1 millisecond has passed

{

TCNT0=0;

count++;

if (count == milliseconds) //checking if required milliseconds delay has passed

{

count=0;

break; // exits the loop

}

}

}

void setup ()

{

pinMode (green, OUTPUT)

pinMode (red, OUTPUT)

pinMode (switch, INPUT)

TCCR0A = 0b00000000;

TCCR0B = 0b00000101;

TCNT0 = 0;

}

Void loop ()

{

If (TCNT0>= count)

{

TCNT0 = 0;

Switch= digitalRead(switch);

DigitalWrite(green,HIGH);

DigitalWrite(green,Low);

If(Switch==1)

{

DigitalWrite(red,HIGH);

delay\_timer(red);

DigitalWrite(red,LOW);

DigitalWrite(green,HIGH);

}

else{

DigitalWrite(green,Low);

}

}

}

1. A convention hall manager wants to set up flashing lights at its entrance gate. In the upper row, there will be red color light, then green color light, then blue color light, then yellow color light, then orange color light, and finally in the bottom row there will be white color light. These lights will be implemented using different color LEDs. In each row, there would be five LEDs. The flashing of the lights will maintain the following sequence:
   1. The upper row with red lights will turn on first and remain turned on for 6 s, and then it would be turned off
   2. After that, the blue lights will turn on and remain turned on for 4 s, and then it would be turned off
   3. After that, the orange lights will turn on and remain turned on for 2 s, and then it would be turned off
   4. After that, the green lights will turn on and remain turned on for 6 s, and then it would be turned off
   5. After that, the yellow lights will turn on and remain turned on for 4 s, and then it would be turned off
   6. After that, the white lights will turn on and remain turned on for 2 s, and then it would be turned off
   7. After that, the lights of all rows will turn on and remain turned on for 5 s, and then all of them would be turned off for a duration of 2.5 s
   8. Then this cycle continues to repeat from step (a) to step (g) until a mechanical switch is pressed to zero. However, this switch status must be set to HIGH to continue the flashing sequence. Assume that the switch is debounced using a hardware circuit.

Answer:

Answer:(a-h)

This is a ATMega328p microcontroller based development board which consists of 14 digital pins and 6 analog pins to connect with the analog or digital devices.

Arduino Programming

The basic arduino program consists of two functions

void setup()

This function is used for the configuration of the modules used for the application. Digital pins , Serial communication modules are configured inside this function. This function will only be executed once.

Syntax-

void setup()

{

Configuration statements

}

void loop()

This function is used for executing the operations to be performed using the controller. As the name suggests, this function will execute till the power down or reset.

Syntax-

void loop()

{

Operations to be performed

}

pinMode() - This function is used to configure a digital pin as input or output according to the requirement.

Syntax -

pinMode(5,OUTPUT) - This statement configures digital pin 5 as output

pinMode(6,INPUT) - This statement configures digital pin 6 as input

digitalWrite() - This function is used to write either logic high (5 V) or logic low ( 0 V) to the digital pin which is configured as output

Syntax-

digitalWrite(5,HIGH ) - This statement write logic high value on pin 5

digitalWrite(5,LOW) - This statement write logic low value on pin 5

delay()

It generates the delay according to the argument in milli seconds.

Example: delay(100) - This will generted 100 ms delay

attachInterrupt()

This is the function used in Arduino for interrupt handling. In Arduino uno , digital pins 2 and 3 can be used for interrupts.

Syntax-

attachInterrupt(digitalPinToInterrupt(pin), ISR,mode)

Where, pin is the digital pin used for interrupt, and ISR is the set of actions when interrupt occurs.

Mode - This is used to define when the interrupt must be generated.

LOW - Interrupt triggered whenever pin is low

CHANGE - Interrupt triggered when the pin state changed

RISING - Interrupt triggered on low to high transition

FALLING - Interrupt triggered on high to low transition

The code will be,

void setup()

{

pinMode(13,OUTPUT); // LED pins configured as output

pinMode(12,OUTPUT);

pinMode(11,OUTPUT);

pinMode(10,OUTPUT);

pinMode(9,OUTPUT);

pinMode(8,OUTPUT);

pinMode(2,INPUT); // Button pins configured as input

attachInterrupt(digitalPinToInterrupt(2),button,FALLING);

}

void loop()

{

digitalWrite(13,HIGH); // Red LEDs ON for 6 seconds

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

delay(6000);

digitalWrite(13,LOW); // Blue LEDs ON for 4 seconds

digitalWrite(12,LOW);

digitalWrite(11,HIGH);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

delay(4000);

digitalWrite(13,LOW); // Orange LEDs ON for 2 seconds

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,HIGH);

digitalWrite(8,LOW);

delay(2000);

digitalWrite(13,LOW); // Green LEDs ON for 6 seconds

digitalWrite(12,HIGH);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

delay(6000);

digitalWrite(13,LOW); // Yellow LEDs ON for 4 seconds

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,HIGH);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

delay(4000);

digitalWrite(13,LOW); // White LEDs ON for 2 seconds

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,HIGH);

delay(2000);

digitalWrite(13,HIGH); // All LEDs ON for 5 seconds

digitalWrite(12,HIGH);

digitalWrite(11,HIGH);

digitalWrite(10,HIGH);

digitalWrite(9,HIGH);

digitalWrite(8,HIGH);

delay(5000);

digitalWrite(13,LOW); // All LEDs OFF for 2.5 seconds

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

delay(2500);

}

void button() // When button pressed

{

digitalWrite(13,LOW); // All LEDs off

digitalWrite(12,LOW);

digitalWrite(11,LOW);

digitalWrite(10,LOW);

digitalWrite(9,LOW);

digitalWrite(8,LOW);

while(digitalRead(2)==1); // Wait till released

}

RED LED On for 6 seconds