American International University – Bangladesh (AIUB) Faculty of Engineering

**Department of Electrical and Electronic Engineering**

|  |  |  |  |
| --- | --- | --- | --- |
| **Final Assignment** | | | |
| **Course Name:** | Microprocessor and Embedded Systems | **Course Code:** | EEE |
| **Semester:** | Spring 2023-2024 | **Section:** |  |
| **Faculty Name:** | **Engr. Md Shaoran Sayem** | | |
|  | | | |
| **Assignment No:** | 1F **(individual submission consisting of 30 marks)** | | |
|  | | | |
| **Submission Date:** |  | **Due Date:** | **02/05/2024** |

**Student Information:**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Student Name:** |  | | | | | | | | | | | | **Section:** |  |
| **Student ID #:** |  |  |  |  |  |  |  |  |  |  | **Assigned Date:** | **25.04.2024** | **Department:** |  |
| **p** | **q** | **-** | **a** | **b** | **c** | **d** | **e** | **-** | **r** |

# Special Instruction: Questions may be copied from here through copy-paste. Online submission via TEAMS is needed. However, hardcopy must also be submitted.

**Assessment Rubrics:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| COs-POIs | Excellent [19-20] | Proficient [15-18] | Good [11-14] | Acceptable [6-10] | Unacceptable [1-5] | No Response [0] | Secured Marks |
|  | All the problems are | All the problems are solved correctly. The results are generated by combining all possible input patterns with appropriate outcomes.  A few necessary drawings and computations are missing but no wrong  drawing. | All the problems are | All the problems are not solved correctly. The results are generated by combining several wrong or less no of input patterns with in/appropriate outcomes. Some necessary drawings and  computations are missing or wrong. | All the problems are not |  |  |
|  | solved correctly. The | solved correctly. The | solved correctly. The |  |
|  | results are generated | results are generated by | results are generated by | No |
|  | by combining all | combining all possible | combining mostly | responses |
| **CO3** | possible input patterns | input patterns with | wrong input patterns | at all or |
| **P.a.4.C.3** | with appropriate | appropriate outcomes. | with inappropriate | copied |
|  | outcomes. All | A few necessary | outcomes. Almost all | from |
|  | necessary drawings | drawings and | the necessary drawings | others |
|  | and computations are | computations are | and computations are |  |
|  | shown correctly. | missing or wrong. | missing or wrong. |  |
| **Comments** |  | | | | **Total Marks (20)** |  | |

1. Find the baud rate for the three operating modes when the oscillator frequency, *fOSC* = ac MHz (put side-by-side), and register data is, UBRRn = 010110101110. Calculate the baud error and comment on whether there will be any communication errors or not. Standard Baud rates are 300, 600, 1200, 2400, 4800, 9600, 14400,19200, 38400, 57600, 115200, 230400, ... bps.
2. Compute the duty cycle and sketch the waveform obtained at port D of the Arduino. Identify the modes of  
   operation and compute the operating frequency of that mode based on the following program segment.  
   Identify the Timer of the Arduino Microcontroller. The system clock frequency is rq MHz. Draw the relevant  
   circuit diagram using Proteus and show its timing diagram.  
   DDRD |= (1<<PD5);  
   pinMode(5, OUTPUT);  
   OCR0A = (200+ a + b + c); // Load a value in the OCR0A register  
   OCR0B= (100 + d + e); // Load a value in the OCR0B register  
   // Configure TCCR0A and TCCR0B registers for the mode and pre-scaler  
   TCCR0A |= (1 << COM0B1) | (1 << COM0A0) | (1<<WGM01) | (1<<WGM00);  
   TCCR0B |= (1<<WGM02) | (1<<CS01) | (1<<CS00);
3. Compute the duty cycle and sketch the waveform obtained at port D of the Arduino. Identify the modes of operation and compute the operating frequency of that mode based on the following program segment. Identify the Timer of the Arduino Microcontroller. The system clock frequency is pq MHz.

DDRD |= (1<<PD5);

pinMode(5, OUTPUT);

OCR0B= (150+a+b); // Load OCR0B for setting its duty cycle

// Configure TCCR0A and TCCR0B registers for the mode and pre-scaler

TCCR0A |= (1 << COM0B1) | (1<<WGM01) | (1<<WGM00);

TCCR0B |= (1<<CS02) | (1<<CS00);

1. Design an ***a***-bit shifter circuit for the listed shift functions provided in Table 1. Explain its operation for various cases of select inputs.

**Table 1: Functions of control variables**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Binary Code** | **Functions of selection variables** | | | | | |
| ***A*** | ***B*** | ***D*** | ***F* with *Cin* = 0** | ***F* with *Cin* = 1** | ***H*** |
| 0 0 0 | Input Data | Input Data | None | A-1 | A | 1’s to the output Bus |
| 0 0 1 | R1 | R1 | R1 | A+B | A+B+1 | Shift Left with *IL* = 0 |
| 0 1 0 | R2 | R2 | R2 | A-B-1 | A-B | No Shift |
| 0 1 1 | R3 | R3 | R3 | A | A+1 | Circulate Left with Carry |
| 1 0 0 | R4 | R4 | R4 | 𝐴̅ | X | 0’s to the output Bus |
| 1 0 1 | R5 | R5 | R5 | AX**OR** B | X | 0’s & 1’s to the lower and upper nibbles |
| 1 1 0 | R6 | R6 | R6 | A **AND** B | X | Circulate-Right with Carry |
| 1 1 1 | R7 | R7 | R7 | A **OR** B | X | Shift Right with *IR* = 0 |

1. Design a (*q* + *r*)-bit shifter for the four shifting operations listed in the following Table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Binary Code** | **The function of selection variables** | | | | | |
| **A** | **B** | **D** | **F with Cin = 0** | **F with Cin = 1** | **H** |
| **0 0** | Input Data | Input Data | None | A | A+1 | **Shift Left with IL=0** |
| **0 1** | R1 | R1 | R1 | A+B | A+B+1 | **Shift Right with IR=0** |
| **1 0** | R2 | R2 | R2 | A+B’ | A+B’+1 | **1’s to the output Bus** |
| **1 1** | R3 | R3 | R3 | A-1 | A | **0’s to the output Bus** |

1. Prepare a flow chart that will count the number of 1’s in a register, R4 and then store the counts in register R6.

Determine the outputs of the R6 (in binary) and R4 (in decimal) registers as well as of the carry flag (C) after each clock cycle or timing state.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Timing States | R4 | | | | | | | | C | R6 |
| 1 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| **T1** |  |  |  |  |  |  |  |  |  |  |
| **T2** |  |  |  |  |  |  |  |  |  |  |
| **T3** |  |  |  |  |  |  |  |  |  |  |
| **T4** |  |  |  |  |  |  |  |  |  |  |
| **T5** |  |  |  |  |  |  |  |  |  |  |
| **T6** |  |  |  |  |  |  |  |  |  |  |
| **T7** |  |  |  |  |  |  |  |  |  |  |
| **T8** |  |  |  |  |  |  |  |  |  |  |

1. Develop the control words in binary and hexadecimal formats using the information provided in Table 1 for the following micro-operations:

|  |  |
| --- | --- |
| i. Re←Ra+Rb | ii. Rd←3(Re – 0)/3 |
| iii. Rq←SHL Rp | iv. Output←Rc |
| v. Rd←Rc | vi. Rb←0 |
| vii. Rq←Input | viii. Rq←Rp-Ra |
| ix. Rr←SHR Rb | x. Rc←CRC Rd |

*\* If any value of a-e goes above 7 then it should be assumed as 7.*

The necessary bits for the control word are presented in Table 2.

**Table 2: 16-bit control word sequence**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| *A* | | | *B* | | | *D* | | | *F* | | | *Cin* | *H* | | |

One example is shown as follows:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Micro-operation | *A* | *B* | *D* | *F* | *Cin* | *H* | In Hex |
| R5 CRC (R3+R4) | 011 | 100 | 101 | 001 | 0 | 110 | 7296h |

1. Develop the control memory outputs for the sequence in Table 3 using the information listed in Table 1. To complete the memory outputs, use the microinstructions that you have developed in question no. 7.

**Table 3: Control memory bit sequence**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | ROM outputs | | | | | | | | | | | | | |
| ROM Address | | | Control Word | | | | | | | | | Address | | | Mux Select | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
| 0 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 0 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 0 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1 | 1 | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |