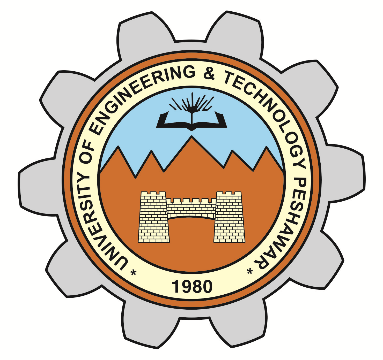
****

**Department of Computer Systems Engineering**

**University of Engineering & Technology**

# Peshawar

**Microprocessor Based System Design (MBSD)**

**6th Semester Mid-Term, Spring 2023**

**Max. Time:** 2 hours **Max. Points: 20**

**Instructions:**

1. Attempt ALL questions.

2. Exam is open book and open notes.

3. Cell phones and Laptops are strictly prohibited.

4. Exam is worth **20%** of the final grade. **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Q 1)**. **8-points**

**CLO-5/PLO-3 [Cognitive Domain: Synthesis]**

**Design** a system, where the Software in C will generate, a signal of

1. 0.25KHz with a duty cycle of 10% on P2.0 pin.
2. Whenever a user presses a button at (P3.2), the signal toggles to 0.5KHz with a duty cycle of 50%.
3. Again, pressing the same button will generate a signal of 1KHz with a duty cycle of 75%.

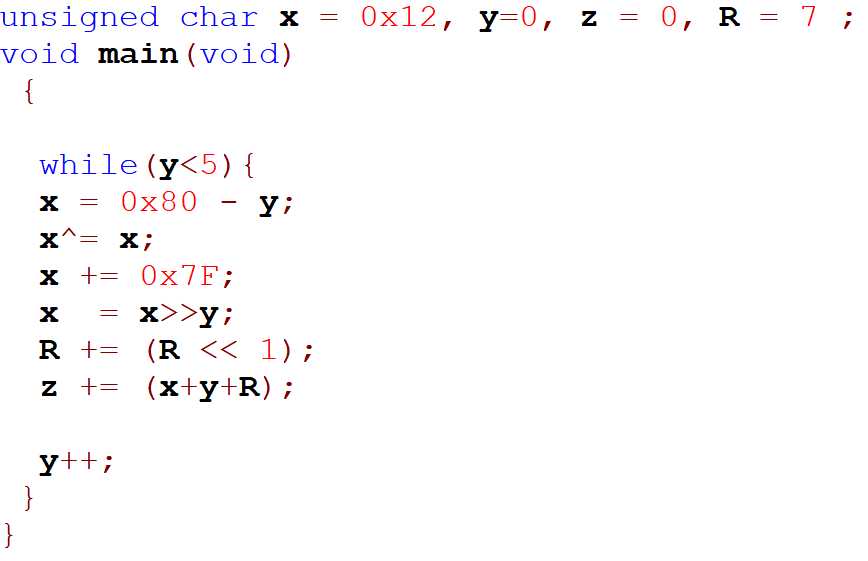
A third time button press will result in the generation of case A and so on.

* Draw the schematic diagram showing clearly the button circuit and oscilloscope.
* Draw the timing diagram with cursors clearly showing the time period with appropriate units.
* Assuming oscillator clock of 24MHz is used.
* Use **timer** interrupt. Use External interrupts.
* Polling is not allowed.

**Q 2)**. **8-points**

**CLO-3/PLO-2 [Cognitive Domain: Application]**

Fill the table by **solving** the values of variables in the program.

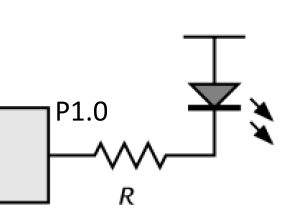


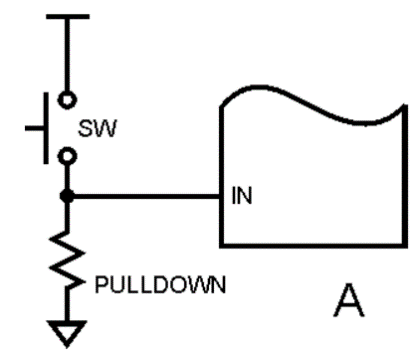
|  |  |  |  |
| --- | --- | --- | --- |
|  | **X** | **R** | **Z** |
| **Y = 0** | 18 (x12) | 7 (x07) | 0 (x00) |
| **Y = 1** | 127 (x7F) | 21 (x15) | 148 (x94) |
| **Y = 2** | 63 (x3F) | 63 (x3F) | 19 (x13) |
| **Y = 3** | 31 (x1F) | 189 (xBD) | 241 (xF1) |
| **Y = 4** | 15 (x0F) | 55 (x37) | 58 (x3A) |
| **Y=5** | 7 (x07) | 165 (xA5) | 234 (xEA) |

**Q 3)**. **4-points**

**CLO-2/PLO-1 [Cognitive Domain: Comprehension]**

**Translate** the following tasks into C code.

1. If we have an active-high button (A) at **P2.5** pin and an ***active-low*** *LED at P1.0* as shown below,



*P2.5 = \_1\_;*

Scan the button using polling

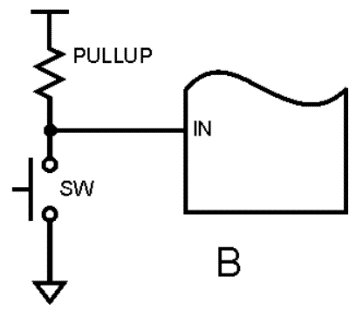
*While ((P2 & 0x20) == 1)*

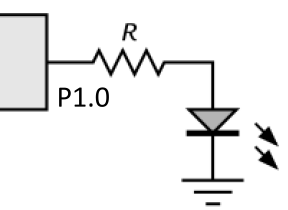
*{*

*P1.0 = \_\_0;\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ // Button pressed, TURN ON the* ***active-low*** *LED at P1.0*

*}*

1. If we have an active-low button (B) at **P2.5** and an **active-high** LED at P1.0 as shown below,





*P2.5 = 1;*

Scan the button, using polling

*While (\_\_\_(P2 & 0x20) == 0)*

*{*

*P1.0 = 1 ; // Button pressed, TURN ON the* ***active-high*** *LED at P1.0*

*}*