MID-TERM EXAM EVEN SEMESTER ACADEMIC YEAR 2022/2023
STUDY PROGRAM OF COMPUTER ENGINEERING
FACULTY OF ENGINEERING AND INFORMATICS
UNIVERSITAS MULTIMEDIA NUSANTARA

Subject	**:**	**CE432 Microprocessor System**	**Date**	**:**	
Lecturer(s)	**:**	**Dareen Kusuma Halim**	**Time**	**:**	**120 minutes**
Form	**:**	**Lab/Programming**	**Type**	**:**	**Onsite (**with Lab's PC / own laptop if necessary**)**

EXAM CONDITIONS / INSTRUCTIONS:

1. Submit only the ASM file for each question. The naming format is **name-studentid-q1.asm** and **name-studentid-q2.asm**
2. Provide explanation / comments in your program -> this is the differentiating factor between a good and bad score.
3. Please use EdSim51 with the default schematic.
4. You are allowed to write the comments in Bahasa or English.
5. DO NOT try to copy your course mate's answers, and DO NOT try to give answers to your course mate. The consequence applies for those **receiving** and **giving** answers.

COURSE SUB LEARNING OUTCOMES (SUB-CLO):

SUB LEARNING OUTCOMES (SUB-CLO)		ELO
Code	**Description**	
SUB-CLO 2	Students are able to design a simple embedded system by utilizing understanding of 8051 processor structure and language structure (C6)	ELO G
SUB-CLO 3	Students are able to design a 8051-based embedded system by utilizing JUMP, LOOP, and CALL instruction, according to the application (C6)	ELO G
SUB-CLO 4	Students are able to design a 8051-based embedded system by utilizing input/output ports (I/O) in a program, according to the application (C6)	ELO G
SUB-CLO 5	Students are able to design a 8051-based embedded system by utilizing various addressing modes in 8051 processor, according to the application (C6)	ELO G
SUB-CLO 6	Students are able to design a 8051-based embedded system by utilizing arithmetic operations in 8051 processor, according to the application (C6)	ELO G
SUB-CLO 7	Students are able to design a 8051-based embedded system by utilizing logic operations in 8051 processor, according to the application (C6)	ELO G

PROBLEM/QUESTIONS:

1. **Question 1: Sub-CLO 2 (Weight 16%), 3 (Weight 20%), 4 (Weight 16%), 7 (Weight 16%)**

For this program, you are using this setup.

Inputs:

- Switch at P2.0 as **Switch_1**, P2.1 as **Switch_2**.
- Switch at P2.7 as **Switch_start**.
- In EdSim51, if the switch is **closed (pressed)**, it gives a LOW (0) value. If the switch is **open (not pressed)**, it gives a HIGH (1) value.

Outputs:

- P1 connected to 8 LEDs.
- In EdSim51, to turn on an LED, you must provide a LOW (0) value. For example, `CLR P1.0` to turn on the right-most LED.

You are asked to design a running LED system. The system has three modes as follows:

- **Mode running right**. In this mode, the LED starts from the MSB (bit 7 of P1), runs through bit 6 to 0 → return to bit 7, runs through bit 6 to 0, and so on.

Time T0	■							
Time T1		■						
Time T2			■					
…				*Continue the sequence …*				
Time T7								■
Time T8	■							
Time T9		■						
…			*Continue the sequence …*					

- **Mode running left**. The opposite of mode running right. Starts from LSB (bit 0 of P1) to bit 7 and repeat the sequence.

The program flow is as follows:

- On system start (*initial state*), all LEDs are turned off. The system waits for user input.
- User first press either **Switch_1 (mode running right), Switch_2 (mode running left)**. For simplicity, we assume there is no user pressing more than on switch at a time (so you don't have to think "*what if Switch_1 and Switch_2 is pressed together*").
- User then press **Switch_start** to start the running / bouncing LED.
- The running LED will not stop unless user depress **Switch_start**. When stopped, the program goes back to the *initial state*.

N.B. for simplicity, you don't need to use a delay function. Just do a single NOP operation as the delay and set EdSim51's Update Frequency to 1.

ASSESSMENT RUBRIC (per question):

Rated aspect	**Assessment criteria**				
	Very poor	**Poor**	**Satisfactory**	**Good**	**Excellent**
	1-3	**4-6**	**7-9**	**10-13**	**14-16**
Sub-CLO 2 Processor & language	Depending on the correct use of register for certain operations and ASM language structure				
Sub-CLO 4 I/O	Depending on the correct use of input output, as well as input checking				
Sub-CLO 7 Logic ops	Depending on the use of logical operation for the running / bouncing LED				
	1-5	**6-9**	**10-13**	**14-17**	**18-20**
Sub-CLO 3 Branching	Depending on the program state and flow				

2. Question 2: Sub-CLO 5 (Weight 16%), 6 (Weight 16%)

For this program, you are using this setup.

Outputs:

- P1 connected to 8 LEDs.
- In EdSim51, to turn on an LED, you must provide a LOW (0) value. For example, `CLR P1.0` to turn on the right-most LED.

Design a program that adds the numbers in the last 5 digits of your student ID (split into three 2-digit decimal numbers). After the addition, format the resulting number as Binary-coded-decimal (BCD) and outputs it to the LED. The tens (*puluhan*) digit occupies P1.7 to P1.4, while the ones (*satuan*) digit occupies P1.3 to P1.0.

For example, assume the last 6 digits of a student ID is 032016.

- We have three numbers (**decimal**): 03, 20, 16.
- Adding the numbers result in 8051 results in a Hexadecimal number 0x27 or a Decimal number 39.
- Output the '3' as binary 0011 to P1.7-P1.4 and output the '9' as binary 1001 to P1.3-P1.0.
- Utilize 8051 BCD-related instructions for this.

The three numbers from your student ID **must be stored in address** #100H, #101H, and #102H (any order is fine). Use the appropriate addressing mode to access those values (*anything that is easiest for you, but still utilizes addressing*).

ASSESSMENT RUBRIC (per question):

Rated aspect	Assessment criteria				
	Very poor	Poor	Satisfactory	Good	Excellent
	1-3	4-6	7-9	10-13	14-16
Sub-CLO 5 Addressing modes	Depending on the correct use of addressing modes to store and access the numbers				
Sub-CLO 6 Math ops	Depending on the correct use of arithmetic operation for the BCD				

References:	Created by:	Approved by:
1. Lecture slides 1-7 2. Lab modules 1-7 3. Reference books	(Dareen K. Halim, S.Kom., M.Eng.Sc.) Course Coordinator	(Samuel, M.T.I.) Head of Study Program