BLG212E - SAMPLE FINAL EXAM
(2 hours)

QUESTIONS

QUESTION 1) [30 points]

a) [20 points] Write an Assembly program (use EDU-CPU instruction set) to calculate the **squares** of decimal numbers from 1 to 15, and save the squares to an **array**. Each element of array is 1 byte.

- Main program should perform a **loop**, and call the subroutine named **CALCULATE_SQUARE**, for each number.
- There are two parameters of the subroutine: Number (as input), and Square (as output)
- For parameter passing, either use the registers or memory variables.
- At the end of program, the array should contain the squares like follows : 1, 4, 9, 16, 25, 36, 49, . . . , 225.

b) [10 points] Write the **subroutine** CALCULATE_SQUARE. Do not use the **MUL** instruction for multiplication. Instead, calculate the square of a number by **looping** and adding. ($N^2 = N*N = N+N+N+...+N$)

QUESTION 2) [10 points]

Write the **addressing method** name of the source operand, for each of the Assembly instructions (EDU-CPU) listed on the right.

Also determine the invalid instructions, if there are any.

1. LDA A, 5
2. MOV A, B
3. LDA A, <$0020>
4. LDA A, <CD>
5. LDA C, <AB>
6. MOV A, CD
7. LDA A, <SK+4>
8. LDA C, <SK+CD+0>
9. LDA A, <SK+2>+1
10. LDA A, <YG+3>

QUESTION 3) [10 points] Consider a hypothetical microprocessor with the following features.
Data bus:16-bits, Address bus:16-bits, Number of assembly language instructions:250, Number of 16-bit registers:16.
Write the **Instruction Format** for the following hypothetical Assembly instruction : **ADD R1, R2, M**, where R1 and R2 are registers (sources), and M is memory address (target).

QUESTION 4) [50 points] A microprocessor-based system will be designed with the following components.

- CPU, PIA, A/D Converter (Analog to Digital), 8 LEDs, and a Potentiometer.
- PIA is connected to CPU; LEDs and A/D Converter are connected to PIA.
- A potentiometer is a variable resistor which can be used to change the voltage in a circuit. Its analog voltage reading pin is connected to the input of the A/D Converter.
- When user moves the handle of potentiometer, its analog voltage reading will increase or decrease (based on the moving direction of the handle) between 0 and 5 V.
- The output of the A/D Converter will be an 8-bit discrete digital value between 0 and 255.

POTENTIOMETER

a) [25 points] **Draw** the detailed design **diagram** of the system. Show all necessary connections between components.

b) [25 points] Write an Assembly program (use EDU-CPU instruction set) to do followings.

- By using the potentiometer, the user will control a series of 8 LEDs.
  Moving the potentiometer handle will turn ON or OFF more of the LEDs. (LED lightings will be leftmost-first.)
- Assume PIA registers are already defined as DIRECTION.A, DIRECTION.B, PORT.A, PORT.B.
- Perform conditionings of the PIA direction registers.
- By looping (endless), program should continuously read the discrete potentiometer values from PORT.A, then turn ON or OFF the respective number of LEDs in the PORT.B.
- Notice that there are 32 discrete values per one LED. (256 values / 8 LEDs = 32)
  (You should use this information when calculating the number of LEDs to be lighted.)

ANSWERS

ANSWER 1) [30 points]

a) [20 points]

```
SIZE EQU 15
SQUARES RMB SIZE

*Subroutine parameter variables:
NUMBER RMB 1
RETURNED_RESULT RMB 1

START
  LDA SK, SQUARES          ;Address of array
  LDA B, 1                 ;Loop counter initialization

Dongu
  STA B, NUMBER            ;Initialize the number
  BSR CALCULATE_SQUARE ;Call subroutine
  LDA A, <RETURNED_RESULT>
  STA A, <SK+0>    ;Save to array
  INC SK           ;Increment array index
  INC B            ;Increment loop counter
  CMP B, SIZE      ;Compare counter to limit
  BLT Dongu        ;Branch if less than
  INT
```

b) [10 points]

```
*Formula for N*N = N+N+N+...+N
CALCULATE_SQUARE
  LDA A, 0                 ;Result initialization
  LDA C, 1                 ;Loop counter initialization

Devam
  ADD A, <NUMBER>   ;Add number to A
  INC C             ;Increment counter
  CMP C, <NUMBER>   ;Compare counter to limit
  BLT Devam         ;Branch if less than

  STA A, RETURNED_RESULT  ;Store A to Result
  RTS
```

ANSWER 2) [10 points]

1. Immediate
2. Implied (Register)
3. Direct
4. Indirect
5. INVALID instruction
6. INVALID instruction
7. Indexed
8. Register Relative Indexed
9. Incremented Indexed
10. Stack Pointer Relative

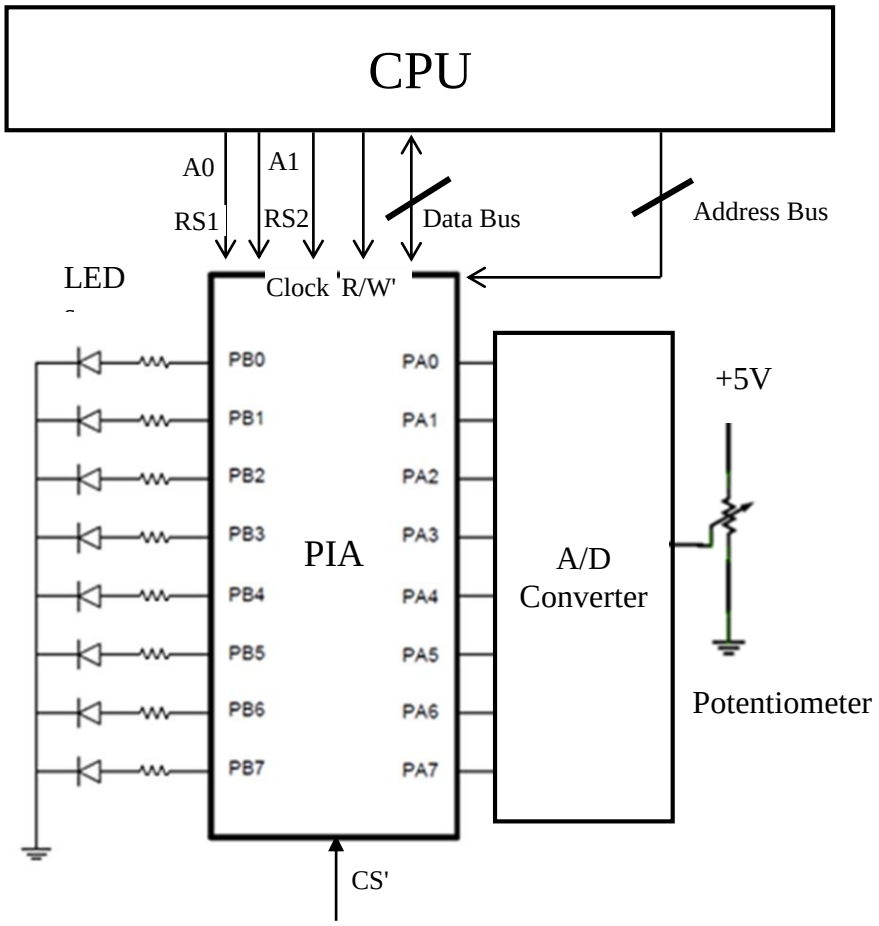
ANSWER 3) [10 points]

Multiple Words, 1-Address Instruction Format (32-bit)

1. Octal	2. Octal		3. Octal	4. Octal
Opcode 8 bits	Register1 4 bits	Register2 4 bits	Upper half of memory address 8 bits	Lower half of memory address 8 bits

ANSWER 4) [50 points]

a) [25 points]



b) [25 points]

```
NUM_OF_LEDS RMB 1      ;Number of LEDs to be lighted
START
* Conditioning of PIA ports
  STA  $00,  <YÖNLEN.A>      ; All bits of Port-A are input (from A/D Converter)
  STA  $FF,  <YÖNLEN.B>      ; All bits of Port-B are output (LEDs)

MAIN_LOOP
*Endless loop
  LDA AB, 0                  ;B will be used to store potentiometer value
  LDA B, <İSKELE.A>          ;Read potentiometer value (0-255)
  DIV AB, 32                 ;Calculate the number of LEDs to be lighted
  STA B,  NUM_OF_LEDS        ;Store the result to the memory variable
  SET E                      ;Set Carry flag to 1 (for the ROR operation later)
  LDA B, 0                   ;Inner loop counter (From 1 to NUM_OF_LEDS)
  LDA A, %10000000           ;Initially only leftmost LED bit is 1

DISPLAY_LOOP
  STA A, İSKELE.B            ;Display the LEDs
  ROR A                      ;Rotate Right A   (so that another 1 is shifted to A, from Carry bit)
  INC B                      ;Increment inner loop counter
  CMP B, <NUM_OF_LEDS> ;Compare to inner loop limit
  BLT DISPLAY_LOOP           ;Goto inner loop

  BRA MAIN_LOOP              ;Goto outer loop
```