BLG 212E - SAMPLE FINAL EXAM QUESTIONS

Books, notes, electronic devices, etc are closed. Exam duration is 2 hours.

Q1) [20 points] Write an Assembly program to find result of XOR operations performed on the following number series.
%00000001 xor %00000011 xor %00000111 xor %00001111 XOR . . . . xor %11111111

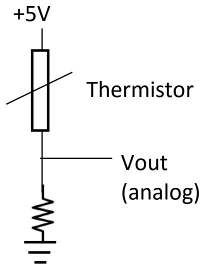
- Each number has a length of 1 byte.
- Order of XOR operations (left-to-right, or right-to-left) does not change the final result.
- Find the final result and store it into a variable named RESULT (1 byte).

Q2) [30 points] Write an Assembly program to do followings.

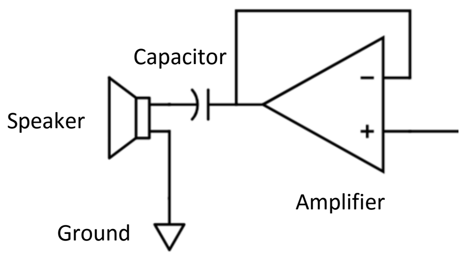
- Define a constant named SIZE, which is equal to 4.
- Define a variable named MATRIX, whose length is SIZE*SIZE. Each element is 1 byte.
- Initialize the MATRIX with the following data (row-wise): 6, 1, 8, 0, 5, 9, 4, 3, 7, 1, 5, 2, 2, 0, 8, 7
- By looping, calculate sum of first column and store result into a variable named SUM_COL1 (1 byte).
  The expected sum is (decimal) : 6 + 5 + 7 + 2 = 20

Q3) [50 points] A microprocessor-based system will be designed with the components below.

- CPU, PIA, Analog to Digital Converter, Blue LED, Red LED, Thermistor, Sound Component.
- PIA (Peripheral Interface Adapter) is connected to CPU.
- Two LEDs, A/D Converter, Sound Component are all connected to PIA.
- Thermistor (thermal resistor used as temperature sensor), is connected to A/D Converter input.
- Output of A/D Converter is 8-bit discrete digital value between 0 and 255.



Sound Component



a) [30 points] Draw design diagram of system. Show all necessary connections between components.

b) [20 points] Write an Assembly program to do followings.

- Assume PIA register names are already defined as DIRECTION.A, DIRECTION.B, PORT.A, PORT.B.
- Perform conditionings of PIA direction registers.
- In an endless loop, program should read the digital temperature value from PIA.
  - If temperature is less than 128, program should flash (turn on and off) Blue LED continuously.
- Otherwise, program should flash Red LED continuously, and also should generate a beep sound.
- Call a subroutine named WAIT from main program, after each write operation to PIA.
  (Assume the WAIT subroutine is already written, so there is no need to write its codes.)

INSTRUCTION SET

Transfer	Logic	Pseudo Directives	Branch - Compare	Branch - Compare
MOV Move	AND And	ORG Origin	CMP Compare	BIO Branch if overflow
LDA Load	OR Or	EQU Equal	BIT Bit test	BNO Branch if not
STA Store	XOR Exclusive or	RMB Reserve	BRA Branch	overflow
EXC Exchange	CLR Clear	memory bytes	JMP Jump	BIC Branch if carry
CHN Change	SET Set	DAT Data	JMC Jump conditionally	BNC Branch if not carry
	COM Complement	END End	BEQ Branch if equal	BIH Branch if half carry
Shift/Rotate	NEG Negate		BNE Branch if	BNH Branch if not HC
LSL Logical shift left		Arithmetic	not equal	BSR Branch to
LSR Logical shift right	Operational	ADD Add	BGT Branch if greater	subroutine
ASR Arith. shift right	PSH Push	ADC Add with carry	BGE Branch if greater	JSR Jump to subroutine
ROL Rotate left	PUL Pull	SUB Subtract	or equal	BSC Branch to
ROR Rotate right	EIN Enable interrupt	SUE Subtract with	BLT Branch if less than	subroutine
	DIN Disable interrupt	carry	BHI Branch if higher	conditionally
	NOP No operation	MUL Multiply	BHE Branch if higher or	JSC Jump to subroutine
	INT Interrupt	DIV Divide	equal	conditionally
	RTS Return from subrout	INC Incremenet	BLO Branch if lower	
	RTI Return from interrupt	DEC Decrement		

ANSWERS

ANSWER 1) [20 points]

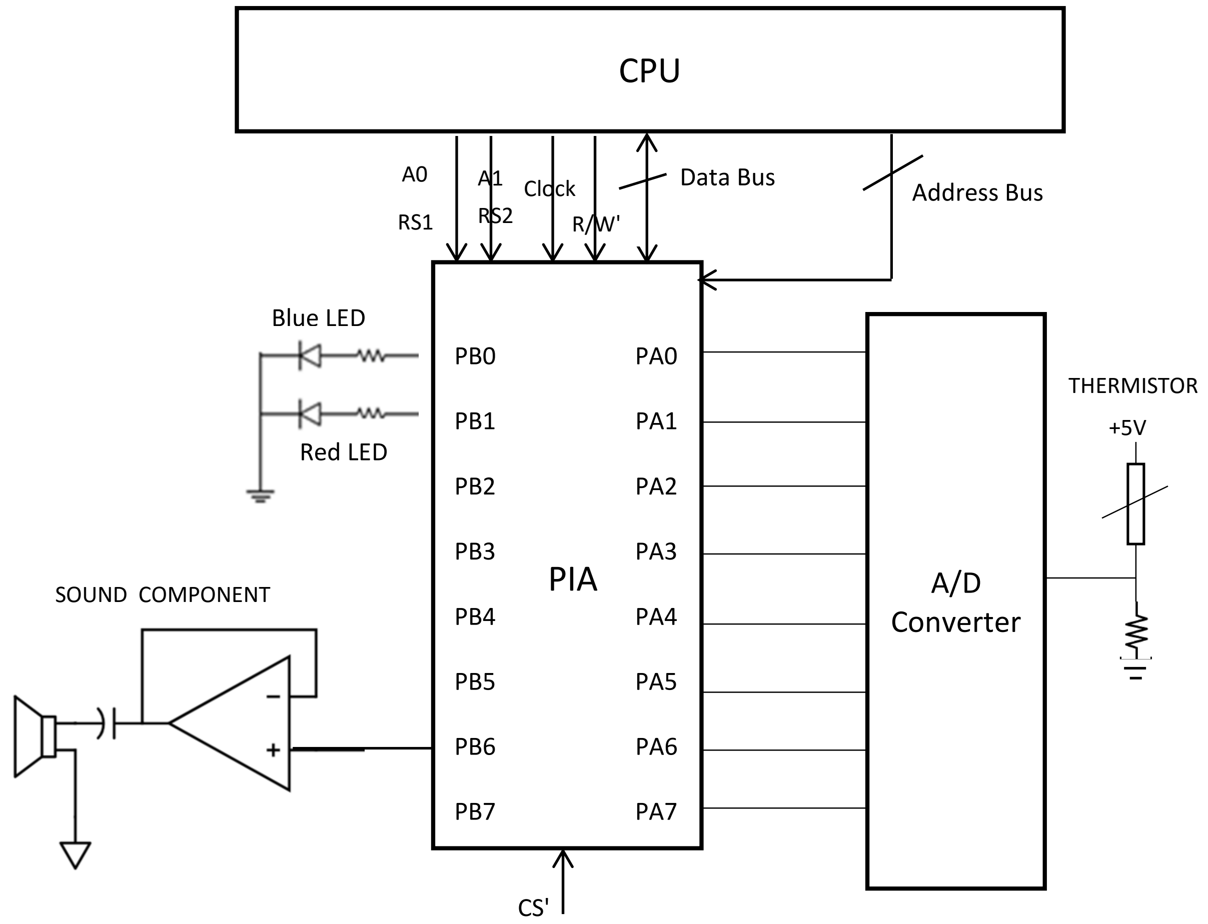
```
RESULT RMB 1
*Expected result is 170 ($AA)

START
  LDA A, %11111111        ;A is first operand of XOR command (Calculates the result)
  MOV B, A                       ;B is second operand of XOR command

LOOP
  LSR B   ;Logical shift right B register (MSB bit becomes zero)
  XOR A, B          ;XOR A and B
  CMP B, %00000001        ;Loop limit checking
  BNE  LOOP         ;Go to loop

  STA A, RESULT   ;Store the result
  INT   ; Stop
```

ANSWER 2) [30 points]

```
SIZE  EQU  4
MATRIX   RMB  SIZE*SIZE
     ORG  MATRIX
     DAT  6, 1, 8, 0
     DAT  5, 9, 4, 3
     DAT  7, 1, 5, 2
     DAT  2, 0, 8, 7

SUM_COL1   RMB  1
*Expected sum result is 20 ($14)

START
  LDA A, 0                      ;Sum register
  LDA SK, MATRIX ; Address index and loop counter

LOOP
*Add a data from matrix (first column)
  ADD A, <SK+0> + SIZE      ;Increment SK by SIZE, after ADD operation
  CMP SK, SIZE*SIZE
  BLT LOOP          ;Go to row loop

  STA A, SUM_COL1           ;Store the result
  INT
```

ANSWER 3) [50 points]

a) [30 points]



b) [20 points]

```
START
* Conditioning of PIA ports
  STA  $00,  DIRECTION.A      ; All bits of Port-A are input
  STA  $FF,  DIRECTION.B      ; All bits of Port-B are output

LOOP
  LDA A, <PORT.A>            ;Read THERMISTOR value (0-255)
  CMP A, 128
  BLO BLUE_LED

  STA %01000010,  PORT.B            ;Display the RED LED (at PB1), and also generate beep  sound (at PB6)
  BRA CONTINUE

BLUE_LED
  STA %00000001,  PORT.B            ;Display the BLUE LED (at PB0)

CONTINUE
  BSR WAIT
  STA %00000000,  PORT.B            ;Turn OFF all LEDs (for the flashing effect), and also stop the sound
  BSR WAIT
  BRA LOOP                          ; Endless loop
```