BLG 212E - Microprocessor Systems

FINAL EXAM

Resources are closed. Exam duration is 2 hours.

QUESTION 1) [25 points] Write an Assembly program to do followings.
(Use instruction set of EDU-CPU.)

- Define a variable named ARRAY with 6 elements (each element is 1 byte).
- Initialize the ARRAY with values 10,20,30,40,50,60.
- By looping, multiply each element of array with 4.
(Note that left-shift instruction can be used as a way to multiply a number by 2.)

QUESTION 2) [20 points] Write an Assembly program to calculate the factorial of a number.
N! = 1*2*3*4*.....*N

- Define a symbol named NUMBER (for N) which is equal to 5.
- Define a variable named FACTORIAL with 2 bytes.
- By looping, calculate the factorial.

QUESTION 3) [55 points]

a) [25 points]

- A microprocessor-based system with a PIA will be designed.
- PIA will be connected to CPU.
- A 7-segment LED display will be connected to PIA.
- Type of LED display is common ground, which gives light when logical 1.
- Draw the detailed design of the hardware system.
- Show all connections between the CPU and PIA, and also between the PIA and LED display.

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

- Assume an array named LED_CHAR_TABLE is already defined and initialized as below (for 0-9).

```
LED_CHAR_TABLE RMB 10
 ORG LED_CHAR_TABLE
 DAT %10000001      ;0
 DAT %11111001      ;1
 DAT %10100100      ;2
 .......
```

- Also assume the following symbols are already defined for PIA registers.
Direction register (PIA_DIRECTION / YÖNLEN.B)
Port register (PIA_PORT / İSKELE.B)
- Perform the conditioning of PIA_DIRECTION as output.
- By looping, program should continuously display the LED characters one by one from 0 to 9.
- Call the subroutine WAIT after displaying each LED character.
- Write the subroutine WAIT, which should count from 1 to 30000.

INSTRUCTION SET

Transfer
MOV Move
LDA Load
STA Store
EXC Exchange
CHN Change

Shift/Rotate
LSL Logical shift left
LSR Logical shift right
ASR Arithmetic shift right
ROL Rotate left
ROR Rotate right

Logic
AND And
OR Or
XOR Exclusive or
CLR Clear
SET Set
COM Complement
NEG Negate

Operational
PSH Push
PUL Pull
EIN Enable interrupt
DIN Disable interrupt
NOP No operation
INT Interrupt
RTS Return from subroutine
RTI Return from interrupt

Pseudo Directives
ORG Origin
EQU Equal
RMB Reserve memory bytes
DAT Data
END End

Arithmetic
ADD Add
ADC Add with carry
SUB Subtract
SUE Subtract with carry
MUL Multiply
DIV Divide
INC Incremenet
DEC Decrement

Branch - Compare
CMP Compare
BIT Bit test
BRA Branch
JMP Jump
JMC Jump conditionally
BEQ Branch if equal
BNE Branch if not equal
BGT Branch if greater than
BGE Branch if greater or equal
BLT Branch if less than
BHI Branch if higher
BHE Branch if higher or equal
BLO Branch if lower

Branch - Compare
BIO Branch if overflow
BNO Branch if not overflow
BIC Branch if carry
BNC Branch if not carry
BIH Branch if half carry
BNH Branch if not half carry
BSR Branch to subroutine
JSR Jump to subroutine
BSC Branch to subroutine conditionally
JSC Jump to subroutine conditionally

BLG 212E - FINAL EXAM ANSWERS

QUESTION 1) [25 points]

```
SIZE EQU  6
ARRAY RMB SIZE
  ORG ARRAY
  DAT 10,20,30,40,50,60

START
     LDA  SK, ARRAY           ;Get beginning address of ARRAY
     LDA  CD, 0               ;CD is loop counter
DEVAM
     LDA  A, <SK+CD+0>        ;Get next number from ARRAY
     LSL A                    ;Logical Shift Left (multiplies by 2)
     LSL A                    ;Logical Shift Left (multiplies by 2)
     STA A, <SK+CD+0>         ;Store result to ARRAY
     INC  CD                  ;Increment loop counter
     CMP  CD, SIZE            ;Compare with array size
     BLT  DEVAM               ;Goto loop
     INT
```

QUESTION 2) [20 points]

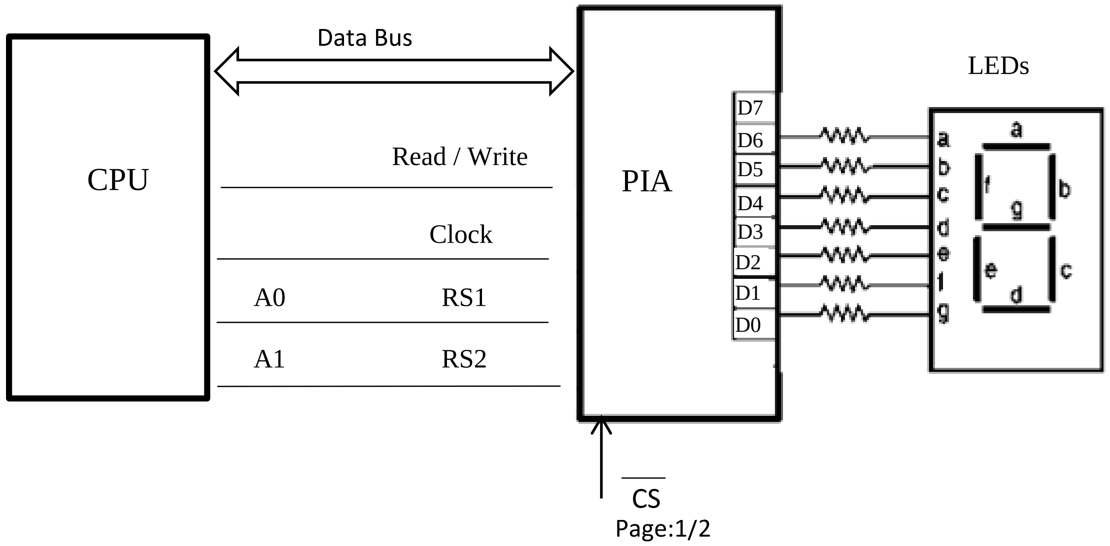
```
NUMBER     EQU  5          ;Loop limit symbol
FACTORIAL      RMB  2      ;Factorial variable

START
  STA  1, FACTORIAL        ;Initialize variable to 1
  LDA  C, 1                ;C register is loop counter

DEVAM
  MOV A, C                  ;Copy C counter to A
  MUL  A, <FACTORIAL>      ; Multiply A with factorial, result is in AB
  STA  AB, FACTORIAL        ;Store AB to factorial
  INC  C                    ;Increment loop counter
  CMP  C, NUMBER            ;Compare with loop limit
  BLO  DEVAM                ;Goto loop
  INT
```

QUESTION 3) [55 points]

a) [25 points]



b) [30 points]

```
START
          STA  $FF, <YÖNLEN.B>       ; All bits of PIA Port-B are output

INITIAL
*SK is used as an index on LED_CHAR_TABLE
*B is used as loop counter

     LDA SK, LED_CHAR_TABLE       ;Get beginning address of table
     LDA B, 0                     ;Initialize loop counter to 0

DEVAM
     LDA A, <SK+0>                ;Get corresponding LED character from Table
     STA   A,  <İSKELE.B>          ;Write to Port-B
     BSR WAIT                     ;Call wait subroutine
     INC B                        ;Increment loop counter
     INC SK                       ;Increment SK
     CMP B,10                     ;Compare to loop limit
     BLO DEVAM                    ;Goto inner loop

     BRA  INITIAL           ;Goto main loop

*-----------------
WAIT   LDA CD,30000     ;CD is loop counter
DONGU   DEC CD          ;Decrement  counter
      CMP CD, 0         ;Compare with 0
      BNE DONGU        ;If not zero goto loop
      RTS              ;Return from subroutine
```