BLG 212E - Microprocessor Systems
MIDTERM EXAM

Resources are closed. Exam duration 90 minutes.

QUESTION 1) [15 points] Consider the Assembly program on right. Suppose pipelined processing method is used for Fetch-Decode-Execute-Write phases.

- Draw the table for pipelining showing instruction numbers and phases for each instruction.
- Find the total number of instruction cycles for the program.

```
1  START LDA A, <$1000>
2    CMP A, 0
3    BLT DEVAM
4    BRA END
5  DEVAM STA 0, <$1000>
6  END  INT
```

QUESTION 2) [55 points]
A memory subsystem will be designed containing the following modules. (Address bus is 13 bits, Data bus is 8-bits).

- Total 2 KB ROM memory, by using 1K×8 bit ROM modules.
- Total 2 KB RAM memory, by using 1K×8 bit RAM modules.
- All memory modules are in consecutive address map.

a) [25 points] Calculate the total capacity and the total used memory. For each memory chip, write the address range (smallest and biggest addresses) with hexadecimal notation.

b) [30 points] Draw the memory design with all necessary connections. (Address bus, Data bus, Chip select signals).

- Use an address decoder.
- Assume memory chip select signals are active high (1).

QUESTION 3) [30 points] Write an Assembly program to do followings.

- Define a symbol named SIZE which is equal to 5 (array size).
- Define a variable named ARRAY, each element is 1 byte.
- Initialize the array with values 10,3,8,6,7.
- Define a variable named EVEN_COUNT which is 1 byte.
- By looping, calculate the count of even numbers (EVEN_COUNT) in array.
- Note that if the rightmost bit of a number is 0, then it is an even number. You may use the right-shift instruction to get rightmost bit into Carry flag and check the Carry flag.

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