BLG 212E - SAMPLE MIDTERM EXAM QUESTIONS

Books, notes, electronic devices, etc are closed. Exam duration is 1.5 hours.

QUESTION 1) [15 points] For each of the followings, write the result (8-bit unsigned) of the operations as Hexadecimal, Binary, and Decimal numbers. Also write the Status Flag bits whose content is set to 1.

a) $7E - $C2

b) $35 Or $1F

c) Not $70
(One's complement)

QUESTION 2) [40 points] A memory subsystem will be designed with the following memory chips.
All memory chips are in consecutive address map. Data bus is 8-bits.

- 4 KB ROM memory, by using one 4K×8 bit ROM chip.
- 4 KB RAM memory, by using two 2K×8 bit RAM chips.
- 1 KB RAM memory, by using one 1K×8 bit RAM chip.

a) [20 points] Determine the minimum number of address lines required.
Calculate the total memory capacity and the total used memory.
For each memory chip, write the address map (smallest and biggest addresses) in hexadecimal notation.

b) [20 points] Draw the memory design with all connections (address bus, data bus, chip select signals, and the address decoder).

QUESTION 3) [45 points]
a) [15 points] Draw an algorithm FlowChart,
b) [30 points] And also write an Assembly program,
to determine whether an array is sorted in increasing order or not.

You should use the EDU-CPU instruction set for the Assembly program.

- Define a constant symbol named SIZE, which is equal to 12.
- Define a variable named ARRAY, whose length is the defined SIZE. Each element is 1 byte.
- Initialize the ARRAY with the following decimal data : 1, 3, 4, 5, 7, 8, 8, 12, 10, 11, 15, 14
- Define a variable named RESULT, whose length is 1 byte.
- By looping, process all elements of ARRAY, in order to determine the value of the RESULT.
  - If the array is sorted increasingly, then the RESULT variable should be assigned 1 (as True).
  - Otherwise the RESULT variable should be assigned zero (as False).

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 overflow
STA Store	XOR Exclusive or	RMB Reserve memory bytes	BRA Branch	BIC Branch if carry
EXC Exchange	CLR Clear	DAT Data	JMP Jump	BNC Branch if not carry
CHN Change	SET Set	END End	JMC Jump conditionally	BIH Branch if half carry
	COM Complement		BEQ Branch if equal	BNH Branch if not half carry
Shift/Rotate	NEG Negate	Arithmetic	BNE Branch if not equal	BSR Branch to subroutine
LSL Logical shift left		ADD Add	BGT Branch if greater than	JSR Jump to subroutine
LSR Logical shift right	Operational	ADC Add with carry	BGE Branch if greater or equal	BSC Branch to subroutine conditionally
ASR Arithmetic shift right	PSH Push	SUB Subtract	BLT Branch if less than	JSC Jump to subroutine conditionally
ROL Rotate left	PUL Pull	SUE Subtract with carry	BHI Branch if higher	
ROR Rotate right	EIN Enable interrupt	MUL Multiply	BHE Branch if higher or equal	
	DIN Disable interrupt	DIV Divide	BLO Branch if lower	
	NOP No operation	INC Incremenet		
	INT Interrupt	DEC Decrement		
	RTS Return from subroutine			
	RTI Return from interrupt			

ANSWERS

QUESTION 1) [15 points] All results are interpreted as <u>8-bit unsigned.</u>

a)

Result:
Hex = $BC
Bin = %1011 1100
Dec = 188

Status Flags:
O=1 N=1 C=1

b)

Result:
Hex = $3F
Bin = %0011 1111
Dec = 63

Status Flags:
None

c)

Result:
Hex = $8F
Bin = %1000 1111
Dec = 143

Status Flags:
N=1

QUESTION 2) [40 points]

2a) [20 points]
The chip with 4K locations (2^{12}) requires 12 address lines for location selection.
The address decoder requires 2 address lines.
Total minimum number of lines in the Address Bus is = 12 + 2 = 14

Total address capacity = $2^{14} = 2^4 * 2^{10}$ = 16 KB
Total used amount of memory = 4 + 2 + 2 + 1 = 9 KB

A12 and A13 lines are used for decoder.
A11 line is used to distinguish between M2 and M3.
For M4, A10 and A11 lines will be zero always.

Smallest Adresses :

Chip	Cap	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	HEXADECIMAL
M1	4K	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000
M2	2K	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000
M3	2K	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1800
M4	1K	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2000

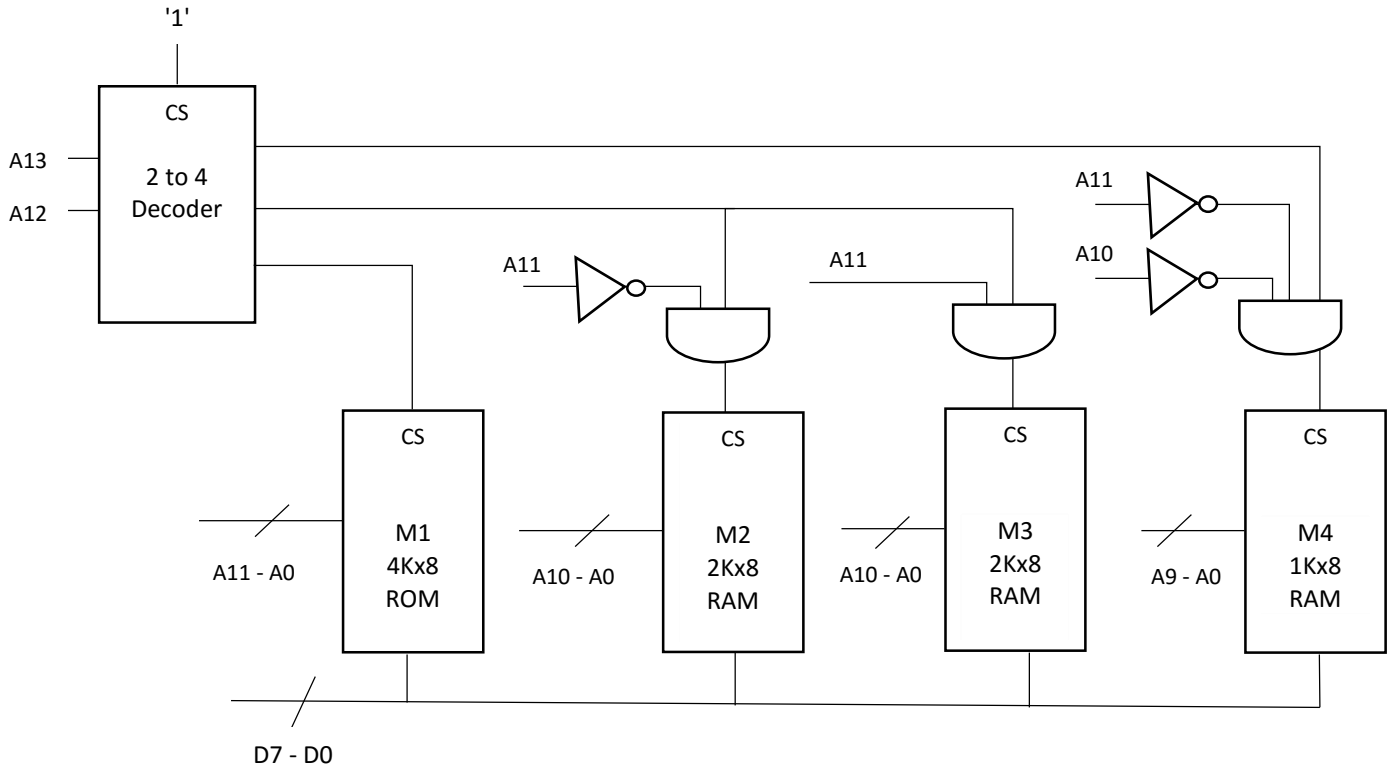
Biggest Adresses :

Chip	Cap	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	HEXADECIMAL
M1	4K	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0FFF
M2	2K	0	1	0	1	1	1	1	1	1	1	1	1	1	1	17FF
M3	2K	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1FFF
M4	1K	1	0	0	0	1	1	1	1	1	1	1	1	1	1	23FF

Memory Map :

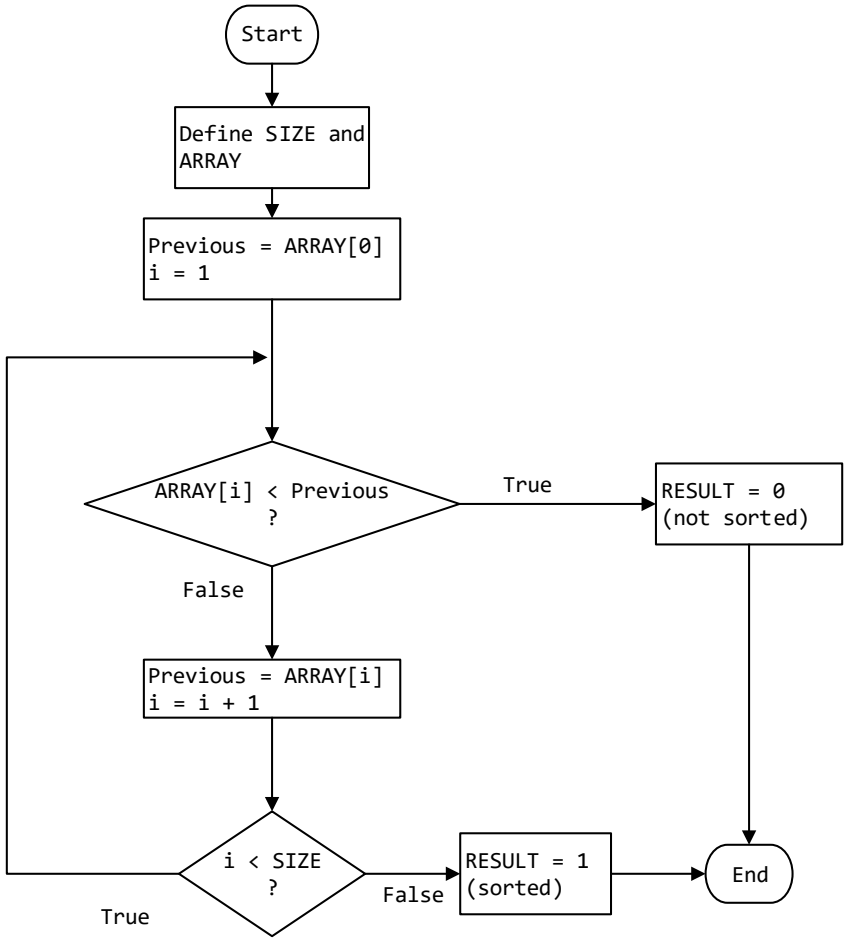
Chip Name	Memory Chip Type	Smallest Address	Biggest Address
M1	4 K x 8 bit ROM	0000	0FFF
M2	2 K x 8 bit RAM	1000	17FF
M3	2 K x 8 bit RAM	1800	1FFF
M4	1 K x 8 bit RAM	2000	23FF

2b) [20 points]



QUESTION 3) [45 points]

3a) [15 points]



3b) [30 points]

```
*Program determines whether an array is sorted or not (increasing order).

SIZE EQU 12
ARRAY   RMB  SIZE
   ORG  ARRAY
*    DAT  1, 3, 4, 5, 7, 8, 8, 12, 10, 11, 15, 14          ;Not sorted sample
   DAT  1, 3, 4, 5, 7, 8, 8, 9, 10, 11, 15, 16           ;Sorted sample

RESULT   RMB  1          ; $01 means array is sorted, $00 means NOT sorted

START
   LDA  B, <ARRAY>       ;Previous data (Get first element in ARRAY)
   LDA  SK, ARRAY+1      ;Adress of second element in ARRAY

DONGU
   LDA A, <SK+0>;Get current data from ARRAY
   CMP A, B      ;Compare current data with previous data
   BLT UNSORTED          ;If A is less than B

   MOV B, A      ;Move Current data to Previous data
   INC  SK               ;Increment loop counter
   CMP SK, ARRAY+SIZE    ;Compare to loop limit
   BNE  DONGU   ;Go to loop

   STA  $01, RESULT      ;Array is sorted
   INT    ;Stop

UNSORTED
   STA  $00, RESULT      ;Array is NOT sorted
   INT    ;Stop
```