a) The largest value that a 60-bit unsigned binary integer can represent is ___$2^{60}-1 = 1.1529215e+18$

b) How many bits are required to address a 4M x 16 main memory If the memory is

Byte addressable: There are 4M x 2 bytes which equals $2^2 \times 2^{20} \times 2 = 2^{23}$; so 23 bits are needed

Word addressable: There are 4M words, which equals $2^2 \times 2^{20} = 2^{22}$; so 22 bits are needed

c) Let's assume that register ECX contains the following 32 bit number represented in hexadecimal: B876508Eh. The programmer writes the byte 6Ah into register CH. What is the new value of ECX: ____B8766A8E_____

d)

```
.data
     myWord  WORD 1000h
     myDword DWORD 10000000h
.code
     inc myWord
     dec myWord
     inc myDword
     mov ax,00FFh
     inc ax
     mov ax,00FFh
     inc al
```

what is the value of AX: _____0000h_____

e)

```
.data
     myByte BYTE 0FFh, 0
.code
     mov al,myByte
     mov ah,[myByte+1]
     dec ah
     inc al
     dec ax
```

what is the value of AX: _____FEFFh_____

f)

```
.data
     valB BYTE -1
     valW WORD +32767
.code
     mov al,valB
     neg al
     neg valW
```

what is the value of valW: _____-32767_____

g) **For the following instruction**

```
mov bl,10001111b
movsx ax,bl
```

what is the value of AX: _____11111111 10001111_____

h) The decimal number $(0100)_{10}$ in hexadecimal number system can be written as **0100h or 0000 0001 0000 0000**

i) $(EF)_{16} + (11)_{10} =$ **$(239)_{10} + (11)_{10} = (250)_{10}$ or $(EF)_{16} + (B)_{16} = (FA)_{16}$**

j) The fractional binary number 0.11 has a decimal value of _____**0.75**_____

k) For the binary number 1000, the weight of the column with the 1 is _____**8**_____

```
start: sbn temp,temp,.+1    # Sets temp to zero
       sbn temp,a,.+1       # Sets temp to -a
       sbn b,b,.+1          # Sets b to zero
       sbn b,temp,.+1       # Sets b to -temp, which is a
```

Question 5 [25 Marks]

One line Answer for the following questions is required, write to the point answers otherwise negative marking would be done for detailed/extra information.

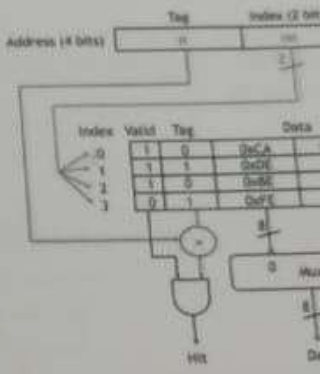
a) When we copy a block of data from main memory to the cache, where exactly should we put it?
The lowest k bits of the address specify a cache block. — The upper (m - k) address bits are stored in the block's tag field. — **The data from main memory is stored in the block's data field.**

b) How can we tell if a word is already in the cache, or if it has to be fetched from main memory first? [1 mark]
The tags contain the address information required to identify whether a word in the cache corresponds to the requested word.

c) Eventually, the small cache memory might fill up. To load a new block from main RAM, we'd have to replace one of the existing blocks in the cache, which one? [1 mark]
Based on highest LRU factor or mod procedures

d) How can write operations be handled by the memory system? [1 mark]
Memory write operation **transfers the address of the desired word to the address lines, transfers the data bits to be stored in memory to the data input lines. Then it activates the write control line.**

Figure 1: Question 5 reference



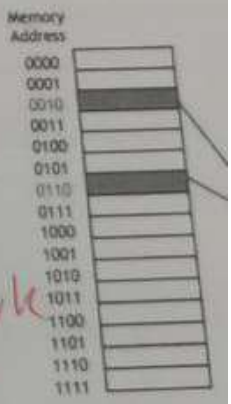
e) For the addresses below, what byte is read from the cache (or is there a miss)? [2 marks]

Mandatory: Fill the given block for required address distribution (offset, tag, block/set no)

1. 1010 _____HIT___
2. 1110 _____MISS___
3. 0001 _____HIT___
4. 1101 _____**MISS**__

Tag	Index	Offset
___1_bits	___2_bits	__1_bits

f) If we have a direct mapped cache and memory as below, what happens if a program uses addresses 2, 6, 2, 6, 2, for 1 byte block size data? [2 marks]



2,miss 6,miss 2,miss 6, miss 2 miss

g) Considering the previous question memory and addresses, what happened if we have a fully-associative cache? [2 marks]

2,miss 6,miss 2,hit 6, hit, 2 hit
we might put memory address 2 in cache block 2, and address 6 in block 3. Then subsequent repeated accesses to 2 and 6 would all be hits instead of misses

h) Where would data from memory byte address 6195 be placed, assuming the eight-block cache designed, with 16 bytes per block? [3 marks]

- For the 1-way cache: set index 011
- For the 2-way cache: set index 11
- For the 4-way cache: set index 1

Reason/Explanation:

6195 in binary is 00...0110000 0110011. ♣ Each block has 16 bytes, so the lowest 4 bits are the block offset. ♣ For the 1-way cache, the next three bits (011) are the set index. For the 2-way cache, the next two bits (11) are the set index. For the 4-way cache, the next one bit (1) is the set index.

i) Assume a number of cache lines, each holding 16 bytes. Assume a 24–bit address. The simplest arrangement is an **associative cache** would be : [2 marks]

Bits		
Fields	Tag	Offset

Divide the 24–bit address into two parts: a 20–bit tag and a 4–bit offset.

Bits	23 – 4	3 – 0
Fields	Tag	Offset

j) Assume 256 cache lines, each holding 16 bytes. Assume a 24–bit address. The simplest arrangement is an **direct mapped cache** would be: [3 marks]

Bits	12	8	4
Cache View	Tag	Line	Offset
Memory Address View	20		4

call that $256 = 2^8$, so that we need eight bits to select the cache line.

vide the 24–bit address into three fields: a 12–bit explicit tag, an 8–bit line number, and a 4–bit set within the cache line. Note that the 20–bit memory tag is divided between the 12–bit cache tag l 8–bit line number.

Bits	23 – 12	11 – 4	3 – 0
Cache View	Tag	Line	Offset
Address View	Block Number		Offset

k) Consider the address **0xAB7129**. From the above cache/memory view it would have

1. Tag:
2. Line:
3. Offset:

Suppose a 2-way set-associative implementation of the same cache memory. Again assume 256 cache lines, each holding 16 bytes. Assume a 24-bit address. Following information is required based on spatial and temporal locality principals. [5 marks]

- Tag:
- Line:
- Offset:

If followed (5.

Attempt => 2/1

Correct => 0.5

Consider addresses:

1. **0xCD4128**
2. **0xAB7129**

Entry 0				Entry 1			
D	V	Tag	Contents	D	V	Tag	Contents

Recall that $256 = 2^8$, so that we need eight bits to select the cache line.

Consider addresses 0xCD4128 and 0xAB7129. Each would be stored in cache line 0x12. Set 0 of this cache line would have one block, and set 1 would have the other.

Entry 0				Entry 1			
D	V	Tag	Contents	D	V	Tag	Contents
1	1	0xCD4	M[0xCD4120] to M[0xCD412F]	0	1	0xAB7	M[0xAB7120] to M[0xAB712F]

l) Let's examine the sample address in terms of the bit divisions below. [2 marks]

The Address 0xAB7129

Bits	23 – 4	3 – 0
Fields	Tag	Offset

Bits:	23 – 20	19 – 16	15 – 12	11 – 8	7 – 4	3 – 0
Hex Digit						
Field (hex values)						

Bits:	23 – 20	19 – 16	15 – 12	11 – 8	7 – 4	3 – 0
Hex Digit	A	B	7	1	2	9
Field	0xAB712					0x09

Question 6 [10 Marks]

Assume direct mapped caches. [5 marks]

1. Assume that memory block 0xAB712 is present in cache line 0x12.
2. We now get a memory reference to address 0x895123. This is found in memory block 0x89512, which must be placed in cache line 0x12.
3. Two bonus marks will be given based on writing favorite point of this paper page 1.
4. What steps would holds for each of a memory read from or memory write to 0x895123.

1. The valid bit for cache line 0x12 is examined. If (Valid = 0) go to Step 5.
2. The memory tag for cache line 0x12 is examined and compared to the desired tag 0x895. If (Cache Tag = 0x895) go to Step 6.
3. The cache tag does not hold the required value. Check the dirty bit. If (Dirty = 0) go to Step 5.
4. Here, we have (Dirty = 1). Write the cache line back to memory block 0xAB712.
5. Read memory block 0x89512 into cache line 0x12. Set Valid = 1 and Dirty = 0.
6. With the desired block in the cache line, perform the memory operation.

Consider variations of mappings to store 256 memory blocks [5 marks]

Direct Mapped Cache	256 cache lines	256 Sets
1–Way Set Associative	256 cache lines (fill below accordingly)	256 sets (fill below accordingly)
2–Way Set Associative		
4–Way Set Associative		
8–Way Set Associative		
16–Way Set Associative		
32–Way Set Associative		
64–Way Set Associative		

Rubrics
full mark for correct answer

5 → last two invalid values.
4 → any one invalid value.

128–Way Set Associative		
256–Way Set Associative		
Fully Associative		

Direct Mapped Cache	256 cache lines	
"1–Way Set Associative"	256 cache lines	1 set per line
2–Way Set Associative	128 cache lines	2 sets per line
4–Way Set Associative	64 cache lines	4 sets per line
8–Way Set Associative	32 cache lines	8 sets per line
16–Way Set Associative	16 cache lines	16 sets per line
32–Way Set Associative	8 cache lines	32 sets per line
64–Way Set Associative	4 cache lines	64 sets per line
128–Way Set Associative	2 cache lines	128 sets per line
256–Way Set Associative	1 cache line	256 sets per line
Fully Associative Cache	1 num	256 sets

Question 7 [15 Marks]

Fill in the following table for different caches. Assume all caches are direct mapped. Also rough work is mandatory for each part and carry marks. [12 marks]

No.	Address Bits	Cache Size	Block Size	Tag Bits	Index Bits	Offset Bits	Bits per Row
a.	16	16KB	8B	2	11	3	67
b.	32	32KB	16B	17	11	4	146
c.	64	1024	64B	44	14	6	57
d.	32	512KB	32B	13	14	5	270

Rubrics
→ only correct values → 1.5 num each
→ 3 no each (correct rough work)
→

Consider the above question address distribution (part a and b) and give appropriate example if possible: [3 marks]

Description	Address Bits	Example
Two addresses referring to same address block	16	Having same bloc bits
Two addresses having same tag bits	32	Having same tag bits

9	H
10	H
11	H
12	M

12 ⇒ 50 55

⇒ Miss Rate

⇒ 8 ⇒ 006

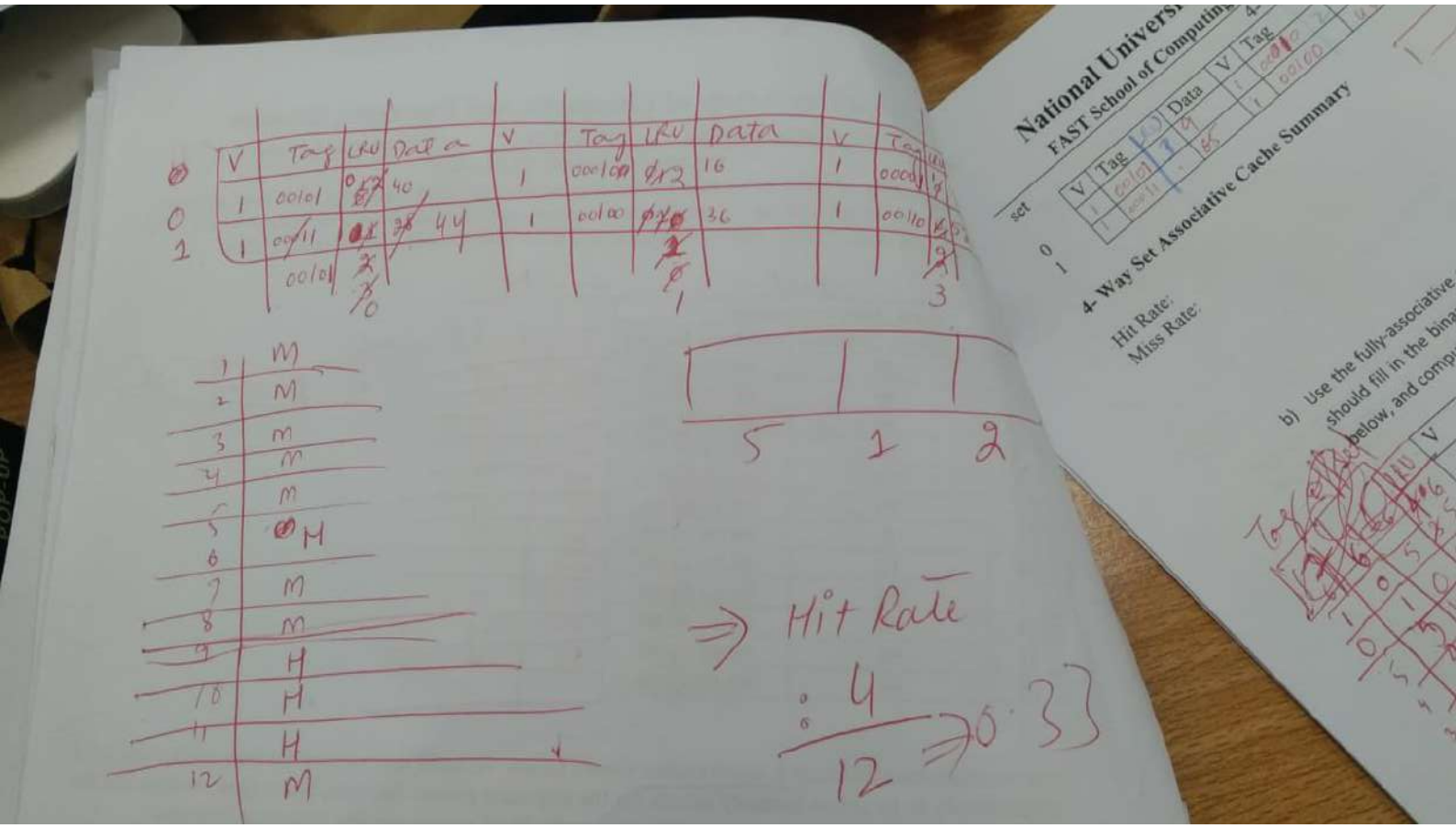
12

Rubrics:

miss/rate:	~~0~~
LRU:	~~1~~
End Data filled:	~~3~~ 1

3c. memory free in 1 still miss (1.5)

greater hit Rate (1.5)

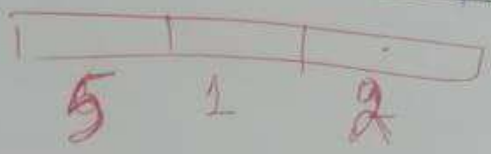

National Universi
FAST School of Computing
set
V
Tag
Data
4- Way Set Associative Cache Summary
Hit Rate:
Miss Rate:
b) Use the fully-associative
should fill in the bina
below, and comp
5
1
2
⇒ Hit Rate
: 4
12 ⇒ 0.33

4-WAY Set Associative Cache

set	V	Tag	LRU	Data	V	Tag	Data	V	Tag	Data	V	Tag	Data
0	1	00101	1	9	1	00010	486	1	00001	345	1	00101	9
1	1	00111	0	65	1	00100	435	1	00110	97	1	00001	456

4- Way Set Associative Cache Summary

Hit Rate:
Miss Rate:



b) Use the fully-associative cache to facilitate memory access for the memory sequence above. You should fill in the binary form of the Tag values. Show the final contents of the cache in the table below, and compute the hit rate and miss rate. [8 marks]

V	Tag	Data
	28	
	40	
	36	
	16	
	52	
	8	
	12	
	44	

Fully Associative Cache Summary

Hit Rate: 4/12 => 0 3
Miss Rate: 8/12 => 0 6

Rough Work

c) Write the comparison for both approaches.

Question 9 [15 Marks]

Perform unsigned binary division: Invalid bit selection will be marked as zero. [3 marks]
Note: your computer width is 8-bit, whereas dividend is 04Eh and divisor is 003h.

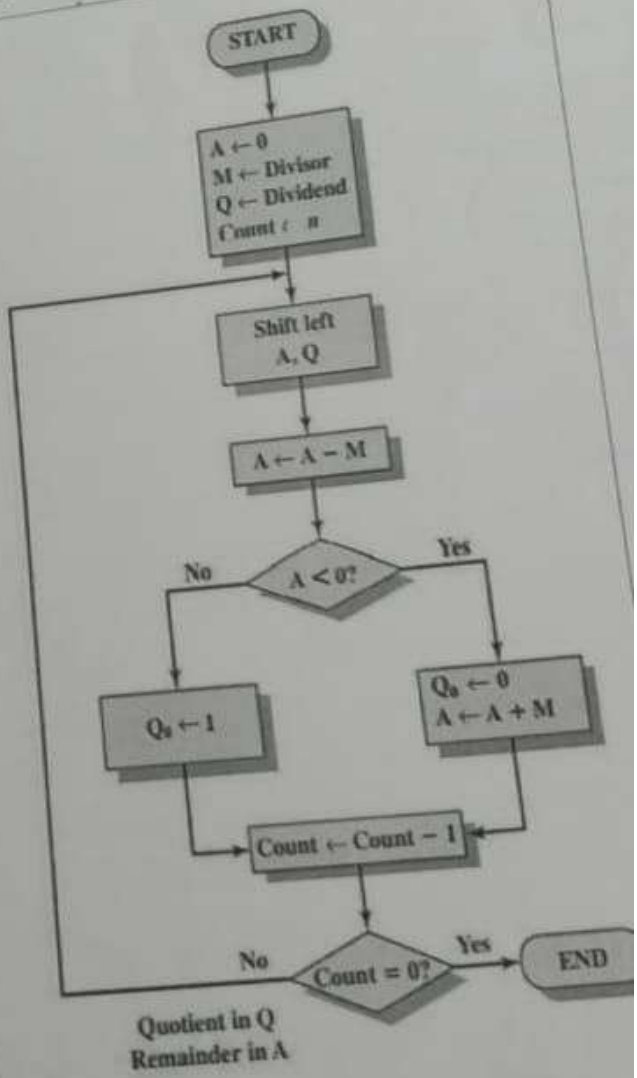
a) Hexa to Binary Conversion for both dividend and divisor

0 4 E

0000 0100 1110

003h

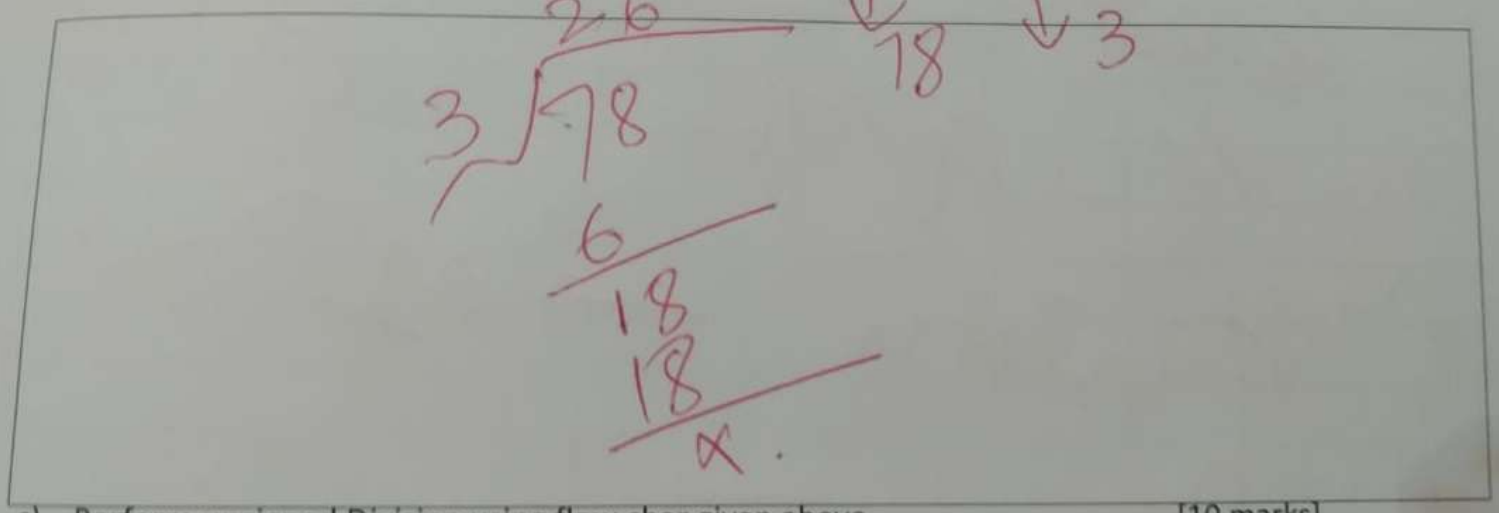
00 0000 0011



National University of Computer and Emerging Sciences

FAST School of Computing **Fall-2022** **Islamabad Campus**

b) Perform Decimal Division using above decimal dividend and divisor [2 marks]



c) Perform unsigned Division using flow char given above [10 marks]

M= 0000 0011 , -M= 1111 1101