CSE 332/Sections: 4 & 5/MT-2/Marks = 40/ Time = 1 Hour 30 mins/Fall 2020

1. For a system, assume, RAM= 64KB, block size = 4 bytes, cache size = 64 bytes, 2-way set associative mapping. Given the state of the cache shown below, find hit/miss for the following addresses generated by CPU. In case of cache hit, show the contents of memory read by the processor. 10

CACHE MEMORY

SET	TAG OF LINE	CONTENTS
0	9B1H	1CH
		2DH
		30H
		4FH
	2BAH	EDH
		BDH
		ACH
		CAH
1	7A1H	A0H
		A2H
		A0H
		A0H
	468H	4BH
		3BH
		2BH
		1BH
2	90CH	30H
		20H
		60H
		40H
	2F7H	49H
		39H
		29H
		69H
3	736H	14H
		C4H
		34H
		4CH
	763H	49H
		33H
		24H
		10H

CACHE MEMORY

SET	TAG OF LINE	CONTENTS
4	6ECH	10H
		20H
		30H
		40H
	73FH	FFH
		EEH
		DDH
		CCH
5	3BBH	10H
		20H
		30H
		40H
	F44H	41H
		33H
		20H
		1FH
6	CF7H	11H
		22H
		33H
		44H
	1DDH	25H
		35H
		45H
		55H
7	67BH	91H
		21H
		31H
		B1H
	09BH	D0H
		A0H
		20H
		1FH

Memory address	Set	Tag	Hit/Miss	Content read from cache in case of Cache Hit
90C4H	1	486H	M	
9B1AH	6	4D8H	M	
41EAH	2	20FH	M	
3BB8H	6	1DDH	H	25H
F445H	1	7A1H	H	A2H
6ECFH	3	736H	H	C4H
E7F1H	4	73FH	H	FFH
5EEBH	2	2F7H	H	69H
5743H	0	2BAH	H	CAH
CF7CH	7	67BH	H	91H

Hints:
90C4H = 1001 0000 1100 0100
Address format: (need to derive)

Tag	Set	word
11 bits	3 bits	2 bits
100 1000 0110	001	00
4 8 6 H	1	

Compare the Tag with two tags (7A1H & 468H) associated with Set-3.
Doesn't match, so it's not found in cache: Cache Miss

2. For a system, RAM = 64KB, Block size = 4 bytes, Cache size = 128 bytes, Direct mapped cache. Calculate the Hit ratio while CPU runs program "Test_Cache". Also count how many blocks are replaced in cache memory assuming the cache is empty at the beginning. 10
For the same RAM, block size and Cache memory, what would be the Hit ration in case of 2-way Set Associative Mapping?

Main Program—"Test_Cache"

Memory address (decimal)	
0003	Instruction-1
0004	Instruction-2
0005	Instruction-3 (Call Function NSU-1)
0006	Instruction-4
0007	Instruction-7 (Call Function NSU-2)
0008	Instruction-8
0009	Instruction-9 (End of program)

Function NSU-1

Memory address (decimal)	
0128	Instruction-1
0129	Instruction-2
0130	Instruction-3
0131	Instruction-4
0132	Instruction-5
0133	Instruction-6 (return to Instruction-4 of main program--- Test_Cache)

Function NSU-2

Memory address (decimal)	
0136	Instruction-1
0137	Instruction-2
0138	Instruction-3
0139	Instruction-4 (return to Instruction-8 of main program --- Test_Cache)

Direct mapping:
Address format:

Tag (9 bits)	Line (5 bits)	Word (2 bits)

Address in sequence	RAM Block No (Address is divided by 4 and quotient is used	Cache Line No (RAM block no is divided by 32 and remainder is	H/M	Consequence

	here)	used here)		
003	0	0	M	Block-0 is transferred in Line-0
004	1	1	M	Block-1 is transferred in Line-1
005	1	1	H	
128	32	0	M	Block-32 is transferred in Line-0. Block-0 is replaced
129	32	0	H	
130	32	0	H	
131	32	0	H	
132	33	1	M	Block-33 is transferred in Line-1. Block-1 is replaced
133	33	1	H	
006	1	1	M	Block-1 is transferred in Line-1. Block-33 is replaced
007	1	1	H	
136	34	2	M	Block-34 is transferred in Line-2.
137	34	2	H	
138	34	2	H	
139	34	2	H	
008	2	2	M	Block-2 is transferred in Line-2. Block-34 is replaced
009	2	2	H	

Hit ratio: (10/17) x 100 = 58.8%
Number of blocks replaced = 4

2-way Set-Associative mapping:
Address format:

Tag (10 bits)	Set (4 bits)	Word (2 bits)

Address in sequence	RAM Block No (Address is divided by 4 and quotient is used here)	Cache Set No (RAM block no is divided by 16 and remainder is used here)	H/M	Consequence
003	0	0	M	Block-0 is transferred in Line-1 of Set-0
004	1	1	M	Block-1 is transferred in Line-1 of Set-1
005	1	1	H	
128	32	0	M	Block-32 is transferred in Line-2 of Set-0
129	32	0	H	
130	32	0	H	
131	32	0	H	
132	33	1	M	Block-33 is transferred in Line-2 of Set-1
133	33	1	H	

006	1	1	H	
007	1	1	H	
136	34	2	M	Block-34 is transferred in Line-1 of Set-2.
137	34	2	H	
138	34	2	H	
139	34	2	H	
008	2	2	M	Block-2 is transferred in Line-2 of Set-2
009	2	2	H	

Hit ratio: (11/17) x 100 = 64.7%

3. A 12-bit Hamming code word containing 8 bits of data and 4 parity bits is read from memory. What was the original 8-bit data word that was written into memory if the 12-bit word read out is 110010111001 10

Data Bits = 8 bits
Parity bits = 4 bits

Bit position	12	11	10	9	8	7	6	5	4	3	2	1
Position Number	1100	1011	1010	1001	1000	0111	0110	0101	0100	0011	0010	0001
Data bit & Check bit	D_8	D_7	D_6	D_5	C_8	D_4	D_3	D_2	C_4	D_1	C_2	C_1
Word read out	1	1	0	0	1	0	1	1	1	0	0	1

Data Read: 11000110

Error Code Read: 1101

Generating error code:

$C_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_5 \oplus D_7$
$= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$

$C_2 = D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7$
$= 0 \oplus 1 \oplus 0 \oplus 0 \oplus 1 = 0$

$C_4 = D_2 \oplus D_3 \oplus D_4 \oplus D_8$
$= 1 \oplus 1 \oplus 0 \oplus 1 = 1$

$C_8 = D_5 \oplus D_6 \oplus D_7 \oplus D_8$
$= 0 \oplus 0 \oplus 1 \oplus 1 = 0$

Error Code = 0100

XOR

$$
\begin{array}{r}
1101 \\
0100 \\
\hline
1001
\end{array}
$$

Error is in 9th position

D_5 is corrupt

Hence, original 8-bit data word that was written into memory is:

11010110

4. a) Convert 387.5625 to IEEE 32-bit floating point format. 10
b) Convert the following IEEE 32 bit into decimal.

Sign bit	exponent	significant
0	10001110	00110000000000000000000

Given decimal value,

+387.625

Converting 387 to binary gives us

110000011

Converting 0.625 to binary gives us 0.101

$(387.625)_{10} = (110000011.101)_2$

110000011.101

$= 1.10000011101 \times 2^8$

Exponent biased by 127 → 8 + 127 = 135

$(135)_{10} = (10000111)_2$

IEEE 32-bit format:

Sign bit	Exponent	Significant
0	10000111	10000011101000000000000

Note: Sign bit is 0 for positive numbers

Given,
sign bit = 1, so it is a negative number

$$\text{Exponent} = (10001110)_2$$
$$= (142)_{10}$$

$$\text{Actual exponent} = 142 - 127$$
$$= 15$$

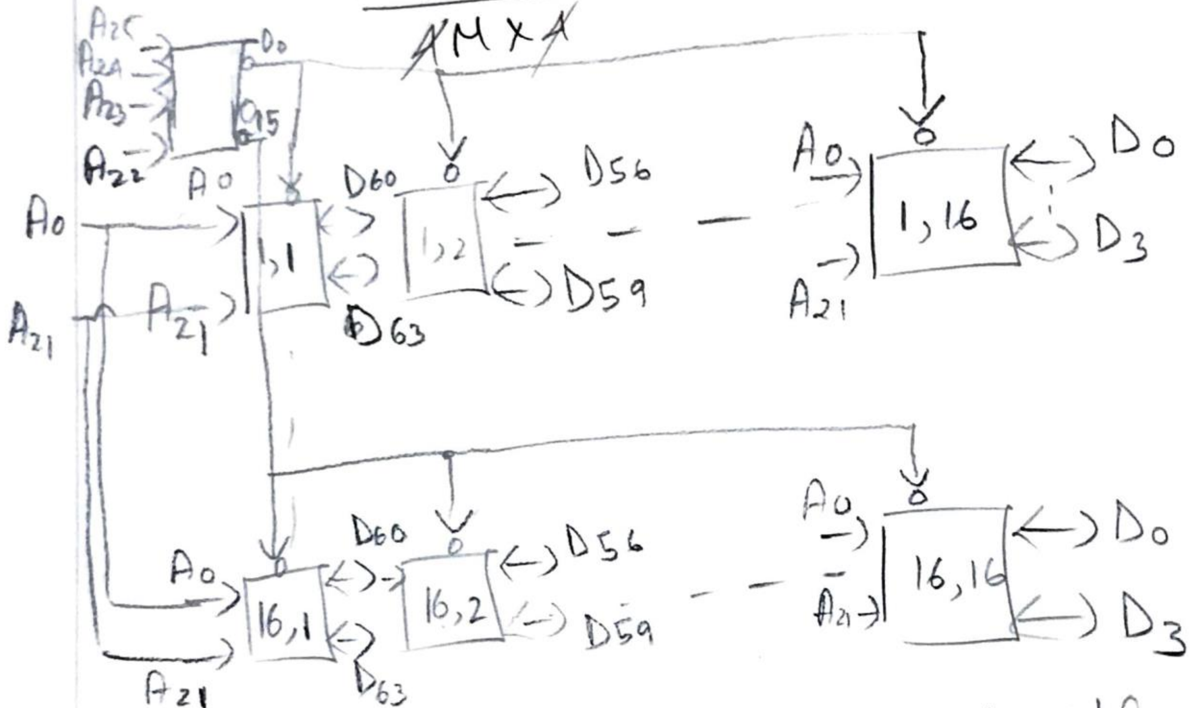
$$\text{So, the binary significant} = (\cancel{0}1.0011)_{10}$$
$$= (1.1875)_2$$

$$\text{Value in decimal} = -(1.1875 \times 2^{15})$$
$$= -38912$$

Quiz - 4

64 M X 64 bits using 4 M X 4 bits

$$\frac{\overset{16}{\cancel{64}} M \times \overset{16}{\cancel{64}}}{\cancel{4} M \times \cancel{4}} = 16 \text{ rows } \cancel{X} \text{ by } 16 \text{ columns}$$



All A0's will be connected with each other

All A21's will be connected

4 to 16 line decoder is needed

All the decoder's output pins will be connected with enable pins of ICs