Q.1 A computer employs RAM chips of 512 x 8 & ROM chips of 2048 x 4. The computer system needs 2M x 16 of RAM & 4K x 16 of ROM & 2 interface units with 256 registers each. A memory mapped I/O configuration is used.

a.) How many RAM & ROM chips are needed?

⇒(i) $p\text{-factor} = \frac{N'}{N} = \frac{2 \times 2^{10} \times 2^{10}}{2^9} = 2^{12}$

$q\text{-factor} = \frac{W'}{W} = \frac{2^4}{2^3} = 2$

∴ Number of RAM chips needed = $p \times q = 2^{13} =$ **8192 chips**

(ii) $p\text{-factor} = \frac{N'}{N} = \frac{2^2 \times 2^{10}}{2^{11}} = 2$

$q\text{-factor} = \frac{W'}{W} = \frac{2^4}{2^2} = 2^2 = 4$

∴ Number of ROM chips needed = $p \times q = 2 \times 4 =$ **8 chips**

S.No.	Type	N x W	N' x W'	p	q	x	y	z	Total (x+y+z)
1	RAM	512 x 8	2M x 16	2^{12}	2	9	12	2	23
2	ROM	2048 x 4	4K x 16	2	4	11	1	2	14
3	Interface		256	1	1	8	01	2	11

(b.) Tell how many lines of the address bus must be used to access total memory?

⇒ RAM:- 9 lines of address bus must be used to access total memory.

ROM:- 11 lines of address bus must be used to access total memory.

Interface:- 8 lines of address bus must be used to access total memory.

[consider 'x' for address bus lines accessing total memory].

(c.) Propose, the number of lines must be decoded for chip select? Specify the size of the decoder.

⇒ We consider 'y+z' for chip select lines

RAM :- $y+z = 12+2 = \boxed{14}$ lines must be decoded for chip select.

Decoder size : 14×2^{14}

ROM :- $y+z = 1+2 = \boxed{3}$ lines must be decoded for chip select

Decoder size : 3×8

Interface :- $y+z = 1+2 = \boxed{3}$ lines must be decoded for chip select

Decoder size : 3×8

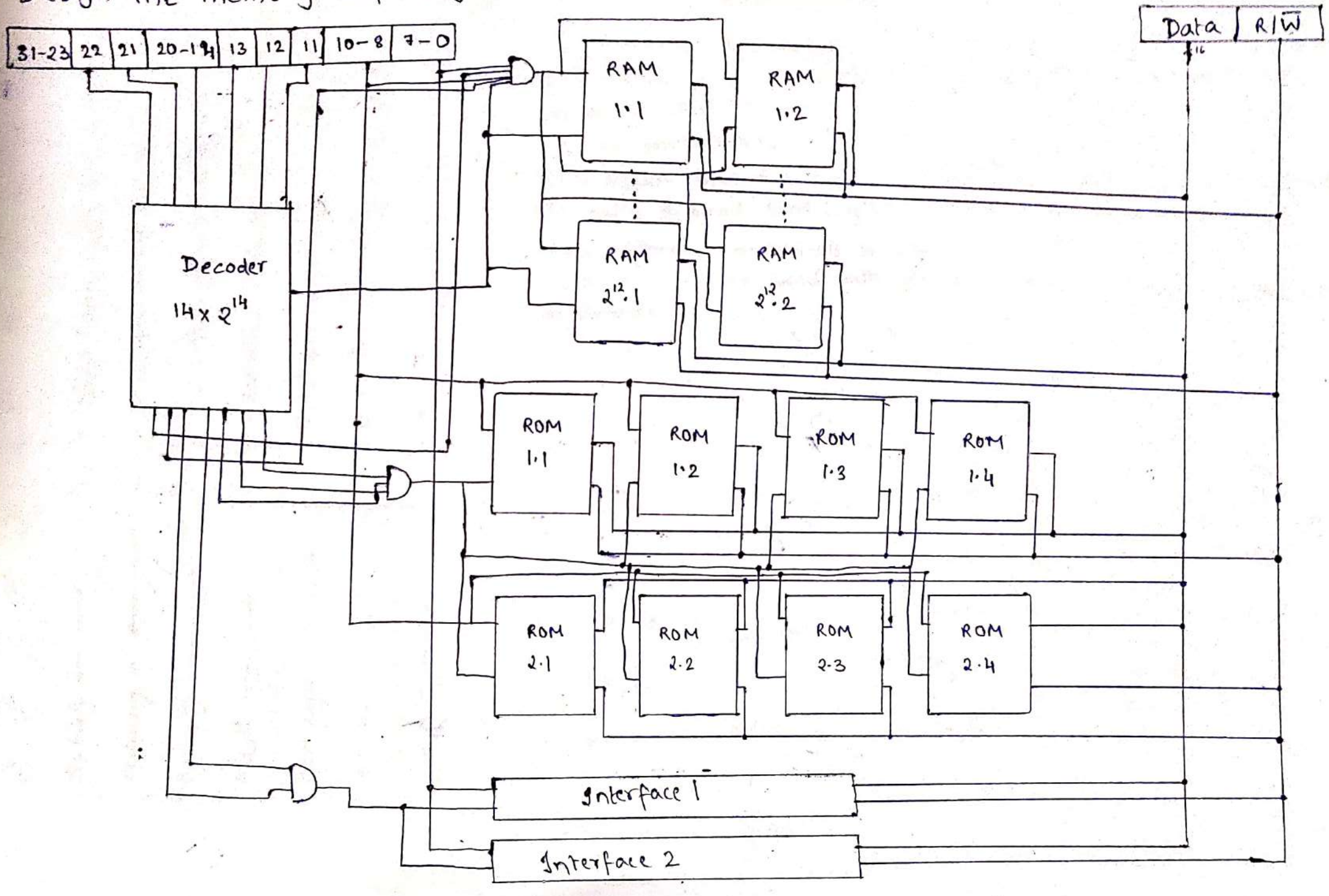
(d.) Devise a memory-address map for the system & Give the address range in hexadecimal for RAM, ROM & Interface

⇒

Component	Hexadecimal From	Hexadecimal To	Address Lines 31 30 29 28 27 26 25 24 23	z 22 21	y 20 19 18 17 16 15 14 13 12 11 10 9	x 8 7 6 5 4 3 2 1 0
RAM 1.1	00000000H	000001FFH	0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 0	x x x x x x x x x
RAM 1.2	00000000H	000001FFH	0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 0	x x x x x x x x x
RAM 2.1	00000000H	000003FFH	0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 1	x x x x x x x x x
RAM 2.2	00000000H	000003FFH	0 0 0 0 0 0 0 0 0	0 0	0 0 0 0 0 0 0 0 0 0 0 1	x x x x x x x x x
⋮	⋮	⋮	⋮	⋮	⋮	⋮
RAM 2^{12}.1	001FF800H	001FFFFFH	0 0 0 0 0 0 0 0 0	0 0	1 1 1 1 1 1 1 1 1 1 1 1	x x x x x x x x x
RAM 2^{12}.2	001FF800H	001FFFFFH	0 0 0 0 0 0 0 0 0	0 0	1 1 1 1 1 1 1 1 1 1 1 1	x x x x x x x x x

Component	Hexadecimal From	Hexadecimal To	Address Lines 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ROM 1.1	1800H	17FFH	0 0 0 1 0 x x x x x x x x x x x x
ROM 1.2	1000H	17FFH	0 0 0 1 0 x x x x x x x x x x x x
ROM 1.3	1000H	17FFH	0 0 0 1 0 x x x x x x x x x x x x
ROM 1.4	1000H	17FFH	0 0 0 1 0 x x x x x x x x x x x x
ROM 2.1	1800H	1FFFH	0 0 0 1 1 x x x x x x x x x x x x
ROM 2.2	1800H	1FFFH	0 0 0 1 1 x x x x x x x x x x x x
ROM 2.3	1800H	1FFFH	0 0 0 1 1 x x x x x x x x x x x x
ROM 2.4	1800H	1FFFH	0 0 0 1 1 x x x x x x x x x x x x
I/F 1	2000H	20FFH	0 0 1 0 0 0 0 0 x x x x x x x x
I/F 2	2800H	28FFH	0 0 1 0 1 0 0 0 x x x x x x x x

e.) Design the memory chip layout for that address map.



Q. 2 Consider a computer with the following characteristics: total of 1M byte of main memory; word size of 1 byte; block size of 16 bytes; & cache size of 64K bytes.

⇒ Main memory = 1M = 2^{20} bytes

1 Block = 16 bytes = 2^4 bytes/words (as 1 word = 1 byte)

Number of blocks in main memory = $\frac{2^{20}}{2^4} = 2^{16}$ bytes

Cache = 64K = 2^{16} bytes

Number of cache lines = $\frac{2^{16}}{2^4} = 2^{12}$ bytes

(A) For the main memory address of F0010, 01234, & CABBE, gin the corresponding tag, cache line address, & word offsets for a direct-mapped cache.

⇒ Bits 0-3 indicate the word offset (4 bits)

Bits 4-15 indicate the cache line address slot (12 bits)

Bits 16-19 indicate the tag (remaining bits)

(i) F0010 = 1111 0000 0000 0001 0000

Word offset = 0000 = 0

Cache line address = 0000 0000 0001 = 001

Tag = 1111 = F

F	001	0

(ii) 01234 = 0000 0001 0010 0011 0100

Word offset = 0100 = 4

Cache line address = 0001 0010 0011 = 123

Tag = 0000 = 0

0	123	4

(iii) CABBE = 1100 1010 1011 1011 1110

Word offset = 1110 = E

Cache line address = 1010 1011 1011 = A B B

Tag = 1100 = C

C	ABB	E

(B) Give any 2 main memory addresses with different tags that map to the same cache slot for a direct mapped cache

⇒ (i) 3FFF0 → | 3 | FFF | 0 |

(ii) 2FFF0 → | 2 | FFF | 0 |

(C) For the main memory addresses of F0010 & CABBE, give the corresponding tag & offset values for a fully-associative cache

⇒ Tag bits = 16 bits (for a fully-associative cache)

(i) F0010

Word offset = 0h

Tag = F001h

(ii) CABBE

Word offset = Eh

Tag = CABBh

(D) For the main memory addresses of F0010 & CABBE, give the corresponding tag, cache set, & offset values for a 2-way set-associative cache.

⇒ K = 2 ∴ Number of sets in a cache = $\frac{2^{12}}{2} = 2^{11}$

Bits 0-3 indicate the word offset (4)

Bits 4-14 indicate the cache set (11)

Bits 15-19 indicate the tag (remaining)

(i) F0010 = 11110 000 0000 0001 0000

Word offset = 0000 = 0

Cache set = 000 0000 0001 = 001

Tag = 11110 = 1 1110 = 1E

(ii) CABBE = 11001 010 1011 1011 1110

Word offset = 1110 = E

Cache set = 010 1011 1011 = 2BB

Tag = 11001 = 19

Q.3 For a direct-mapped cache design with a 32-bit address, the following bits of the addresses are used to access to the cache

Tag (31-10)	Index (9-5)	offset (4-0)

(a.) What is the cache block size (in words)?

⇒ Block size = 2^N

N = number of offset bits − 2

* '−2' because we are using 32 bit words & byte-addressed memory

∴ Cache Line / Block size = $2^{5-2} = 2^3 = 8$ words

(b.) How many entries does the cache have?

⇒ Entries = $2^{\text{index bits}} = 2^5 = 32$ entries

(c.) What is the ratio between total bits required for such a cache implementation over the data storage bits?

⇒ Total bits = Tag bits + valid bit + data bit = 22 + 1 + 32×8 = 279

Data bits = 32×8 = 256

Ratio = $\frac{279}{256} = 1.089$

(d.) Starting from power on, the following byte-addressed cache references are recorded:

Addresses											
0	4	16	132	232	160	1024	30	140	3100	180	2180

Address	Index	Offset	Hit/Miss	Replace?	Final value?
0	00000	00000	M	N	N
4	00000	00100	H	N	N
16	00000	10000	H	N	N
132	00100	00100	M	N	N
232	00111	01000	M	N	Y
160	00101	00000	M	N	Y
1024	00000	00000	M	Y	N
30	00000	11110	M	Y	N
140	00100	01100	H	N	N
3100	00000	11100	M	Y	Y
180	00101	10100	H	N	Y
2180	00100	0100	M	Y	Y

(e.) How many blocks are replaced?

⇒ From the table : 4 blocks are replaced

(f.) What is the hit ratio ?

⇒ Hit ratio = $\frac{\text{No. of hits}}{\text{No. of hits + No. of misses}} = \frac{4}{4+8} = \frac{4}{12} = \frac{1}{3} = 0.33$ i.e 33%

(g.) List the final state of the cache with each valid entry represented as a record <index, tag, data>.

⇒ From the last column in the above table.

<00100, 0010, mem [2176] >

<00101, 0000, mem [160] >

<00000, 0011, mem [3072] >

<00111, 0000, mem [224] >

Q.4 A process references 5 pages A, B, C, D, and E, in the following order : A; B; C; D; A; B; E; A; B; C; D; E;, Assume that the replacement algorithm is (i) First-in-first-out & (ii) LRU.

Find the number of page transfers during this sequence of references starting with an empty main memory with 3 page frames. Repeat for 4 page frames. Calculate hit ratio.

(i) FIFO :-

A →	B →	C →	D →	A →	B →	E →	A →	B →	C →	D →	E	Reference ~~Frame~~
A	A	A	B	C	D	A	A	A	B	E	E	Frame 1
	B	B	C	D	A	B	B	B	E	C	C	Frame 2
		C	D	A	B	E	E	E	C	D	D	Frame 3
1	2	3	4	5	~~6~~	~~7~~	-	-	8	9	-	Count
M	M	M	M	M	M	M	H	H	M	M	H	Hit/Miss

Page transfers during given sequence of ~~frames~~ references with 3 page frames = 9

A →	B →	C →	D →	A →	B →	E →	A →	B →	C →	D →	E	Reference
A	A	A	A	A	A	B	C	D	E	A	B	Frame 1
	B	B	B	B	B	C	D	E	A	B	C	Frame 2
		C	C	C	C	D	E	A	B	C	D	Frame 3
			D	D	D	E	A	B	C	D	E	Frame 4
1	2	3	4	-	-	5	6	7	8	9	10	Count
M	M	M	M	H	H	M	M	M	M	M	M	Hit/Miss

Page transfer count for 4 page frames = 10

$$\text{Hit ratio} = \frac{3}{3+9} = \frac{3}{12} = 0.25 \quad \text{(for 3 page frames)}$$

$$\text{Hit ratio} = \frac{2}{2+10} = \frac{2}{12} = 0.166 \quad \text{(for 4 page frames)}$$

(ii) LRU :- (Least Recently Used)

→ 3 page frames

Reference	A	B	C	D	A	B	E	A	B	C	D	E
Frame 1	A	A	A	D	D	D	E	E	E	C	C	C
Frame 2		B	B	B	A	A	A	A	A	A	D	D
Frame 3			C	C	C	B	B	B	B	B	B	E
Count	1	2	3	4	5	6	7	-	-	8	9	10
Hit/Miss	M	M	M	M	M	M	M	H	H	M	M	M

Page transfer count for 3 page frames = 10

Hit ratio = $\frac{2}{2+10}$ = 0.166

→ 4 page frames

Reference	A	B	C	D	A	B	E	A	B	C	D	E
Frame 1	A	A	A	A	A	A	A	A	A	A	A	E
Frame 2		B	B	B	B	B	B	B	B	B	B	B
Frame 3			C	C	C	C	E	E	E	E	D	D
Frame 4				D	D	D	D	D	D	C	C	C
count	1	2	3	4	-	-	5	-	-	6	7	8
Hit/Miss	M	M	M	M	H	H	M	H	H	M	M	M

Page transfer count for 4 page frames = 8

Hit ratio = $\frac{4}{4+8} = \frac{4}{12}$ = 0.33

Q.5 A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is x^3+1. What is the actual bit transmitted? Suppose the 3rd bit from the left is inverted during transmission. How will the receiver detect this error?

$\Rightarrow$ $x^3+1 \equiv 1001$

Since generator polynomial is of 3rd degree, append 3 zeros to the lower end of frame to be transmitted.

∴ Bit stream = 10011101000

```
          10001100
     ____________
1001 | 10011101000
       1001
       ----
        0001
        0000
        ----
         0011
         0000
         ----
          0110
          0000
          ----
           1101
           1001
           ----
            1000
            1001
            ----
             0010
             0000
             ----
              0100
              0000
              ----
               100   (remainder)
```

CRC = 100

Actual frame transmitted :- 10011101000
+ 100

10011101100

∴ Actual bit transmitted = **10011101100**

Now, 3rd bit from left gets inverted during transmission.

So, the bit stream received by the receiver = 10111101100

Receiver performs binary division with same generator polynomial

```
                1 0 1 0 1 0 0 0
            ____________________
1 0 0 1   | 1 0 1 1 1 1 0 1 1 0 0
            1 0 0 1
            ---------
              0 1 0 1
              0 0 0 0
              -------
                1 0 1 1
                1 0 0 1
                -------
                  0 1 0 0
                  0 0 0 0
                  -------
                    1 0 0 1
                    1 0 0 1
                    -------
                      0 0 0 1
                      0 0 0 0
                      -------
                        0 0 1 0
                        0 0 0 0
                        -------
                          0 1 0 0
                          0 0 0 0
                          -------
                            1 0 0   (remainder)
```

Remainder is non-zero which indicates that an error occurred in the data during the transmission.

Therefore, receiver rejects the data & asks the sender for transmission