Tut 5: Q1.

Memory unit has 256K words of 32 bits.

$\therefore$ 256 K words $\Rightarrow 2^8 \cdot 2^{10} = 2^{18}$

So 18 address bits.

apart from this a mode field to specify one of seven addressing modes.

$\therefore$ to specify seven addressing mode we need 3 bits as $2^3 = 8$.

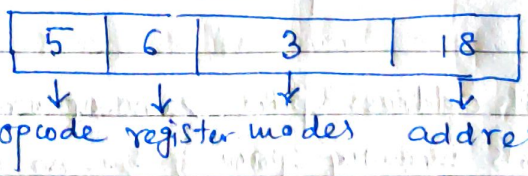
So for Mode $\rightarrow$ 3 bits.

60 processor registers are there

so corresponding required bits are 6. as $2^6 = 64$

So out of 32 bits, 18 bits were used for address.
3 bits are used for specifying modes
6 bits are " " " register.
total of 27 bits
remaining bits are $32 - 27 = 5$ bits.

$\therefore$ structure



Tut 5. Q2.

32 bit instructions

12 bit addresses.

So one word, or one address instruction requires 12 bit addresses.

For two word or two address instruction it requires (12+12) bit.

Now instruction is of 32 bits.

opcode	Address 1	Address 2
8 bit	12 bit	12 bit

→ two address instruction = 32 bits.

So there are 8 bit for opcode

total no of possible combination for one and two address instructions can be $2^8 = 256$.

If there are 250 two-address instructions.

so possible no. of one-address instruction = 256 − 250

= 6.

32 bit instruction 12 bit one address instruction

so in case of one address instruction

opcode	Address
6×2^{12}	12 bit.

Maximum no. of one address instruction = $6 \times 2^{12} = 24,576$

Tut 5.

Q3. 24 bits address space ⇒ a. no. of words in address space = 2^{24}.

16 bits memory space ⇒ b. no. of words in memory space = 2^{16}

c. a page consists of 2K words

∴ no of pages = $2^{24}/2K = 2^{24}/2 \cdot 2^{10} = 2^{13}$ pages

no. of blocks = $2^{16}/2K = 2^{16}/2 \cdot 2^{10} = 2^5$

= 32 blocks.

Tut 5.

Q4. Memory unit of $64K \times 16$.

$\Rightarrow 2^6 \cdot 2^{10} \times 16$

$\Rightarrow 2^{16} \times 16$

so 16 bit address, 16 bit data

cache uses direct mapping of block size of four words.

for four words address bit required $2 [2^2 = 4]$.

a. cache memory size = $1K = 2^{10}$.

so total address lines = 10.

for words, address lines = 2.

∴ for block, address lines = $10 - 2 = 8$.

total address lines = 16.

so no. of bits for tag field = $16 - (2+8) = 6$.

so memory structure

tag	block	word
6	8	2

Index = $8+2 = 10$

b. there are 6 bits for tag

1 bit for valid bit

16 bit of data

valid	tag	data
1	6	16 = 23

so there are 23 bits in each word of cache.

c. cache can accommodate no. of blocks as $2^8 = 256$.

256 blocks of four words each.

Q2. Multiply -8 & -9 using Booth's Algorithm.

$8 \rightarrow 1000 \Rightarrow -8 \rightarrow$ (1's) $\rightarrow 0111$

2's $\rightarrow 1000$

$\therefore -8 \rightarrow 11000$ (5 bit).

$9 \rightarrow 1001 \Rightarrow -9 \rightarrow$ (1's) $\rightarrow 0110$

2's $\rightarrow 0111$

$-9 \rightarrow 10111$ (5 bit).

Multiplier $-9 \Rightarrow 10111$ in BR.

$\therefore \overline{BR} = 01000$

$\overline{BR} + 1 = 01001$

Qn	Qn+1		AC	QR	Qn+1	SC
			00000	11000	0	101
0	0	Shr.	00000	01100	0	100
0	0	Shr.	00000	00110	0	011
0	0	Shr	00000	00011	0	010
1	0	SUB BR	01001			
			01001			
		Shr.	00100	10001	1	001
1	1	Shr.	00010	01000	1	000

Result from AC & QR

$\Rightarrow 00010\ 01000$

$\Downarrow$

72.