A-05

COSC 211

Que 3

(a) Hex encoding :- 0xad100000

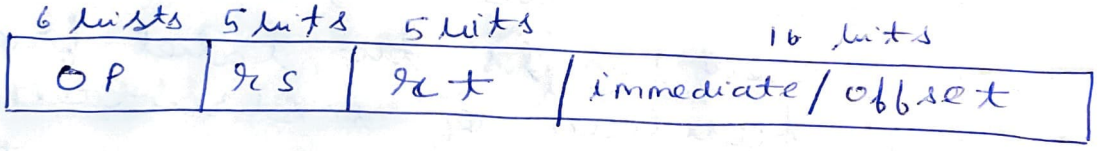
Explanation :-

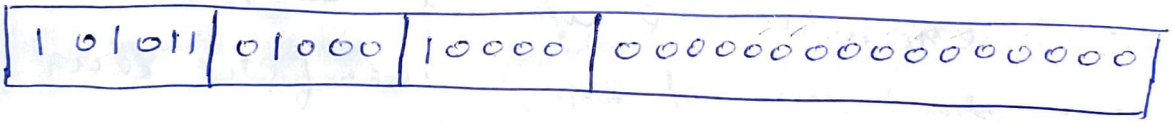
sw $s0 0($t0)

=> sw $16 0($8) offset

$16 → rt, $8 → rs

I - Format





101011 01000 10000 0000 0000 0000 0000

0xad100000

Hence, proved.

(b) beq is encoded as 0x1212 0007

Binary representation :

0001 0010 0001 0010 0000 0000 0000 0111

16 bits.

beq :- | 000100 | rs | rt | offset |

| 000100 | 10000 | 10010 | 0000000000000111 |

The offset is actually determined by counting how far is the label from the relative position of instruction. In our example, exit is 7 instructions far from beq. Therefore our offset is 7.

Binary representation :- 0000 0000 0000 0111

hex representation :- 0x0007.

How we use :- The offset is sign extended, multiplied by 4, and added to value of PC when condition is met, PC is updated with PC + 4 + offset * 4 to point to label.

In our example, offset is 7.

Therefore $PC = PC + 4 + 7 * 4$.

(c) j encoded as 0x08100007:

J- Format.

6 bits	26 bits
OP	address.

In the address we have absolute address of the label divided by 4.

In our question we have:- j loop

address of loop in 26 bits :-

00 0010 0000 0000 0000 0001 1100

when divided by 4 :-

00 0001 0000 0000 0000 0000 0111

000010	00 0001 0000 0000 0000 0000 0111

Binary :-

0000 1000 0001 0000 0000 0000 0000 0111

Hexadecimal :- 0x08100007

Hence, proved.