Operation	operands	description	example (r5=8,r1=2,r3=6, r4=2)		
add	dest,src1,src2	dest = src1+src2	add r3,r5,r1	r3 = r5+r1	r3 = 8+2 = A
sub	dest,src1,src2	dest = src1-src2	sub r3,r5,r1	r3 = r5-r1	r3 = 8-2 = 6
and	dest,src1,src2	dest = src1&src2	and r3,r5,r1	r3=r5&r1	r3 = 8&2 = 0
or	dest,src1,src2	dest = src1|src2	or r3,r5,r1	r3=r5|r1	r3 = 8|2 = A
xor	dest,src1,scr2	dest = src1^src2	xor r3,r5,r1	r3=r5^r1	r3 = 8^2 = A
comp	dest,src1,scr2	dest = 1 if (src1==src2) else 0	comp r3,r5,r1 comp r3,r1,r4	r3 =(8==2)?1:0 r3 = (2==2)?1:0	r3 = 0 r3 = 1
slt	dest,src1,scr2	dest = 1 if (src1<src2) else 0	slt r3,r5,r1 slt r3,r1,r5	r3 =(8<2)?1:0 r3 = (2<8)?1:0	r3 = 0 r3 = 1
addi	dest,src1,immediate_value	dest = src1+immediate_value*	addi r3,r5,4	r3 = r5+4	r3 = 8+4 = C
subi	dest,src1,immediate_value	dest = src1-immediate_value*	subi r3,r5,4	r3 = r5-4	r3 = 8-4 = 4
andi	dest,src1,immediate_value	dest = src1&immediate_value*	andi r3,r5,9	r3 = r5&9	r3 = 8&9=8
ori	dest,src1,immediate_value	dest = src1|immediate_value*	ori r3,r1,9	r3 = r1|9	r3 = 2|9 =B
xori	dest,src1,immediate_value	dest = src1^immediate_value*	xori r3,r1,3	r3 = r1^2	r3 = 2^3 =1
compi	dest,src1,immediate_value	dest = 1 if (src1==immediate_value*) else 0	compi r3,r5,2 compi r3,r1,2	r3 =(8==2)?1:0 r3 = (2==2)?1:0	r3 = 0 r3 = 1
slti	dest,src1,immediate_value	dest = 1 if (src1<=immediate_value*) else 0	slti r3,r5,2 slti r3,r5,10	r3 =(8<2)?1:0 r3 = (8<A)?1:0	r3 = 0 r3 = 1
mov	dest,src	dest=src	mov r3,r5	r3 = r5	r3 = 8
movi	dest,immediate_value	dest=immediate_value*	movi r3,10	r3 = A	r3 = A
sw	src1,src2,offset	datamem(src2+offset) =src1	sw r3,r5,4	datamem(r5+4) =r3	datamem(8+4) =6
swi	immediate_value,src,offset	datamem(src+offset) =immediate_value*	swi 11,r5,5	datamem(r5+5) =B	datamem(8+5) =B
lw	dest,src,offset	dest = datamem(src+offset)	lw r3,r5,9	r3 = datamem(r5+9)	r3 = datamem(8+9)
beq	src1,src2,target	PC = target if (src1==src2) else PC+1	beq r5,r1,target_name beq r4,r1,target_name	PC =(r5==r1)?target_addr:PC+1 PC = (r4==r1)?target_addr:PC+1	PC =PC+1 PC = target_addr
Other macros					
jmp	beq src,src,target	PC = target	Immdeiate_vale* is sign extended to 64 bits. All values while writing the assemble code is in decimal.		
swo	add a,src1,src2 sw src3,a,0	a = src1+src2 datamem(src1+src2) = src3			