Opcode : Beq: 4, Bne: 5, J: 2, Lw: 35

Register convention: S0: 16, T0: 8

## 1.1 0x8D100123：

1. 2’s complement integer: -0x72EFFEDD （-不能缺）
   因为负值→ 1000 1101 0001 0000 0000 0001 0010 0011
   所以：取反+1：0111 0010 1110 1111 1111 1110 1101 1101

2. sign and magnitude: -0x0D100123

3. IEEE single precision FP: -(1.00100000000000100100011)two*$2^{-101}$
   1000 1101 0001 0000 0000 0001 0010 0011
   -(1.00100000000000100100011)$_2$ * $2^{(26-127)}$
   “1.” is defult, not include in the part of mantissa.

4. LW $T0, 291($S0)

```
 1000 11 01 000 1 0000 0000 0001 0010 0011
    Op      rs      rt          imm
    35       8      16          256+32+3=291
```

## 1.2 IEEE single precision FP: -12.3

```
  1100.01001100110011001100*2^0
  1.10001001100110011001100*2^3        3+127 = 130
  1 1000 0010 10001001100110011001100
1 100 0 001 0 100 0100 1100 1100 1100 1100
  C       1   4    4    C    C    C    C
```

## 1.3

| $A | $B | $C=$A+$B | CF | OF | $D after<br>Slt $D, $A,$B | $D after<br>Sltu $D, $A,$B |
|---|---|---|---|---|---|---|
| 121 | 107 | E4 | 0 | 1 | 0 | 0 |
| 98 | -112 | F2 | 0 | 0 | 0 | 1 |
| -87 | -76 | 5D | 1 | 1 | 1 | 1 |
| -67 | 123 | 38 | 1 | 0 | 1 | 0 |

```
121:    0111 1001        98:   0110 0010          87:   0101 0111          67:   0100 0011
107:    0110 1011       112:   0111 0000         -87:   1010 1001         -67:   1011 1101
------------------     -112:   1001 0000          76:   0100 1100         123:   0111 1011
        1110 0100      -----------------         -76    1011 0100        ------------------
                               1111 0010               10101 1101               10011 1000
```

二、

| | | | |
|---|---|---|---|
| 1000 | R1: Beq $s0, $t0, R1 | 000100 10000 01000 1111 1111 1111 1111 | 0x1208FFFF |
| 1004 | Sub $s0, $s1, $s2 | 000000 10001 10010 10000 00000 100010 | 0x02328022 |
| 1008 | Bne $s1, st1, R1 | 000101 10001 01001 1111 1111 1111 1101 | 0x1629FFFD |
| 1012 | Lw $t0, -124($s0) | 100011 10000 01000 1111 1111 1000 0100 | 0x8E08FF84 |
| 1016 | J R1 | 000010 00000 00000 0000 0000 1111 1010 | 0x080000FA |
| | | | |

PC + 4 + imm = PC imm = -4      0000 0000 0000 0100→1111 1111 1111 1100 →shift 2 bits right
PC+4 + imm =PC-8 imm = -12      0000 0000 0000 1100→1111 1111 1111 0100→ shift 2 bits right
124 → 64 + 32+ 16 + 8+4 → 0000 0000 0111 1100 → 1111 1111 1000 0100
1000 → 1111 101 000→ 0011111010

三、

| | | |
|---|---|---|
| Ble $r1, $r2, R1 | If ($r1<=$r2) goto R1 | slt $at,$r2,$r1<br>beq $at,$zero,R1 |
| Bge $r1, $r2, R1 | If ($r1>$r2) goto R1 | slt $at,$r2,$r1<br>bne $at,$zero,R1 |
| Mov $r, BIG | $r = BIG in 2's complement format | lui $r, BIG_Hi + adj<br>ori $r, $r, BIG_Lo |
| Sne $r1, $r2, $r3 | $r1 = ($r2!=$r3) ? 1:0 | sub $at,$r2,$r3<br>sltu $r1,$zero,$at |

四、

```
int    abssum(int *p, int n )
{   int    i,m=p[0];
     For (i=1; i<n; i++) {
              If (p[i] >0)    m+=p[i];    else m-=p[i];
     }
     Return m.
   }
```

```
# i->$t0,         m->$t1
abssum: lw    $t1,0($a0)       #get the first value of m from parameter
        addi      $t0,$zero,1           #set the initial value i = 1
for:  slt    $t2,$t0,$a1
        beq $t2,$zero,Exit            #i !<n
        sll    $t2,$t0,2                   #i*4
        add $t2,$t2,$a0          #p[i]    address =    p0+ i*4
        lw    $t2,0($t2)         #p[i]
        slt    $t3,$t2,$zero            #p[i]<0
        bne $t3,$zero,R1
        add $t1,$t1,$t2         #m+=p[i]
        j      R2
R1:  sub $t1,$t1,$t2       #m-=p[i]
R2:  addi      $t0,$t0,1                #i++
        j      for
Exit:      add $v0,$t1,$zero
        jr     $ra
```

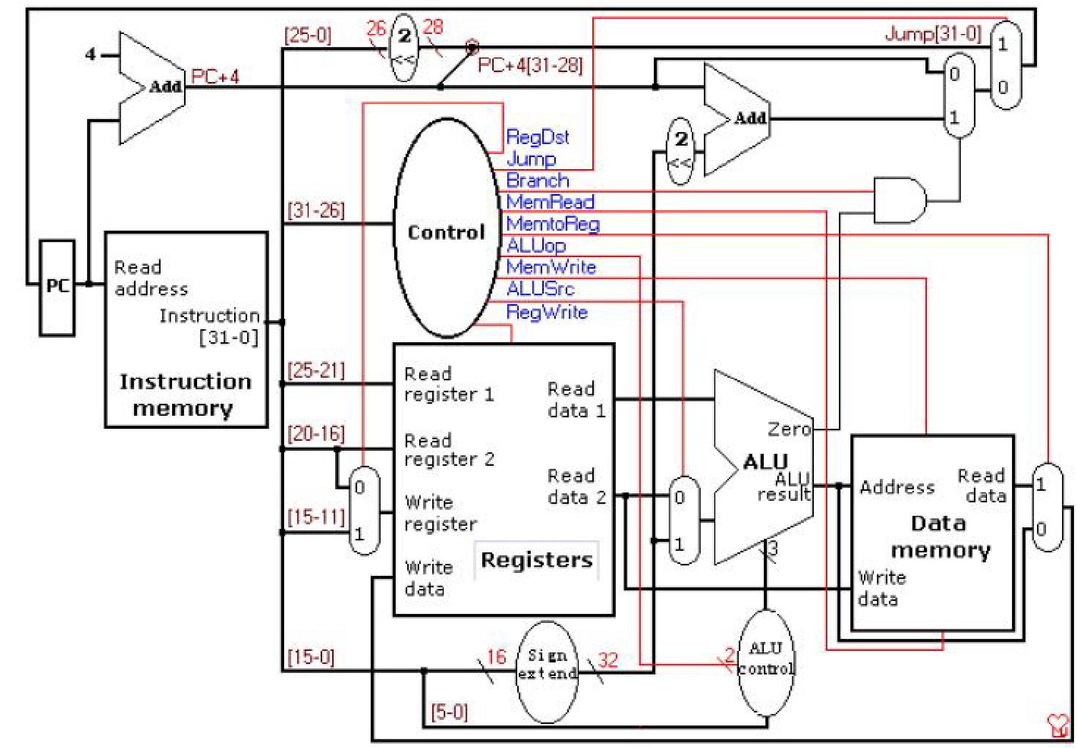
五、BNE $r1, $r2, label 实现

1.OP:6, $r1:5, $r2:5,  Label:16

2.ALU 的 zero 加一 zero 非，连一多路选择器 MUX 二选一，由新控制信号：bne 控制

3.Branch=1,ALUop=01,bne(New)=1

其余均为 0。



4
Add
PC+4
[25-0]
26
2
<<
28
PC+4[31-28]
Jump[31-0]
1
0
0
1
Add
2
<<
RegDst
Jump
Branch
MemRead
MemtoReg
ALUop
MemWrite
ALUSrc
RegWrite
Control
[31-26]
PC
Read address
Instruction [31-0]
Instruction memory
[25-21]
[20-16]
[15-11]
0
1
Read register 1
Read register 2
Write register
Write data
Read data 1
Read data 2
Registers
0
1
Zero
ALU
ALU result
3
Address
Read data
Data memory
Write data
1
0
[15-0]
16
Sign extend
32
2
ALU control
[5-0]