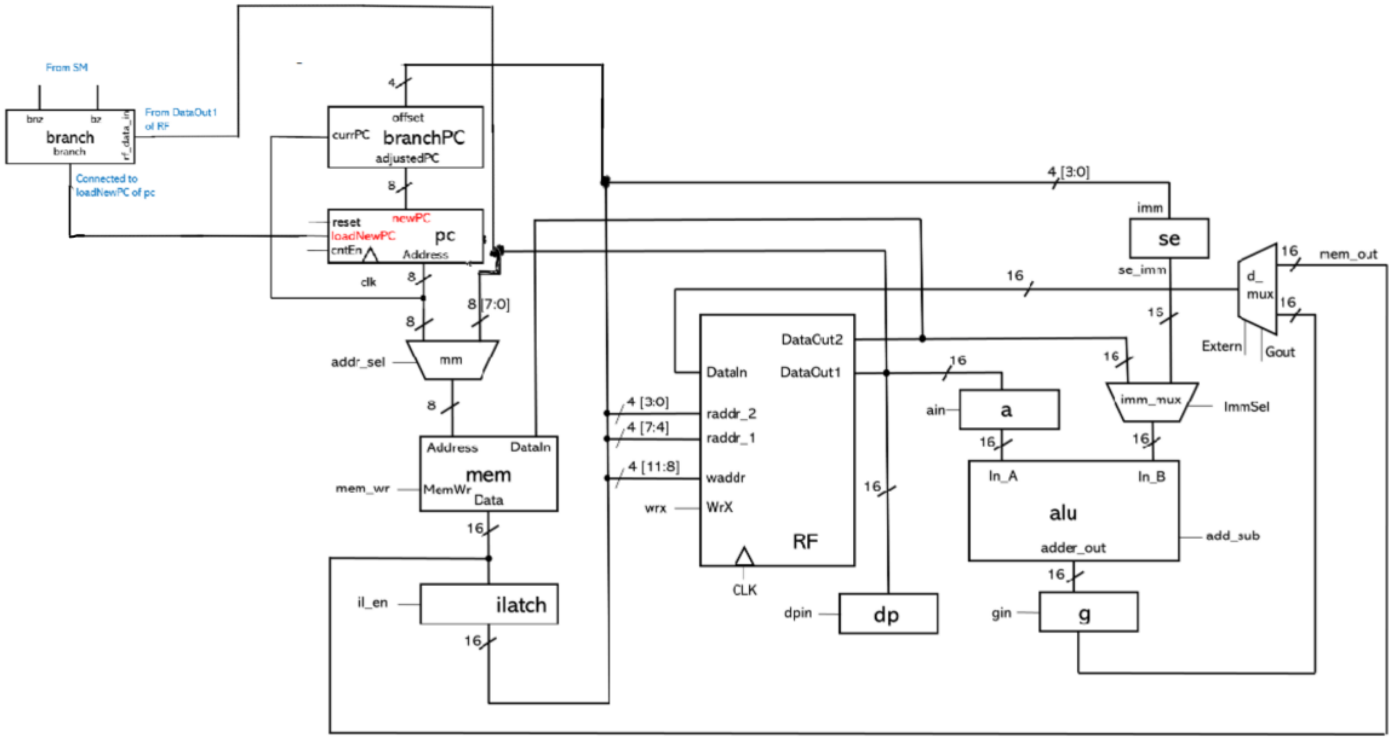
i. 1001 - bz
1010 - bnz

ii.

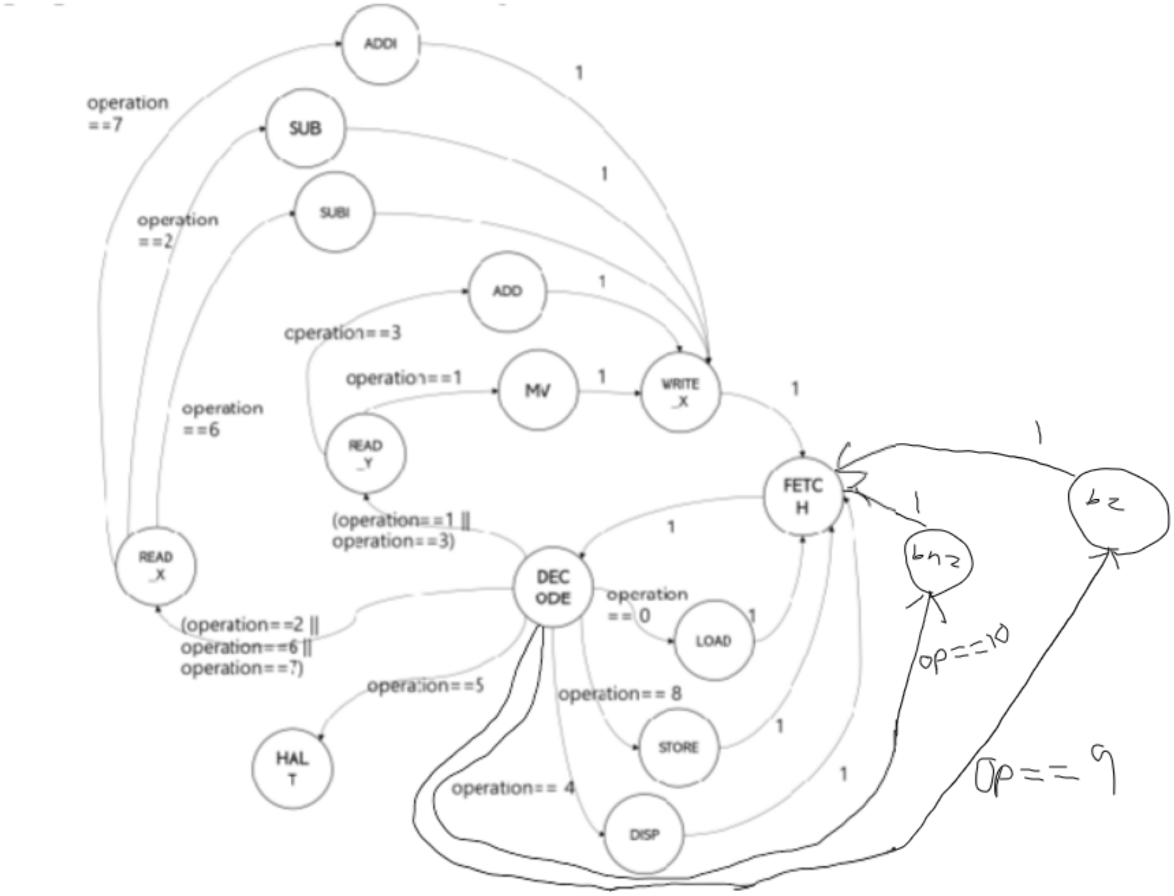


```
iii. module l6_branch(
          input bnz, /* asserted if the instruction is bnz */
          input bz,  /* asserted if the instruction is bz */
          input [15:0] rf_data_in,
          output reg branch  /* output: assert if the branch condition is fulfilled */
);

  /* TODO #2: complete the following */
  always@(*)
  begin
    if(bnz==1)  /* if the instruction is bnz? */
        if(rf_data_in==0) branch=0;
        else branch=1;
    else if (bz==1) /* if the instruction is bz */
        if(rf_data_in==0) branch=1;
        else branch=0;
    else /* if the instruction is not a branch-related one? */
        branch=0;
  end

endmodule
```

iv.



V.

Higher level pseudo code

```
2444 // sub r4, r4,r4; r4= number
2333 // sub r3,r3,r3; r3=count
2222 // sub r2,r2,r2; r2 = address
2111 // sub r1, r1,r1; r1=accum=0
9012 // branch to dis (skip two instruction) if r1=0
7115 // add 5 into r1
3111 // multiply r1*2= 10
7226 // add 6 into r2
3222 // mult r2 by 2 (12)
3222 // mult r2 by 2 (24)
3222 // mult r2 by 2 (48)
```

```
0320 // count = <value at address>
//begin while loop
7221 // add 1 to r2 (address)
0420 // number = <value at address>
3114 // add r1, r1, r4; accum +=number
6331 // subi r3, r3, 1
A03C //bnz; branch up 4 if r3 !=0
//end while loop
4010 // displayr1/accum
5000 //HALT!
```

Short code:

```
2111 // sub r1, r1, r1
9012 // branch to dis (skip two instruction) if r1=0
7115 // add 5 into r1
3111 // multiply r1*2= 10
4010 // display r1
5000 // HALT!
```