IV. Appendix

```
module labproject(clk, reset, w, A, B, currState, nextState, c, zf, sf, cf);

	//inputs
	input clk, reset;
	input [2:0]w; //opcode 3 bit
	input [3:0]A; //inputA 4 bit
	input [3:0]B; //inputB 4 bit

	//outputs
	output reg zf, sf, cf; //flags 1 bit
	output reg [3:0]c; // output 4 bit
	output reg [2:0] currState, nextState;

	reg carryBit; //1 bit to carry around the carry bit
	reg tc3; //to hold temporary c3 value for sub cope code
	reg [3:0]temp, tempB; //1 bit to hold borrowed 1 for Sub op code, temp to modify input A's values

	parameter	reset1 = 3'b000,
				xnor1 = 3'b001,
				secondBit = 3'b101,
				thirdBit = 3'b110,
				sub1 = 3'b010,
				nand1= 3'b011,
				add1 = 3'b100; //states


	always @(posedge clk, posedge reset)
	begin
```

```
if(reset == 1)
begin
        currState = reset1;
        nextState = reset1;
        c[3] = 0;
        c[2] = 0;
        c[1] = 0;
        c[0] = 0;
        cf = 0;
        zf = 0;
        sf = 0;
        carryBit = 0;
        temp[3] = 0;
        temp[2] = 0;
        temp[1] = 0;
        temp[0] = 0;
        tempB[3] = 0;
        tempB[2] = 0;
        tempB[1] = 0;
        tempB[0] = 0;
        tc3 = 0;

end
else
begin
        currState = nextState;
        case(currState)
                reset1: if (w[2] == 0 && w[1] == 0 && w[0] == 0)
                                begin
                                        nextState = reset1;
```

```
            //c[3] = 0;
            //c[2] = 0;
            //c[1] = 0;
            //c[0] = 0;
            cf = 0;
            zf = 0;
            sf = 0;
            carryBit = 0;
            temp[3] = 0;
            temp[2] = 0;
            temp[1] = 0;
            temp[0] = 0;
            tempB[3] = 0;
            tempB[2] = 0;
            tempB[1] = 0;
            tempB[0] = 0;
            tc3 = 0;

        end
        //-----------------------------------------
        else if (w[2] == 0 && w[1] == 0 && w[0] == 1) //xnor
part for 1st bit
        begin
            nextState = xnor1;
            c[3] = 0;
            c[2] = 0;
            c[1] = 0;
            c[0] = 0;

            if (A[0] == 1'b0 && B[0] == 1'b0)
```

```
            begin
              c[0] = 1'b1;
            end
            else if (A[0] == 1'b1 && B[0] == 1'b1)
            begin
              c[0] = 1'b1;
            end
            else
              c[0] = 1'b0;
        end
        //-----------------------------------------
        else if (w[2] == 0 && w[1] == 1 && w[0] == 0) //sub
part for 1st bit (A-B)
        begin
            nextState = sub1;

            c[3] = 0;
            c[2] = 0;
            c[1] = 0;
            c[0] = 0;

            temp[3] = A[3];
            temp[2] = A[2];
            temp[1] = A[1];
            temp[0] = A[0];

            tempB[3] = B[3];
            tempB[2] = B[2];
            tempB[1] = B[1];
            tempB[0] = B[0];
```

```
checking sign bit for A = 0 , B = 1

// A = 1 , B = 0

// A = 1 , B = 1

tempB[1] == 1'b0)

tempB[0] == 1'b0)

if (temp[3] == 1'b0 && tempB[3] == 1'b1) //
begin
  tc3 = 1'b0; // pos
end
else if (temp[3] == 1'b1 && tempB[3] == 1'b0)
begin
  tc3 = 1'b1; //neg
end
else if (temp[3] == 1'b1 && tempB[3] == 1'b1)
begin
        if (temp[2] == 1'b1 && tempB[2] ==
1'b0)
        begin
                tc3 = 1'b1; //neg
        end
        else if (temp[2] == tempB[2])
        begin
                if (temp[1] == 1'b1 &&
                begin
                        tc3 = 1'b1; //neg
                end
                else if (temp[1] == tempB[1])
                begin
                        if (temp[0] == 1'b1 &&
                        begin
                                tc3 = 1'b1; //neg
```

```
                    end
                    else
                        tc3 = 1'b0; //pos
                end
                else
                begin
                    tc3 = 1'b0; //pos
                end
            end
            else
            begin
                tc3 = 1'b0; //pos
            end
        end

        else if (temp[3] == 1'b0 && tempB[3] == 1'b0)
        begin
            if (temp[2] == 1'b1 && tempB[2] ==
1'b0)
            begin
                tc3 = 1'b0; //pos
            end
            else if (temp[2] == tempB[2])
            begin
                if (temp[1] == 1'b1 &&
tempB[1] == 1'b0)
                begin
                    tc3 = 1'b0; //pos
                end
                else if (temp[1] == tempB[1])
// A = 0 , B = 0
```

```
                begin
                        if (temp[0] == 1'b1 &&
tempB[0] == 1'b0)
                        begin
                                tc3 = 1'b0; //pos
                        end
                        else
                                tc3 = 1'b1; //neg
                end
                else
                begin
                        tc3 = 1'b1; //neg
                end
        end
        else
        begin
                tc3 = 1'b1; //neg
        end
end // sign bit check done

if (temp[3] == 1'b0 && tempB[3] == 1'b0)
begin
        if(tc3 == 0) // A = big Pos, B = small
Neg  (A-B)
        begin
                if (temp[0] == 1'b0 &&
tempB[0] == 1'b0)
                begin
                  c[0] = 1'b0;
                end
```

```
                else if (temp[0] == 1'b1 &&
tempB[0] == 1'b1)
                begin
                  c[0] = 1'b0;
                end
                else if (temp[0] == 1'b1 &&
tempB[0] == 1'b0)
                begin
                  c[0] = 1'b1;
                end
                else
                        if (temp[1] == 1'b1)
                        begin
                                temp[1] = 1'b0;
                                c[0] = 1'b1;
                        end
                        else if (temp[2] == 1'b1)
                        begin
                                temp[2] = 1'b0;
                                temp[1] = 1'b1;
                                c[0] = 1'b1;
                        end
        end
        else if(tc3 == 1) // A = small Pos, B =
big Neg (B-A)
        begin
                if (temp[0] == 1'b0 &&
tempB[0] == 1'b0)
                begin
                  c[0] = 1'b0;
                end
```

```
                else if (temp[0] == 1'b1 &&
tempB[0] == 1'b1)
                begin
                  c[0] = 1'b0;
                end
                else if (tempB[0] == 1'b1 &&
temp[0] == 1'b0)
                begin
                  c[0] = 1'b1;
                end
                else
                        if (tempB[1] == 1'b1)
                        begin
                                tempB[1] =
1'b0;
                                c[0] = 1'b1;
                        end
                        else if (B[2] == 1'b1)
                        begin
                                tempB[2] =
1'b0;
                                tempB[1] =
1'b1;
                                c[0] = 1'b1;
                        end
        end
end
else if (temp[3] == 1'b1 && tempB[3] == 1'b1)
begin
        if(tc3 == 0) // A = small Neg, B =  big
Pos (B-A)
        begin
```

```
                                        if (temp[0] == 1'b0 &&
tempB[0] == 1'b0)
                                        begin
                                          c[0] = 1'b0;
                                        end
                                        else if (temp[0] == 1'b1 &&
tempB[0] == 1'b1)
                                        begin
                                          c[0] = 1'b0;
                                        end
                                        else if (tempB[0] == 1'b1 &&
temp[0] == 1'b0)
                                        begin
                                          c[0] = 1'b1;
                                        end
                                        else
                                                if (tempB[1] == 1'b1)
                                                begin
                                                        tempB[1] =
1'b0;
                                                        c[0] = 1'b1;
                                                end
                                                else if (B[2] == 1'b1)
                                                begin
                                                        tempB[2] =
1'b0;
                                                        tempB[1] =
1'b1;
                                                        c[0] = 1'b1;
                                                end
                                end
```

```
else if(tc3 == 1) // A = big Neg, B =
small Pos (A-B)
begin
        if (temp[0] == 1'b0 &&
tempB[0] == 1'b0)
        begin
          c[0] = 1'b0;
        end
        else if (temp[0] == 1'b1 &&
tempB[0] == 1'b1)
        begin
          c[0] = 1'b0;
        end
        else if (temp[0] == 1'b1 &&
tempB[0] == 1'b0)
        begin
          c[0] = 1'b1;
        end
        else
                if (temp[1] == 1'b1)
                begin
                        temp[1] = 1'b0;
                        c[0] = 1'b1;
                end
                else if (temp[2] == 1'b1)
                begin
                        temp[2] = 1'b0;
                        temp[1] = 1'b1;
                        c[0] = 1'b1;
                end
end
```

```
                end
                //--
                else if ((temp[3] == 1'b0 && tempB[3] == 1'b1)
|| (temp[3] == 1'b1 && tempB[3] == 1'b0))
                begin
                        if (temp[0] == 1'b1 && tempB[0] ==
1'b1)
                        begin
                          c[0] = 1'b0;
                          carryBit = 1'b1;
                        end
                        else if (temp[0] == 1'b0 && tempB[0]
== 1'b0)
                        begin
                          c[0] = 1'b0;
                          carryBit = 1'b0;
                        end
                        else
                        begin
                          c[0] = 1'b1;
                          carryBit = 1'b0;
                        end
                end

        end
        //-----------------------------------------
        else if (w[2] == 0 && w[1] == 1 && w[0] == 1) //nand
part for 1st bit
        begin
                nextState = nand1;
```

```
            c[3] = 0;
            c[2] = 0;
            c[1] = 0;
            c[0] = 0;

            if (A[0] == 1'b1 && B[0] == 1'b1)
            begin
              c[0] = 1'b0;
            end
            else
              c[0] = 1'b1;
        end
        //-----------------------------------------
        else if (w[2] == 1 && w[1] == 0 && w[0] == 0) //add
part for 1st bit
        begin
            nextState = add1;

            c[3] = 0;
            c[2] = 0;
            c[1] = 0;
            c[0] = 0;

            if (A[0] == 1'b1 && B[0] == 1'b1)
            begin
              c[0] = 1'b0;
              carryBit = 1'b1;
            end
            else if (A[0] == 1'b0 && B[0] == 1'b0)
            begin
```

```
                    c[0] = 1'b0;
                    carryBit = 1'b0;
                end
                else
                begin
                    c[0] = 1'b1;
                    carryBit = 1'b0;
                end
            end
            //-----------------------------------------
//xnor1
xnor1:  if (w[2] == 0 && w[1] == 0 && w[0] == 0)
            begin
                nextState = reset1;
                c[3] = 0;
                c[2] = 0;
                c[1] = 0;
                c[0] = 0;
                cf = 0;
                zf = 0;
                sf = 0;
                carryBit = 0;
                temp[3] = 0;
                temp[2] = 0;
                temp[1] = 0;
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
```

```
                tc3 = 0;
            end
            else if (w[2] == 0 && w[1] == 0 && w[0] == 1) //xnor
part for 2nd bit
            begin
                nextState = secondBit;
                if (A[1] == 1'b0 && B[1] == 1'b0)
                begin
                  c[1] = 1'b1;
                end
                else if (A[1] == 1'b1 && B[1] == 1'b1)
                begin
                  c[1] = 1'b1;
                end
                else
                  c[1] = 1'b0;
            end
        //sub1:
        sub1: if (w[2] == 0 && w[1] == 0 && w[0] == 0)
                begin
                nextState = reset1;
                c[3] = 0;
                c[2] = 0;
                c[1] = 0;
                c[0] = 0;
                cf = 0;
                zf = 0;
                sf = 0;
                carryBit = 0;
                temp[3] = 0;
```

```
                temp[2] = 0;
                temp[1] = 0;
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
                tc3 = 0;
            end
            //-----------------------------------------
            else if (w[2] == 0 && w[1] == 1 && w[0] == 0) //sub
part for 2nd bit
            begin
                nextState = secondBit;
                if (temp[3] == 1'b0 && tempB[3] == 1'b0)
                begin
                    if(tc3 == 0) // A = big Pos, B = small
Neg (A-B)
                    begin
                        if (temp[1] == 1'b0 &&
tempB[1] == 1'b0)
                        begin
                          c[1] = 1'b0;
                        end
                        else if (temp[1] == 1'b1 &&
tempB[1] == 1'b1)
                        begin
                          c[1] = 1'b0;
                        end
                        else if (temp[1] == 1'b1 &&
tempB[1] == 1'b0)
                        begin
```

```
                c[1] = 1'b1;
            end
            else
                if (temp[2] == 1'b1)
                begin
                    temp[2] = 1'b0;
                    c[1] = 1'b1;
                end
    end
    else if(tc3 == 1) // A = small Pos, B =
big Neg (B-A)
    begin
        if (temp[1] == 1'b0 &&
tempB[1] == 1'b0)
        begin
            c[1] = 1'b0;
        end
        else if (temp[1] == 1'b1 &&
tempB[1] == 1'b1)
        begin
            c[1] = 1'b0;
        end
        else if (tempB[1] == 1'b1 &&
temp[1] == 1'b0)
        begin
            c[1] = 1'b1;
        end
        else
            if (tempB[2] == 1'b1)
            begin
                tempB[2] =
1'b0;
```

```
                                        c[1] = 1'b1;
                                    end
                            end
                    end
                    else if (temp[3] == 1'b1 && tempB[3] == 1'b1)
                    begin
                            if(tc3 == 0) // A = small Neg, B =  big
Pos (B-A)
                            begin
                                    if (temp[1] == 1'b0 &&
tempB[1] == 1'b0)
                                    begin
                                      c[1] = 1'b0;
                                    end
                                    else if (temp[1] == 1'b1 &&
tempB[1] == 1'b1)
                                    begin
                                      c[1] = 1'b0;
                                    end
                                    else if (tempB[1] == 1'b1 &&
temp[1] == 1'b0)
                                    begin
                                      c[1] = 1'b1;
                                    end
                                    else
                                            if (tempB[2] == 1'b1)
                                            begin
                                                    tempB[2] =
1'b0;
                                                    c[1] = 1'b1;
                                            end
```

```
                end
                else if(tc3 == 1) // A = big Neg, B =
small Pos (A-B)
                begin
                        if (temp[1] == 1'b0 &&
tempB[1] == 1'b0)
                        begin
                          c[1] = 1'b0;
                        end
                        else if (temp[1] == 1'b1 &&
tempB[1] == 1'b1)
                        begin
                          c[1] = 1'b0;
                        end
                        else if (temp[1] == 1'b1 &&
tempB[1] == 1'b0)
                        begin
                          c[1] = 1'b1;
                        end
                        else
                                if (temp[2] == 1'b1)
                                begin
                                        temp[2] = 1'b0;
                                        c[1] = 1'b1;
                                end
                end
        end
        //--
        else if ((temp[3] == 1'b0 && tempB[3] == 1'b1)
|| (temp[3] == 1'b1 && tempB[3] == 1'b0))
        begin
```

```
if (temp[1] == 1'b1 && tempB[1] ==
1'b1)
begin
        if(carryBit==1)
        begin
          c[1] = 1'b1;
          carryBit = 1'b1;
        end
        else
        begin
                c[1] = 1'b0;
                carryBit = 1'b1;
        end
end
else if (temp[1] == 1'b0 && tempB[1]
== 1'b0)
begin
        if(carryBit==1)
        begin
          c[1] = 1'b1;
          carryBit = 1'b0;
        end
        else
        begin
                c[1] = 1'b0;
                carryBit = 1'b0;
        end
end
else
begin
        if(carryBit==1)
```

```
                                        begin
                                          c[1] = 1'b0;
                                          carryBit = 1'b1;
                                        end
                                        else
                                        begin
                                                c[1] = 1'b1;
                                                carryBit = 1'b0;
                                        end
                                end
                        end
                        //--
                end

//nand1:
nand1:  if (w[2] == 0 && w[1] == 0 && w[0] == 0)
                        begin
                                nextState = reset1;
                                c[3] = 0;
                                c[2] = 0;
                                c[1] = 0;
                                c[0] = 0;
                                cf = 0;
                                zf = 0;
                                sf = 0;
                                carryBit = 0;
                                temp[3] = 0;
                                temp[2] = 0;
                                temp[1] = 0;
```

```
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
                tc3 = 0;
            end
            //----------------------------------------
            else if (w[2] == 0 && w[1] == 1 && w[0] == 1) //nand
part for 2nd bit
            begin
                nextState = secondBit;
                if (A[1] == 1'b1 && B[1] == 1'b1)
                begin
                  c[1] = 1'b0;
                end
                else
                  c[1] = 1'b1;
            end

        //add1:
        add1: if (w[2] == 0 && w[1] == 0 && w[0] == 0)
            begin
                nextState = reset1;
                c[3] = 0;
                c[2] = 0;
                c[1] = 0;
                c[0] = 0;
                cf = 0;
                zf = 0;
```

```
                sf = 0;
                carryBit = 0;
                temp[3] = 0;
                temp[2] = 0;
                temp[1] = 0;
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
                tc3 = 0;
            end
        //----------------------------------------
          else if (w[2] == 1 && w[1] == 0 && w[0] == 0) //add part for
2nd bit
            begin
                nextState = secondBit;
                if (A[1] == 1'b1 && B[1] == 1'b1)
                begin
                    if(carryBit==1)
                    begin
                      c[1] = 1'b1;
                      carryBit = 1'b1;
                    end
                    else
                    begin
                        c[1] = 1'b0;
                        carryBit = 1'b1;
                    end
                end
```

```
                else if (A[1] == 1'b0 && B[1] == 1'b0)
                begin
                        if(carryBit==1)
                        begin
                          c[1] = 1'b1;
                          carryBit = 1'b0;
                        end
                        else
                        begin
                                c[1] = 1'b0;
                                carryBit = 1'b0;
                        end
                end
                else
                begin
                        if(carryBit==1)
                        begin
                          c[1] = 1'b0;
                          carryBit = 1'b1;
                        end
                        else
                        begin
                                c[1] = 1'b1;
                                carryBit = 1'b0;
                        end
                end
        end

secondBit:      if (w[2] == 0 && w[1] == 0 && w[0] == 0)
                begin
```

```
                nextState = reset1;
                c[3] = 0;
                c[2] = 0;
                c[1] = 0;
                c[0] = 0;
                cf = 0;
                zf = 0;
                sf = 0;
                carryBit = 0;
                temp[3] = 0;
                temp[2] = 0;
                temp[1] = 0;
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
                tc3 = 0;
        end
        //----------------------------------------
        else if (w[2] == 0 && w[1] == 0 && w[0] == 1)
//xnor part for 3rd bit
        begin
                nextState = thirdBit;
                if (A[2] == 1'b0 && B[2] == 1'b0)
                begin
                  c[2] = 1'b1;
                end
                else if (A[2] == 1'b1 && B[2] == 1'b1)
                begin
```

```
                c[2] = 1'b1;
            end
            else
                c[2] = 1'b0;
    end
    //----------------------------------------
    else if (w[2] == 0 && w[1] == 1 && w[0] == 0)
//sub part for 3rd bit (A-B)
    begin
            nextState = thirdBit;
            //---
            if (temp[3] == 1'b0 && tempB[3] ==
1'b0)
            begin
                    if(tc3 == 0) // A = big Pos, B =
small Neg (A-B)
                    begin
                            if (temp[2] == 1'b0 &&
tempB[2] == 1'b0)
                            begin
                              c[2] = 1'b0;
                            end
                            else if (temp[2] == 1'b1
&& tempB[2] == 1'b1)
                            begin
                              c[2] = 1'b0;
                            end
                            else if (temp[2] == 1'b1
&& tempB[2] == 1'b0)
                            begin
                              c[2] = 1'b1;
                            end
```

```
                end
                else if(tc3 == 1) // A = small
Pos, B = big Neg (B-A)
                begin
                        if (temp[2] == 1'b0 &&
tempB[2] == 1'b0)
                        begin
                          c[2] = 1'b0;
                        end
                        else if (temp[2] == 1'b1
&& tempB[2] == 1'b1)
                        begin
                          c[2] = 1'b0;
                        end
                        else if (tempB[2] ==
1'b1 && temp[2] == 1'b0)
                        begin
                          c[2] = 1'b1;
                        end
                end
        end
        else if (temp[3] == 1'b1 && tempB[3]
== 1'b1)
        begin
                if(tc3 == 0) // A = small Neg, B
=  big Pos (B-A)
                begin
                        if (temp[2] == 1'b0 &&
tempB[2] == 1'b0)
                        begin
                          c[2] = 1'b0;
                        end
```

```
                else if (temp[2] == 1'b1
&& tempB[2] == 1'b1)
                begin
                  c[2] = 1'b0;
                end
                else if (tempB[2] ==
1'b1 && temp[2] == 1'b0)
                begin
                  c[2] = 1'b1;
                end

            end
            else if(tc3 == 1) // A = big Neg,
B = small Pos (A-B)
            begin
                if (temp[2] == 1'b0 &&
tempB[2] == 1'b0)
                begin
                  c[2] = 1'b0;
                end
                else if (temp[2] == 1'b1
&& tempB[2] == 1'b1)
                begin
                  c[2] = 1'b0;
                end
                else if (temp[2] == 1'b1
&& tempB[2] == 1'b0)
                begin
                  c[2] = 1'b1;
                end
            end
        end
```

```
//--
else if ((temp[3] == 1'b0 && tempB[3] == 1'b1) || (temp[3] == 1'b1 && tempB[3] == 1'b0))

begin
        if (temp[2] == 1'b1 && tempB[2] == 1'b1)

        begin
                if(carryBit==1)
                begin
                  c[2] = 1'b1;
                  carryBit = 1'b1;
                end
                else
                begin
                        c[2] = 1'b0;
                        carryBit = 1'b1;
                end
        end
        else if (temp[2] == 1'b0 && tempB[2] == 1'b0)

        begin
                if(carryBit==1)
                begin
                  c[2] = 1'b1;
                  carryBit = 1'b0;
                end
                else
                begin
                        c[2] = 1'b0;
                        carryBit = 1'b0;
                end
```

```
                end
                else
                begin
                        if(carryBit==1)
                        begin
                          c[2] = 1'b0;
                          carryBit = 1'b1;
                        end
                        else
                        begin
                                c[2] = 1'b1;
                                carryBit = 1'b0;
                        end
                end
        end
        //---
end
//----------------------------------------
else if (w[2] == 0 && w[1] == 1 && w[0] == 1)
```

//nand part for 3rd bit

```
begin
        nextState = thirdBit;
        if (A[2] == 1'b1 && B[2] == 1'b1)
        begin
          c[2] = 1'b0;
        end
        else
          c[2] = 1'b1;
end
//----------------------------------------
```

```
//add part for 3rd bit
                                        else if (w[2] == 1 && w[1] == 0 && w[0] == 0)
                                        begin
                                                nextState = thirdBit;
                                                if (A[2] == 1'b1 && B[2] == 1'b1)
                                                begin
                                                        if(carryBit==1)
                                                        begin
                                                          c[2] = 1'b1;
                                                          carryBit = 1'b1;
                                                        end
                                                        else
                                                        begin
                                                                c[2] = 1'b0;
                                                                carryBit = 1'b1;
                                                        end
                                                end
                                                else if (A[2] == 1'b0 && B[2] == 1'b0)
                                                begin
                                                        if(carryBit==1)
                                                        begin
                                                          c[2] = 1'b1;
                                                          carryBit = 1'b0;
                                                        end
                                                        else
                                                        begin
                                                                c[2] = 1'b0;
                                                                carryBit = 1'b0;
                                                        end
                                                end
```

```
                else
                begin
                        if(carryBit==1)
                        begin
                          c[2] = 1'b0;
                          carryBit = 1'b1;
                        end
                        else
                        begin
                                c[2] = 1'b1;
                                carryBit = 1'b0;
                        end
                end
        end

thirdBit:   if (w[2] == 0 && w[1] == 0 && w[0] == 0)
                begin
                        nextState = reset1;
                        c[3] = 0;
                        c[2] = 0;
                        c[1] = 0;
                        c[0] = 0;
                        cf = 0;
                        zf = 0;
                        sf = 0;
                        carryBit = 0;
                        temp[3] = 0;
                        temp[2] = 0;
                        temp[1] = 0;
```

```
                temp[0] = 0;
                tempB[3] = 0;
                tempB[2] = 0;
                tempB[1] = 0;
                tempB[0] = 0;
                tc3 = 0;
        end
        //----------------------------------------
        else if (w[2] == 0 && w[1] == 0 && w[0] == 1)
//xnor part for 4th bit
        begin
                nextState = reset1;
                if (A[3] == 1'b0 && B[3] == 1'b0)
                begin
                  c[3] = 1'b1;
                end
                else if (A[3] == 1'b1 && B[3] == 1'b1)
                begin
                  c[3] = 1'b1;
                end
                else
                  c[3] = 1'b0;

                if (c[3] == 0 && c[2] == 0 && c[1] ==
0 && c[0] == 0)//zero flag
                begin
                        zf = 1'b1;
                end
                else
                        zf = 1'b0;
```

```
                if (c[3] == 1) //sign flag
                begin
                        sf = 1'b1;
                end
                else
                        sf = 1'b0;
        end
        //----------------------------------------
        else if (w[2] == 0 && w[1] == 1 && w[0] == 0)
//sub part for 4th bit (A-B)
        begin

                nextState = reset1;
                c[3] = tc3;

                if (c[3] == 0 && c[2] == 0 && c[1] ==
0 && c[0] == 0)//zero flag
                begin
                        zf = 1'b1;
                end
                else
                        zf = 1'b0;

                if (c[3] == 1) //sign flag
                begin
                        sf = 1'b1;
                end
                else
                        sf = 1'b0;
                //----
                if (carryBit == 1) //carry flag
```

```
                begin
                        cf = 1'b1;
                end
                else
                        cf = 1'b0;
        end
        //----------------------------------------
        else if (w[2] == 0 && w[1] == 1 && w[0] == 1)
//nand part for 4th bit
        begin
                nextState = reset1;
                if (A[3] == 1'b1 && B[3] == 1'b1)
                begin
                  c[3] = 1'b0;
                end
                else
                  c[3] = 1'b1;

                if (c[3] == 0 && c[2] == 0 && c[1] ==
0 && c[0] == 0)//zero flag
                begin
                        zf = 1'b1;
                end
                else
                        zf = 1'b0;

                if (c[3] == 1) //sign flag
                begin
                        sf = 1'b1;
                end
                else
```

```
                    sf = 1'b0;


end
//----------------------------------------
else if (w[2] == 1 && w[1] == 0 && w[0] == 0)
```

//add part for 4th bit

```
begin
        nextState = reset1;
        if (A[3] == 1'b1 && B[3] == 1'b1)
        begin
                if(carryBit==1)
                begin
                  c[3] = 1'b1;
                  carryBit = 1'b1;
                end
                else
                begin
                        c[3] = 1'b0;
                        carryBit = 1'b1;
                end
        end
        else if (A[3] == 1'b0 && B[3] == 1'b0)
        begin
                if(carryBit==1)
                begin
                  c[3] = 1'b1;
                  carryBit = 1'b0;
                end
                else
                begin
```

```
                c[3] = 1'b0;
                carryBit = 1'b0;
            end
        end
        else
        begin
            if(carryBit==1)
            begin
              c[3] = 1'b0;
              carryBit = 1'b1;
            end
            else
            begin
                c[3] = 1'b1;
                carryBit = 1'b0;
            end
        end
        if (c[3] == 0 && c[2] == 0 && c[1] ==
0 && c[0] == 0)//zero flag
        begin
            zf = 1'b1;
        end
        else
            zf = 1'b0;
        //----
        if (c[3] == 1) //sign flag
        begin
            sf = 1'b1;
        end
        else
```

```
                    sf = 1'b0;
                //----
                if (carryBit == 1) //carry flag
                begin
                    cf = 1'b1;
                end
                else
                    cf = 1'b0;
            end

        endcase
    end

end

endmodule
```