DSD LAB

LAB 2

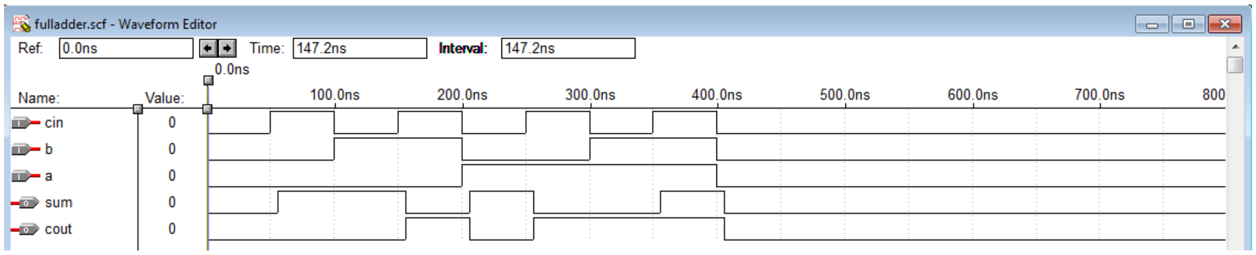
Write behavioral Verilog code to implement the following and simulate
1. Full adder
2. Four-bit adder/ subtractor
3. Single-digit BCD adder using a four-bit adder(s).

Q1)

Source Code:

```
module fulladder(a, b, cin, cout, sum);
input a, b, cin;
output cout, sum;

assign {cout,sum} = a + b + cin;
endmodule
```

Output Waveform:

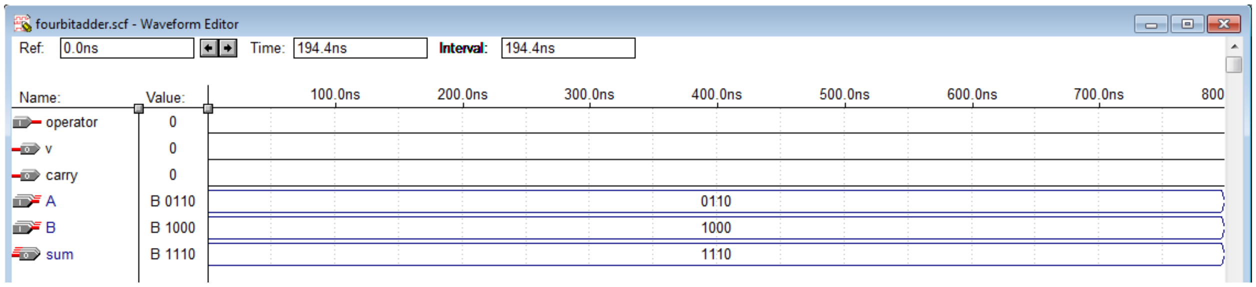


Q2)

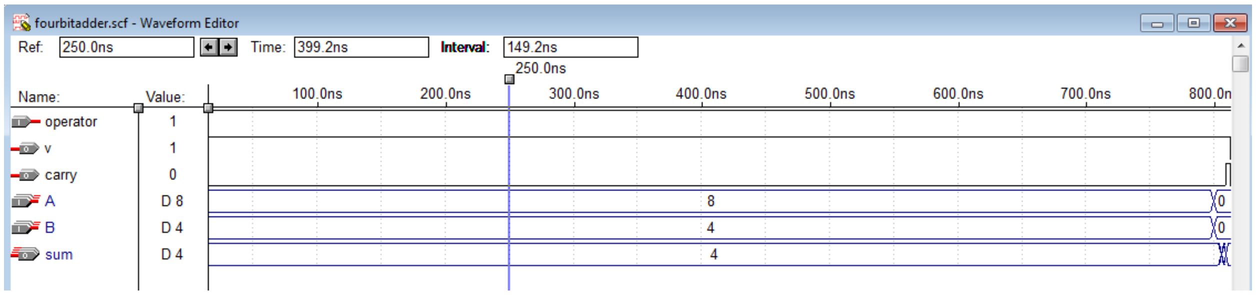
Source Code:

```
module fourbitadder(sum, carry, v, A, B, operator);
output [3:0] sum;
output carry;
output v;
input [3:0] A;
input [3:0] B;
input operator;
wire C0, C1, C2, C3, B0, B1, B2, B3;
assign B0 = B[0] ^ operator;
assign B1 = B[1] ^ operator;
assign B2 = B[2] ^ operator;
assign B3 = B[3] ^ operator;
assign carry = C3 ^ operator;
assign v = C3 ^ C2;
fulladder fa0(A[0], B0, operator, C0, sum[0]); // Least significant bit.
fulladder fa1(A[1], B1, C0, C1, sum[1]);
fulladder fa2(A[2], B2, C1, C2, sum[2]);
fulladder fa3(A[3], B3, C2, C3, sum[3]); // Most significant bit.
endmodule

module fulladder(a, b, cin, cout, sum);
input a, b, cin;
output cout, sum;

assign {cout,sum} = a + b + cin;
endmodule
```

Output Waveform:

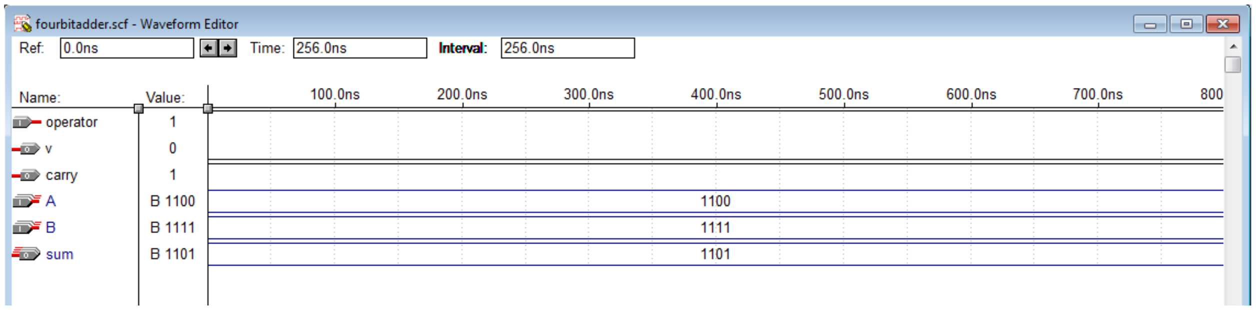


Name:	Value:
operator	0
v	0
carry	0
A	B 0110
B	B 1000
sum	B 1110



Ref: 250.0ns Time: 399.2ns Interval: 149.2ns

Name:	Value:
operator	1
v	1
carry	0
A	D 8
B	D 4
sum	D 4



Ref: 0.0ns Time: 256.0ns Interval: 256.0ns

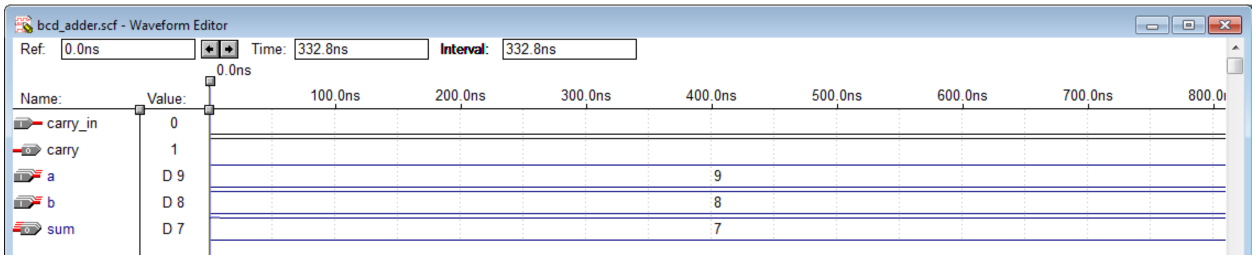
Name:	Value:
operator	1
v	0
carry	1
A	B 1100
B	B 1111
sum	B 1101

Q3)

Source Code:

```
module bcd_adder(a,b,carry_in,sum,carry);
    input [3:0] a,b;
    input carry_in;
    output [3:0] sum;
    output carry;
    reg [4:0] sum_temp;
    reg [3:0] sum;
    reg carry;

    always @ (a | b | carry_in)
    begin
        sum_temp = a + b + carry_in;
        if(sum_temp > 9)
            begin
            sum_temp = sum_temp + 6;
            carry = 1;
            sum = sum_temp[3:0];
            end
        else
            begin
            carry = 0;
            sum = sum_temp[3:0];
            end
    end
endmodule
```

Output Waveform:



Name:	Value:	
carry_in	0	
carry	1	
a	D 9	9
b	D 8	8
sum	D 7	7