DLD LAB ASSIGNMENT

EXPERIMENT 2:

1. HALF SUBTRACTOR:

- Design code-

```
module
 half_sub(a,b,d,br);
 input a,b;
 output br,d;
 assign {br,d}=a-b;
endmodule
```

- Testbench-

```
module test();
 reg a,b;
 wire d,br;
 half_sub uut(.a(a),.b(b),.d(d),.br(br));
 initial
  begin
   $dumpfile("half_sub.vcd");
   $dumpvars(1);
  end
 initial
  begin
   a=0; b=0; #100;
   a=0; b=1; #100;
   a=1; b=0; #100;
   a=1; b=1; #100;
  end
endmodule
```

- Output:

Inputs		Outputs	
A	B	Diff	Borrow
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

2. FULL SUBTRACTOR:

- Design code:

```
module full_sub(a,b,c,d,br);
 input a,b,c;
 output br,d;
 assign {br,d}=a-b-c;
endmodule
```

- TestBench:

```
module test();
 reg a,b,c;
 wire d,br;
 full_sub uut(.a(a),.b(b),.c(c),.d(d),.br(br));
 initial
  begin
   $dumpfile("full_sub.vcd");
   $dumpvars(1);
  end
 initial
  begin
   a=0; b=0; c=0; #100;
   a=0; b=0; c=1; #100;
   a=0; b=1; c=0; #100;
   a=0; b=1; c=1; #100;
   a=1; b=0; c=0; #100;
```

```
    a=1; b=0; c=1; #100;
    a=1; b=1; c=0; #100;
    a=1; b=1; c=1; #100;
  end
endmodule
```

- Output:

Inputs			Outputs	
A	B	$Borrow_{in}$	Diff	Borrow
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

3.HALF ADDER:

- Design code:

```
module
halfadder(a,b,s,c);
 input a,b;
 output s,c;
 assign {c,s}=a+b;
endmodule
```

- TestBench code:

```
module test();
 reg a,b;
 wire s,c;
 halfadder uut(.a(a),.b(b),.s(s),.c(c));
 initial
  begin
   $dumpfile("halfadder.vcd");
   $dumpvars(1);
  end
 initial
  begin
   a=0; b=0; #100;
```

```
    a=0; b=1; #100;
    a=1; b=0; #100;
    a=1; b=1; #100;
  end
endmodule
```

- Output:

Input		Output	
A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

4.FULL ADDER:

- Design code:

```
module full_adder(s,c0,a,b,c);
  output s,c0;
  input a,b,c;
  assign {c0,s}=a+b+c;
endmodule
```

- TestBench code:

```
module test();
  reg a,b,c;
  wire s,c0;
  full_adder uut(.a(a),.b(b),.c(c),.s(s),.c0(c0));
  initial
    begin
      $dumpfile("full_adder.vcd");
```

```
    $dumpvars(1);
  end
 initial
  begin
    a=0; b=0; c=0; #100;
    a=0; b=0; c=1; #100;
    a=0; b=1; c=0; #100;
    a=0; b=1; c=1; #100;
    a=1; b=0; c=0; #100;
    a=1; b=0; c=1; #100;
    a=1; b=1; c=0; #100;
    a=1; b=1; c=1; #100;
  end
endmodule
```

- Output:

INPUTS			OUTPUT	
A	B	C-IN	C-OUT	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1