4. **Design Verilog HDL to implement Binary Adder-Subtractor – Half and Full Adder, Half and Full Subtractor**

1. **Half Adder: Verilog code**

**module** halfadder(x,y,sum,carryout);

**input** x,y;

**output** sum,carryout;

**assign** sum = ((!x)&&y)||(x&&(!y));

**assign** carryout = (x&&y);

**endmodule**

**Test bench waveform code for half adder**

**module** TestModule;

**reg** x,y;

**wire** sum,carryout;

// Instantiate the Unit Under Test (UUT)

**halfadder** uut (.x(x),.y(y),.sum(sum),.carryout(carryout));

**initial begin**

**$dumpfile**("dump.vcd");

**$dumpvars**(1);

**end**

**initial begin**

// Initialize Inputs

x = 0;y = 0;

#5 x = 0;y = 1;

#5 x = 1;y = 0;

#5 x = 1;y = 1;

#5;

**$finish;**

**end**

**endmodule**

**b) Full Adder: Verilog code**

**module** fulladder(x,y,z,sum,carryout);

**input** x,y,z;

**output** sum,carryout;

**assign** sum = ((!x)&&(!y)&&z)||((!x)&&y&&(!z))||(x&&(!y)&&(!z))||(x&&y&&z);

**assign** carryout = (x&&y)||(x&&z)||(y&&z);

**endmodule**

**Test bench waveform code for full adder**

**module** TestModule;

**reg** x,y,z;

**wire** sum,carryout;

// Instantiate the Unit Under Test (UUT)

**fulladder** uut(.x(x),.y(y),.z(z),.sum(sum),.carryout(carryout));

**initial begin**

**$dumpfile**("dump.vcd");

**$dumpvars**(1);

**end**

**initial begin**

// Initialize Inputs

x = 0;y = 0;z = 0;

#5 x = 0;y = 0;z = 1;

#5 x = 0;y = 1;z = 0;

#5 x = 0;y = 1;z = 1;

#5 x = 1;y = 0;z = 0;

#5 x = 1;y = 0;z = 1;

#5 x = 1;y = 1;z = 0;

#5 x = 1;y = 1;z = 1;

#5;

**$finish;**

**end**

**endmodule**

**c) Half Subtractor: Verilog code**

**module** halfsub(x,y,diff,brr);

**input** x,y;

**output** diff,brr;

**assign** diff = ((!x)&&y)||(x&&(!y));

**assign** brr = ((!x)&&y);

**endmodule**

**Test bench waveform code half subtractor**

**module** TestModule;

**reg** x,y;

**wire** diff,brr;

// Instantiate the Unit Under Test (UUT)

**halfsub** uut (.x(x),.y(y),.diff(diff),.brr(brr));

**initial begin**

**$dumpfile**("dump.vcd");

**$dumpvars**(1);

**end**

**initial begin**

// Initialize Inputs

x = 0;y = 0;

#5 x = 0;y = 1;

#5 x = 1;y = 0;

#5 x = 1;y = 1;

#5;

**$finish;**

**end**

**endmodule**

**d) Full Subtractor: Verilog code**

**module** fullsub(x,y,z,diff,brr)**;**

**input** x,y,z;

**output** diff,brr;

**assign diff =** ((!x)&&(!y)&&z)||((!x)&&y&&(!z))||(x&&(!y)&&(!z))||(x&&y&&z);

**assign brr =** ((!x)&&y)||((!x)&&z)||(y&&z);

**endmodule**

**Test bench waveform code full subtractor**

**module** TestModule;

**reg** x,y,z;

**wire** diff,brr;

**//** Instantiate the Unit Under Test (UUT)

**fullsub** uut(.x(x),.y(y),.z(z),.diff(diff),.brr(brr));

**initial begin**

**$dumpfile**("dump.vcd");

**$dumpvars(1);**

**end**

**initial begin**

**//** Initialize Inputs

x = 0;y = 0;z = 0;

#5 x = 0;y = 0;z = 1;

#5 x = 0;y = 1;z = 0;

#5 x = 0;y = 1;z = 1;

#5 x = 1;y = 0;z = 0;

#5 x = 1;y = 0;z = 1;

#5 x = 1;y = 1;z = 0;

#5 x = 1;y = 1;z = 1;

#5;

**$finish;**

**end**

**endmodule**