```
module multiplier4bit(output [7:0]out,input [3:0]a,b);
wire [3:0]w1,w2,w3;
wire [4:0]s1,s2,w0;
mul in0(w0[3:0],a,b[0]);
assign w0[4]=1'b0;
mul in1(w1,a,b[1]);
mul in2(w2,a,b[2]);
mul in3(w3,a,b[3]);
four gh1(s1,w1,w0[4:1]);
four gh2(s2,w2,s1[4:1]);
four gh3(out[7:3],w3,s2[4:1]);
assign out[0]=w0[0];
assign out[1]=s1[0];
assign out[2]=s2[0];
endmodule
module mul(output [3:0]a,input [3:0]w,input b);
assign a[0]=b&w[0];
assign a[1]=b&w[1];
assign a[2]=b&w[2];
assign a[3]=b&w[3];
endmodule
module four(output [4:0]out,input [3:0]a,b);
wire w1,w2,w3;
fa i1(w1,out[0],a[0],b[0],1'b0);
fa i2(w2,out[1],a[1],b[1],w1);
fa i3(w3,out[2],a[2],b[2],w2);
fa i4(out[4],out[3],a[3],b[3],w3);
endmodule
module fa(output ca,s,input a,b,c);
assign s=a^b^c;
assign ca= a&b | b&c | a&c;
```

```
endmodule
```

Testbench

```
module tb;
reg [3:0]a,b;
wire [7:0]out;
multiplier4bit inst(out,a,b);
integer i;
initial begin
for(i=0;i<256;i=i+1) begin
{a,b}=i;
#10;
end
end
endmodule
```

Testbench

```
module tb;
reg [3:0]a,b;
wire [7:0]out;
multiplier4bit inst(out,a,b);
integer i;
initial begin
for(i=0;i<256;i=i+1) begin
{a,b}=i;
#10;
end
end
endmodule
```