

```

module signed_mul(X, Y, Z);
input signed [3:0] X, Y;
output signed [7:0] Z;
reg signed [7:0] Z;
reg [1:0] temp;
integer i;
reg E1;
reg [3:0] Y1;
always @(X, Y)
begin
Z = 8'b00000000;
E1 = 1'b0;
for (i = 0; i < 4; i = i + 1)
begin temp = {X[i], E1};
Y1 = - Y;
case (temp)
2'd2: Z [7:4] = Z [7: 4] + Y1;
2'd1: Z [7:4] = Z [7: 4] + Y;
default: begin end
endcase
Z = Z >> 1;
Z[7] = Z[6];
E1 = X[i];
end
if (Y == 4'd8)
begin
Z = Z;
end
end
endmodule

```

```
module tb;
reg signed[3:0]X,Y;
wire signed[7:0]Z;
signed_mul inst(X,Y,Z);
initial begin
X=-4;
Y=2;
# 10;
end
endmodule
```