```
module booth (X, Y, Z,en);
input signed [31:0] X, Y; //Declaratioon of 32 bit inputs
input en; //Declaration of enable
output signed [63:0] Z; //Declaration of output
reg signed [63:0] Z;
reg [1:0] temp;
integer i;
reg R1;
reg [31:0] Y1;

always @ (X, Y,en)
begin
Z= 64'd0;
R1 = 1'd0;
for (i = 0; i < 32; i = i + 1)
begin
temp = {X[i], R1}; //Concatenation of Q,Qn-1
Y1 = - Y; //Y1 is the 2â€™ complement of Y
Â Â
case (temp)
2'd2 : Z [63 : 31] = Z [63 : 31] + Y1; // If temp == 10 then A : A-Y
2'd1 : Z [63 : 31] = Z [63 : 31] + Y; // If temp == 10 then A : A+Y
default : begin end // In other case just right shfits the register
endcase
Z = Z >> 1; Â Â //Shift right operation
Z[63] = Z[62];
R! = X[i];
end

Â Â if (Y == 32'd64)Â Â //If Y = 1000; then Y1 = 1000 (-8 not 8, because Y1 is 4 bits only). so we have to compliment the res
Â Â begin
Z = - Z;
end
end
endmodule
```