

# Download File PDF Binary Multiplier Verilog Code Book

#Jenny



Finally I get this ebook, thanks for all these I can get now!

#Rio



Cool! I'am really happy

#Markus Jensen



I did not think that this would work, my best friend showed me this website, and it does! I get my most wanted eBook

#Hun Tsu



wtf this great ebook for free?!

#Che Salsa



My friends are so mad that they do not know how I have all the high quality ebook which they do not!

#Diego Butler



so many fake sites. this is the first one which worked! Many thanks

## TASK 1: 8-bit Verilog Code for Booth's Multiplier

```
module multiplier8proc, busy, nr, np, cin, stacti;
output [15:0] prod;
output busy;
input [7:0] nr, np;
input cin, stacti;
reg [7:0] A, B, M;
reg [1:0] count;

wire [7:0] num, difference;

always @posedge cin
begin
if (stacti) begin
M <= 0'b0;
N <= nr;
C_in <= cin;
count <= 8'b0;
end
else begin
case ({0,0}, 2'b)
2'b0_0 : (A, B, C_in) <= (sum[7], num, 0);
2'b0_1 : (A, B, C_in) <= (difference[7], difference, 0);
default : (A, B, C_in) <= (A[7], A, 0);
endcase
count <= count + 1'b1;
end
end

sis adder (num, A, M, 1'b0);
sis subtractor (difference, A, ~M, 1'b1);
assign prod = (A, B);
assign busy = (count < 8);
endmodule

//The following is an alt.
//It is an adder, but capable of subtraction.
//Recall that subtraction means adding the two's complement--
//A - B = A + (-B) = A + (inverted B + 1)
//The 1 will be coming in as cin (carry-in)
module altadder, A, B, cin;
output [7:0] out;
input [7:0] A;
input [7:0] B;
input cin;
assign out = A + B + cin;
endmodule
```

[Download PDF version of :](#)  
**Binary Multiplier Verilog Code Book**