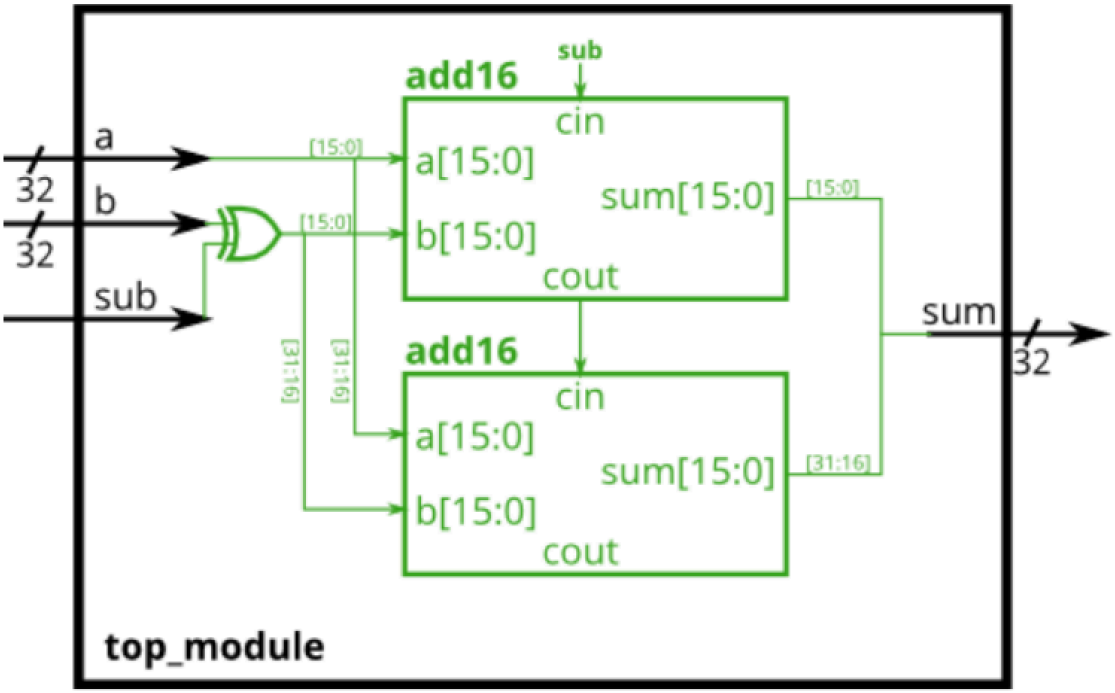
KARTHICK RAJA B
126004121

Given:

You have to design a 16-bit adder module, which you need to instantiate twice:

```
module add16 (input[15:0] a, input[15:0] b, input cin,
output[15:0] sum, output cout );
```



Verilog Code:

```
module RTL_Day_10(a,b,sub,sum);
input [31:0] a,b;
input sub;
output [31:0] sum;
wire w1;
wire[31:0] w2;
assign w2=(b^sub);
add16 a1 (a[15:0],w2[15:0],sub,sum[15:0],w1);
add16 a2 (a[31:16],w2[31:16],w1,sum[31:16],cout);
endmodule
```

Testbench code:

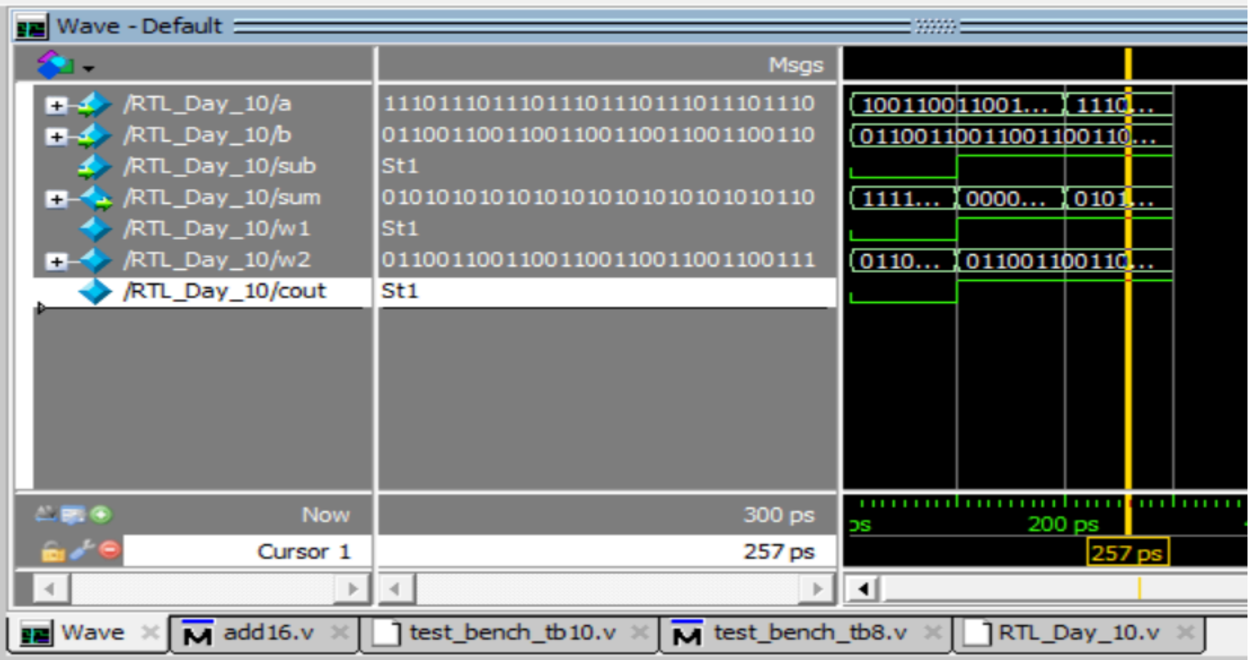
```
module test_bench_tb10 ();
reg [31:0] a,b;
reg sub;
wire [31:0] sum;

RTL_Day_10 s1 (a,b,sub,sum);
initial
begin

a={32{1'b1}}; b={32{1'b0}};   #100;
a={32{1'b1}}; b={32{1'b1}};   #100;
a={32{1'b1}}; b={16{2'b10}};  #100;
a={16{2'b01}}; b={32{1'b0}};

end
endmodule
```

Functional Simulation:



Wave - Default

```
                        Msgs
/RTL_Day_10/a      11101110111011101110111011101110
/RTL_Day_10/b      01100110011001100110011001100110
/RTL_Day_10/sub    St1
/RTL_Day_10/sum    01010101010101010101010101010110
/RTL_Day_10/w1     St1
/RTL_Day_10/w2     01100110011001100110011001100111
/RTL_Day_10/cout   St1

Now                300 ps
Cursor 1           257 ps
```