Waveform

<pre>/sync_adder_tb/reset</pre>	-No					
- → /sync_adder_tb/a	-No	00000101	00000001	00001010	00000100	
+- /sync_adder_tb/b	-No	00000111	00000110	00001111	00000101	
/sync_adder_tb/Q	-No	-{00001100	(00000111	(00011001	(00001001	(00000000

Verilog Code

```
`timescale 1ns / 1ns
module sync_adder_tb;
//signals needed to test functionality of sync adder
logic clk;
logic reset;
logic [7:0] a;
logic [7:0] b;
logic [7:0] Q;
//creating a sync adder and connecting signals
sync_adder DUT(
.clk(clk),
.reset(reset),
.a(a),
.b(b),
.Q(Q)
);
initial // initial block runs only once
        begin
                clk = 0;
```

```
//Simple test
a = 5;
b = 7;
reset = 0;
#10
//self-checking
a = 1;
b = 6;
reset = 0;
#10
if (Q != 7) $display("1+6 failed");
//self-checking
a = 10;
b = 15;
reset = 0;
#10
if (Q != 0) $display("10+15 is not 0!");
//Add code for 4+5=9 -----
//self-checking
a = 4;
b = 5;
reset = 0;
#10
if (Q != 9) $display("4+5 failed!");
//-----
```

```
//Add code to test reset -----
              //testing reset
              a = 4;
              b = 5;
              reset = 1;
              #10
              if (Q != 0) $display("reset failed!");
              #5
              $stop;
       end
always
       #1 clk = ~clk;
```

endmodule