Verilog

GCD Computation

Datapath + Controller

Top module

gcd_computation.v - D:/codes/gcd

```
1  `timescale 1ns/1ps
2  module gcd_computation(clk,rst,go,data_in1,data_in2,out,done);
3  input clk,rst,go;
4  input [7:0] data_in1,data_in2;
5  output [7:0] out;
6  output done;
7  wire a_gt_b,a_eq_b,a_lt_b; // input to the controller from datapath
8  wire a_ld,a_sel,b_ld,b_sel; // output from controller to datapath
9  wire output_en;
10
11 // instantiate datapath and controller
12
13 controller c1(clk,rst,go,a_gt_b,a_eq_b,a_lt_b,a_ld,b_ld,a_sel,b_sel,output_en,done);
14 datapath d1(clk,rst,data_in1,data_in2,a_sel,b_sel,a_ld,b_ld,a_gt_b,a_eq_b,a_lt_b,output_en,out);
15
16 endmodule
```

Datapath

Scriptum (22.0 Rev 1) - [datapath.v - D:/codes/gcd]

File Edit Search Folding Options Window Help

```
 1  `timescale 1ns/1ps
 2  module datapath(clk,rst,data_in1,data_in2,a_sel,b_sel,a_ld,b_ld,a_gt_b,a_eq_b,a_lt_b,output_en,out);
 3  input [7:0] data_in1,data_in2;
 4  input clk,rst;
 5  input a_sel,b_sel,a_ld,b_ld;
 6  output a_gt_b,a_eq_b,a_lt_b;
 7  input output_en;
 8  output [7:0] out;
 9  wire [7:0] tain,tbin,taout,tbout,t1,t2;
10  subtractor s1(taout,tbout,t1);
11  subtractor s2(tbout,taout,t2);
12  mux m1(tain,a_sel,t1,data_in1);
13  mux m2(tbin,b_sel,t2,data_in2);
14  register r1(clk,rst,tain,a_ld,taout);
15  register r2(clk,rst,tbin,b_ld,tbout);
16  register rout(clk,rst,taout,output_en,out);
17  comparator c1(taout,tbout,a_gt_b,a_eq_b,a_lt_b);
18  endmodule
```

Controller

controller.v - D:/codes/gcd

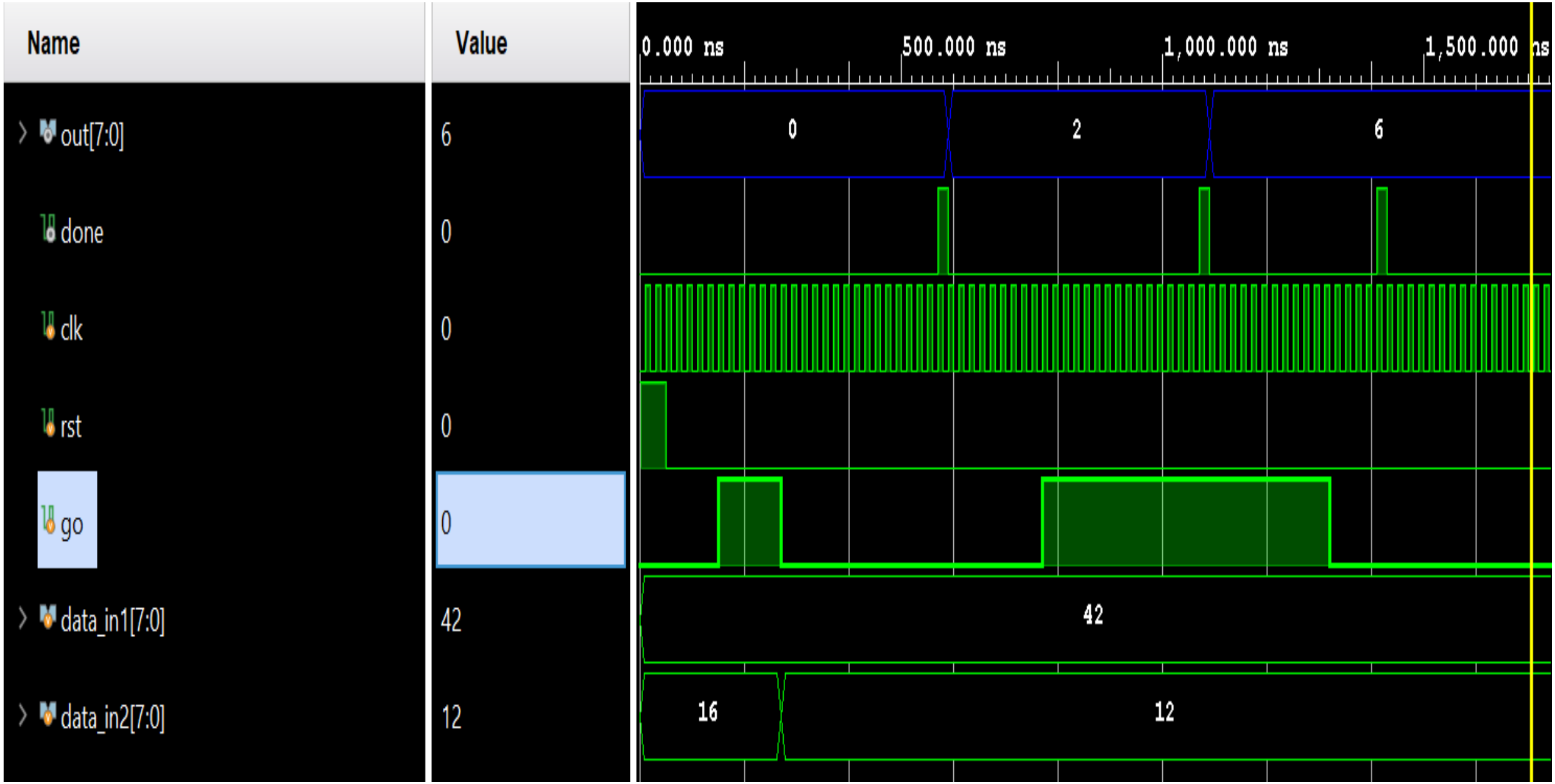
```
1  `timescale 1ns/1ps
2  module controller(clk,rst,go,a_gt_b,a_eq_b,a_lt_b,a_ld,b_ld,a_sel,b_sel,output_en,done);
3  input clk,rst;
4  input go;
5  input a_gt_b,a_eq_b,a_lt_b;
6  output reg a_sel,b_sel,a_ld,b_ld,done,output_en;
7  reg [2:0] ps,ns;
8  parameter s0=3'b000;
9  parameter s1=3'b001;
10 parameter s2=3'b010;
11 parameter s3=3'b011;
12 parameter s4=3'b100;
13 parameter s5=3'b101;
14 parameter s6=3'b110;
15 parameter s7=3'b111;
16
17 // modelling the state register
18
19 always @(posedge clk or posedge rst)
20 begin
21 if (rst==1'b1)
22 ps<=s0;
23 else
24 ps<=ns;
25 end
```

```
27  //Next state logic (combinational)
28  always @(go or a_gt_b or a_lt_b or a_eq_b or ps)
29  begin
30  case(ps)
31  s0: begin
32  if (go==1'b0)
33  ns=s0;
34  else
35  ns=s1;
36  end
37  s1:ns=s2;
38  s2:ns=s3;
39  s3:begin
40  if(a_gt_b==1'b1)
41  ns=s4;
42  else if (a_lt_b==1'b1)
43  ns=s5;
44  else
45  ns=s7;
46  end
47  s4:ns=s6;
48  s5:ns=s6;
49  s6:ns=s3;
50  s7:ns=s0;
51  default:ns=s0;
52  endcase
53  end
```

```
54  // output logic (combinational)
55  always @(ps)
56  begin
57  case(ps)
58  s0:begin
59  a_sel=1'b0;
60  b_sel=1'b0;
61  a_ld=1'b0;
62  b_ld=1'b0;
63  done=1'b0;
64  output_en=1'b0;
65  end
66  s1:begin
67  a_sel=1'b1;
68  b_sel=1'b1;
69  a_ld=1'b1;
70  b_ld=1'b1;
71  done=1'b0;
72  output_en=1'b0;
73  end
74  s2:begin
75  a_sel=1'b0;
76  b_sel=1'b0;
77  a_ld=1'b0;
78  b_ld=1'b0;
79  done=1'b0;
80  output_en=1'b0;
81  end
```

```
 82  s3:begin
 83  a_sel=1'b0;
 84  b_sel=1'b0;
 85  a_ld=1'b0;
 86  b_ld=1'b0;
 87  done=1'b0;
 88  output_en=1'b0;
 89  end
 90  s4:begin
 91  a_sel=1'b0;
 92  b_sel=1'b0;
 93  a_ld=1'b1;
 94  b_ld=1'b0;
 95  done=1'b0;
 96  output_en=1'b0;
 97  end
 98  s5:begin
 99  a_sel=1'b0;
100  b_sel=1'b0;
101  a_ld=1'b0;
102  b_ld=1'b1;
103  done=1'b0;
104  output_en=1'b0;
105  end
```

```
106  s6:begin
107  a_sel=1'b0;
108  b_sel=1'b0;
109  a_ld=1'b0;
110  b_ld=1'b0;
111  done=1'b0;
112  output_en=1'b0;
113  end
114  s7:begin
115  a_sel=1'b0;
116  b_sel=1'b0;
117  a_ld=1'b0;
118  b_ld=1'b0;
119  done=1'b1;
120  output_en=1'b1;
121  end
122  default:begin
123  a_sel=1'b0;
124  b_sel=1'b0;
125  a_ld=1'b0;
126  b_ld=1'b0;
127  done=1'b0;
128  output_en=1'b0;
129  end
130  endcase
131  end
132  endmodule
```


Name
Value
0.000 ns
500.000 ns
1,000.000 ns
1,500.000 ns
out[7:0]
6
0
2
6
done
0
clk
0
rst
0
go
0
data_in1[7:0]
42
42
data_in2[7:0]
12
16
12