

Assignment 3 EE 5193

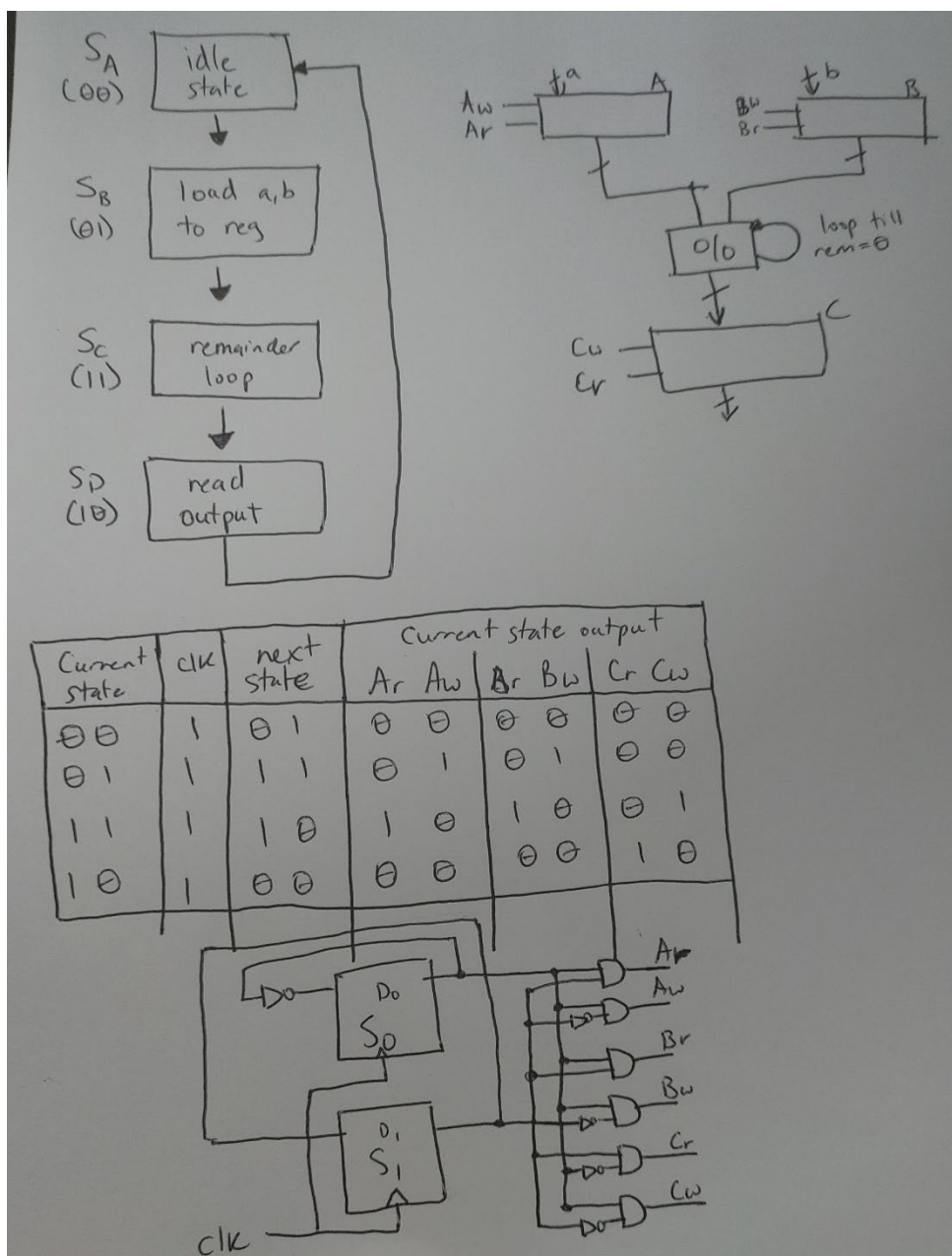
Problem Statement:

Design a processor that would calculate the Greatest Common Divisor of two integers, using the Euclidean Algorithm.

Approach:

My approach was very simple, I pretty much took the in-class example and modified it to be able to properly solve for the GCD. I know that normally the algorithm can be done with only one state, but I figured it would be good to incorporate two states, so the problem waited until both inputs (A and B) were accepted and ready to compute.

Block Diagram & ASM-D charts:



Verilog Code:

(Top Module)

The screenshot shows the Verilog code for the `asmd_GCD` module. The left pane displays the project hierarchy under 'Sources', showing `asmd_GCD` (2) with sub-modules `UUT1: controlUnit_GCD` and `UUT2: dataFlow_GCD`, and a testbench `tb_asmd_GCD`. The right pane shows the code for `asmd_GCD.v`, which includes a header with metadata and a module definition.

```

1  `timescale 1ns / 1ps
2  // Company: EE 5193 FPGA & HDL
3  // Engineer: Oscar Osornio
4  //
5  //
6  // Create Date: 07/24/2022 03:23:23 PM
7  // Design Name: Assignment 3
8  // Module Name: asmd_GCD
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //
21
22
23 module asmd_GCD(
24     input clk, rst,
25     input [15:0] a, b,
26     output [15:0] c,
27     output [1:0] state, nstate,
28     output Ar, Aw, Br, Bw, Cr, Cw
29 );
30
31     controlUnit_GCD UUT1(clk, rst, Ar, Aw, Br, Bw, Cr, Cw, state, nstate);
32     dataFlow_GCD UUT2(Ar, Aw, Br, Bw, Cr, Cw, a, b, c);
33
34 endmodule
35

```

(Control Unit Module)

The screenshot shows the Verilog code for the `controlUnit_GCD` module. The left pane displays the project hierarchy under 'Sources', showing `asmd_GCD` (2) with sub-modules `UUT1: controlUnit_GCD` and `UUT2: dataFlow_GCD`, and a testbench `tb_asmd_GCD`. The right pane shows the code for `controlUnit_GCD.v`, which includes a header with metadata and a module definition.

```

1  `timescale 1ns / 1ps
2  // Company: EE 5193 FPGA & HDL
3  // Engineer: Oscar Osornio
4  //
5  //
6  // Create Date: 07/24/2022 03:23:23 PM
7  // Design Name: Assignment 3
8  // Module Name: controlUnit_GCD
9  // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //
21
22
23 module controlUnit_GCD(
24     input clk, rst,
25     output Ar, Aw, Br, Bw, Cr, Cw,
26     output [1:0] state, nstate
27 );
28
29     reg [1:0] state, nstate;
30     reg Ar, Aw, Br, Bw, Cr, Cw;
31
32     always @(posedge rst or posedge clk) begin
33         if (rst) begin
34             state <= 2'b00;
35             nstate <= 2'b00;
36         end
37         else
38             state <= nstate;
39     end
40

```

Scope Sources x ? _ □ □

Q 🔍 ⚙️ 0

Design Sources (1)

- asmd_GCD (asmd_GCD.v) (2)
 - UUT1 : controlUnit_GCD (controlUnit_GCD.v)
 - UUT2 : dataFlow_GCD (dataFlow_GCD.v)

Constraints

Simulation Sources (1)

- sim_1 (1)
 - tb_asmd_GCD (tb_asmd_GCD.v) (1)

Utility Sources

asmd_GCD.v x dataFlow_GCD.v x tb_asmd_GCD.v x controlUnit_GCD.v x

C:/Users/oscao/Assignment 3/Assignment 3.srscs/sources_1/new/controlUnit_GCD.v

Q 🔍 ⚙️

```

17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module controlUnit_GCD(
24     input clk, rst,
25     output Ar, Aw, Br, Bw, Cr, Cw,
26     output [1:0] state, nstate
27 );
28
29     reg [1:0] state, nstate;
30     reg Ar, Aw, Br, Bw, Cr, Cw;
31
32     always @(posedge rst or posedge clk) begin
33         if (rst) begin
34             state <= 2'b00;
35             nstate <= 2'b00;
36         end
37         else
38             state <= nstate;
39     end
40
41     always @(state) begin
42         case (state)
43             2'b00: nstate <= 2'b01;
44             2'b01: nstate <= 2'b11;
45             2'b11: nstate <= 2'b10;
46             2'b10: nstate <= 2'b00;
47         endcase
48     end
49
50     always @(state) begin
51         case (state)
52             2'b00: begin Ar=0; Aw=0; Br=0; Bw=0; Cr=0; Cw=0; end
53             2'b01: begin Ar=0; Aw=1; Br=0; Bw=1; Cr=0; Cw=0; end
54             2'b11: begin Ar=1; Aw=0; Br=1; Bw=0; Cr=0; Cw=1; end
55             2'b10: begin Ar=0; Aw=0; Br=0; Bw=0; Cr=1; Cw=0; end
56         endcase
57     end
58
59 endmodule
60

```

(Dataflow Module)

Scope Sources x ? _ □ □

Q 🔍 ⚙️ 0

Design Sources (1)

- asmd_GCD (asmd_GCD.v) (2)
 - UUT1 : controlUnit_GCD (controlUnit_GCD.v)
 - UUT2 : dataFlow_GCD (dataFlow_GCD.v)

Constraints

Simulation Sources (1)

- sim_1 (1)
 - tb_asmd_GCD (tb_asmd_GCD.v) (1)

Utility Sources

asmd_GCD.v x dataFlow_GCD.v x tb_asmd_GCD.v x controlUnit_GCD.v x

C:/Users/oscao/Assignment 3/Assignment 3.srscs/sources_1/new/dataFlow_GCD.v

Q 🔍 ⚙️

```

1 //timescale 1ns / 1ps
2 ///////////////////////////////////////////////////////////////////
3 // Company: EE 5193 FPGA & HDL
4 // Engineer: Oscar Osornio
5 //
6 // Create Date: 07/24/2022 03:23:23 PM
7 // Design Name: Assignment 3
8 // Module Name: dataFlow_GCD
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 ///////////////////////////////////////////////////////////////////
21
22
23 module dataFlow_GCD(
24     input Ar, Aw, Br, Bw, Cr, Cw,
25     input [15:0] a, b,
26     output [15:0] c
27 );
28
29     reg [15:0] c;
30     reg [15:0] A, B, C;
31
32     always @(Aw or a) begin
33         if (Aw) begin
34             A <= a;
35         end
36     end
37
38     always @(Bw or b) begin
39         if (Bw) begin
40             B <= b;
41         end
42     end
43

```

The screenshot shows the Vivado IDE interface. On the left, the 'Sources' window displays a project hierarchy with 'Design Sources (1)' containing 'asmd_GCD (asmd_GCD.v) (2)', which includes 'UUT1: controlUnit_GCD (controlUnit_GCD.v)' and 'UUT2: dataFlow_GCD (dataFlow_GCD.v)'. Below this are 'Constraints', 'Simulation Sources (1)' with 'sim_1 (1)', and 'Utility Sources'. The main editor window shows the 'dataFlow_GCD.v' file, which is a Verilog module implementing a Greatest Common Divisor (GCD) algorithm using the Euclidean algorithm. The module has inputs Ar, Aw, Br, Bw, Cr, and Cw, and an output c. It uses registers for c, A, B, and C, and always blocks to update these registers based on the inputs and the current state of the algorithm.

```

19 //
20 //
21 //
22 //
23 module dataFlow_GCD(
24     input Ar, Aw, Br, Bw, Cr, Cw,
25     input [15:0] a, b,
26     output [15:0] c
27 );
28
29     reg [15:0] c;
30     reg [15:0] A, B, C;
31
32     always @(Aw or a) begin
33         if (Aw) begin
34             A <= a;
35         end
36     end
37
38     always @(Bw or b) begin
39         if (Bw) begin
40             B <= b;
41         end
42     end
43
44     always @(Cr) begin
45         if (Cr) begin
46             c <= C;
47         end
48     end
49
50     always @(Ar or Br or Cw) begin
51         if (Ar && Br && Cw) begin
52             while (B > 0) begin
53                 C = A % B;
54                 A = B;
55                 B = C;
56             end
57             c = A;
58         end
59     end
60
61 endmodule
62

```

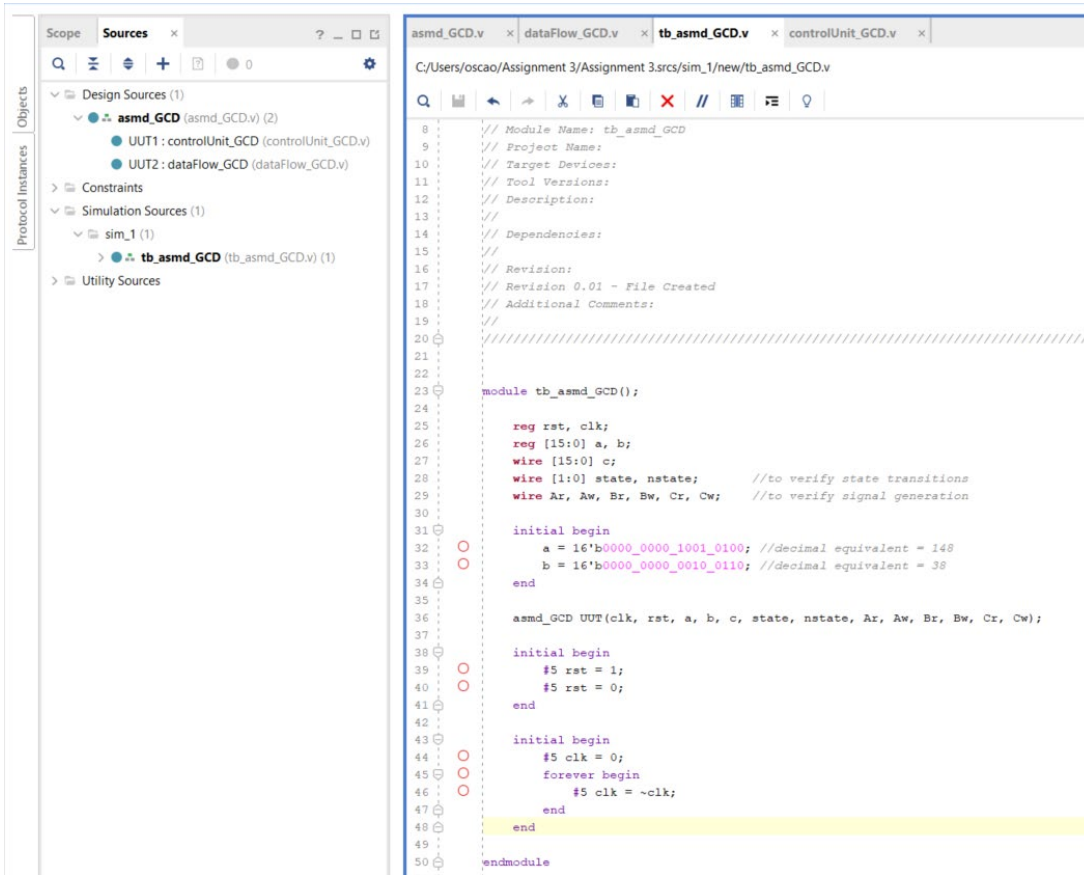
(Test Bench)

The screenshot shows the Vivado IDE interface. On the left, the 'Sources' window displays a project hierarchy with 'Design Sources (1)' containing 'asmd_GCD (asmd_GCD.v) (2)', which includes 'UUT1: controlUnit_GCD (controlUnit_GCD.v)' and 'UUT2: dataFlow_GCD (dataFlow_GCD.v)'. Below this are 'Constraints', 'Simulation Sources (1)' with 'sim_1 (1)', and 'Utility Sources'. The main editor window shows the 'tb_asmd_GCD.v' file, which is a Verilog testbench for the 'asmd_GCD' module. It includes a timescale, a header section with metadata, and an initial block to set up the test environment. The testbench uses the 'asmd_GCD' module and applies initial values to the inputs and outputs.

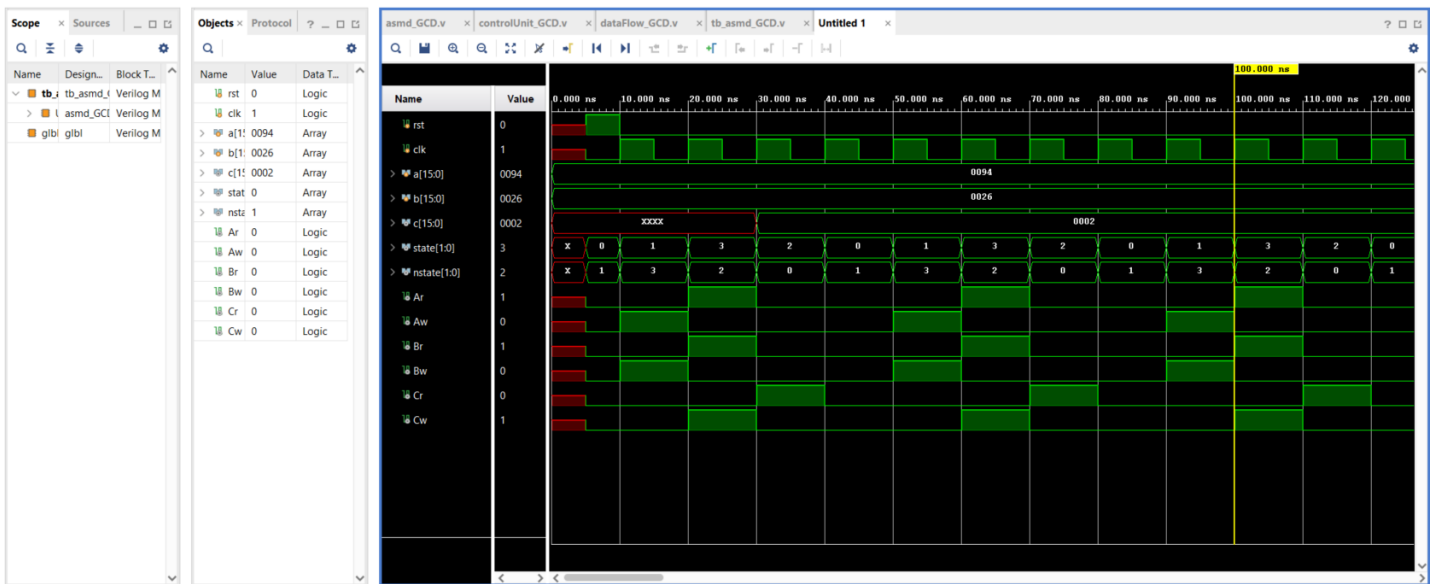
```

1 //timescale 1ns / 1ps
2 //
3 // Company:
4 // Engineer:
5 //
6 // Create Date: 07/24/2022 03:57:08 PM
7 // Design Name:
8 // Module Name: tb_asmd_GCD
9 // Project Name:
10 // Target Devices:
11 // Tool Versions:
12 // Description:
13 //
14 // Dependencies:
15 //
16 // Revision:
17 // Revision 0.01 - File Created
18 // Additional Comments:
19 //
20 //
21 //
22 //
23 module tb_asmd_GCD();
24
25     reg rst, clk;
26     reg [15:0] a, b;
27     wire [15:0] c;
28     wire [1:0] state, nstate; //to verify state transitions
29     wire Ar, Aw, Br, Bw, Cr, Cw; //to verify signal generation
30
31     initial begin
32         a = 16'b0000_0000_1001_0100; //decimal equivalent = 148
33         b = 16'b0000_0000_0010_0110; //decimal equivalent = 38
34     end
35
36     asmd_GCD UUT(clk, rst, a, b, c, state, nstate, Ar, Aw, Br, Bw, Cr, Cw);
37
38     initial begin
39         #5 rst = 1;
40         #5 rst = 0;
41     end
42

```



Simulation Screenshot:



Problems:

I ended up not having any simulation errors during the whole process. Since I went ahead and followed the same structure, compared to the example code, when writing the code, I was able to properly run the simulation on the first try.