Oscar Osornio

Assignment 3 EE 5193

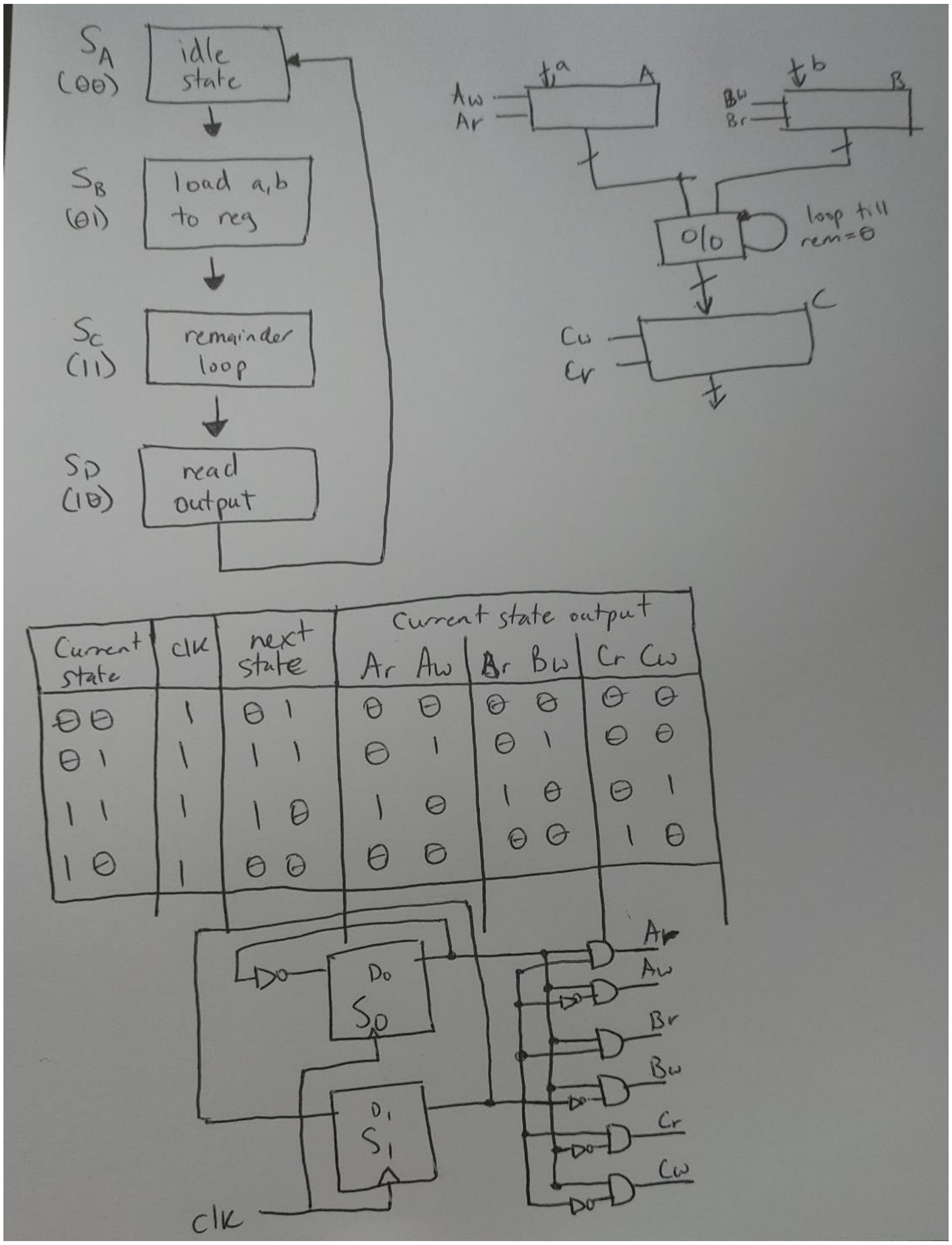
Problem Statement:

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:

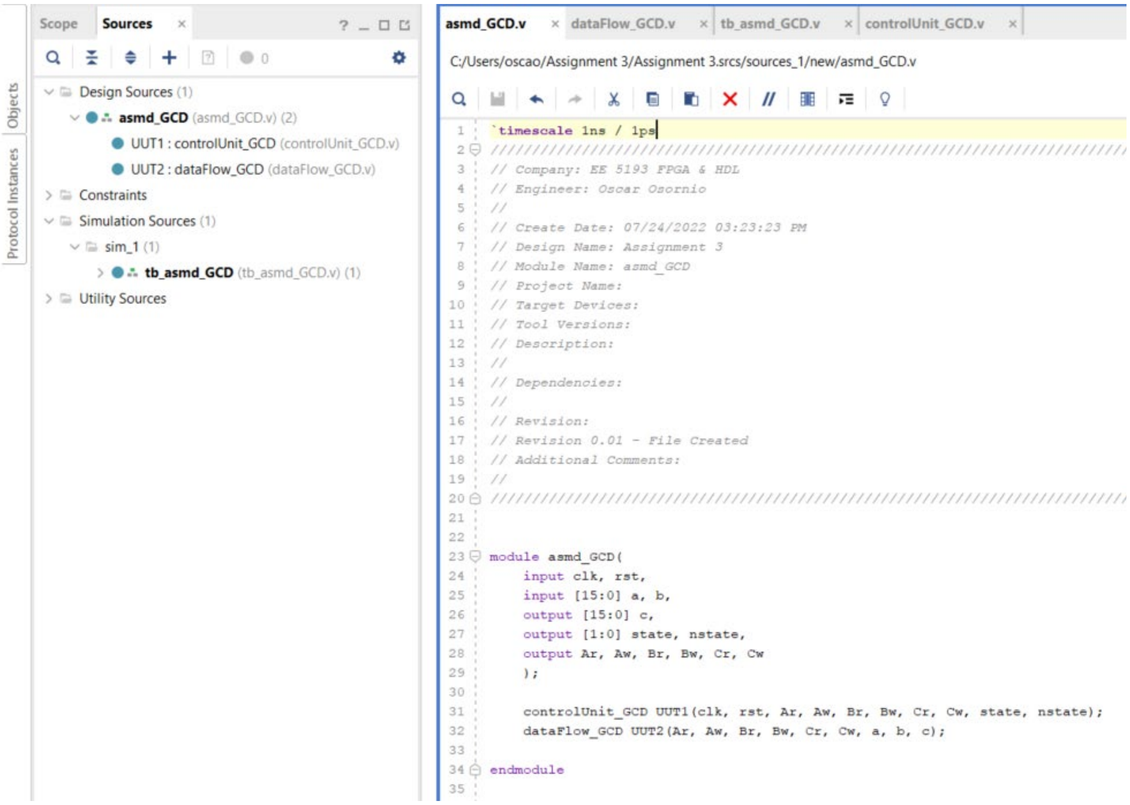


Current state	clk	next state	Current state output					
			Ar	Aw	Br	Bw	Cr	Cw
0 0	1	0 1	0	0	0	0	0	0
0 1	1	1 1	0	1	0	1	0	0
1 1	1	1 0	1	0	1	0	0	1
1 0	1	0 0	0	0	0	0	1	0

Oscar Osornio

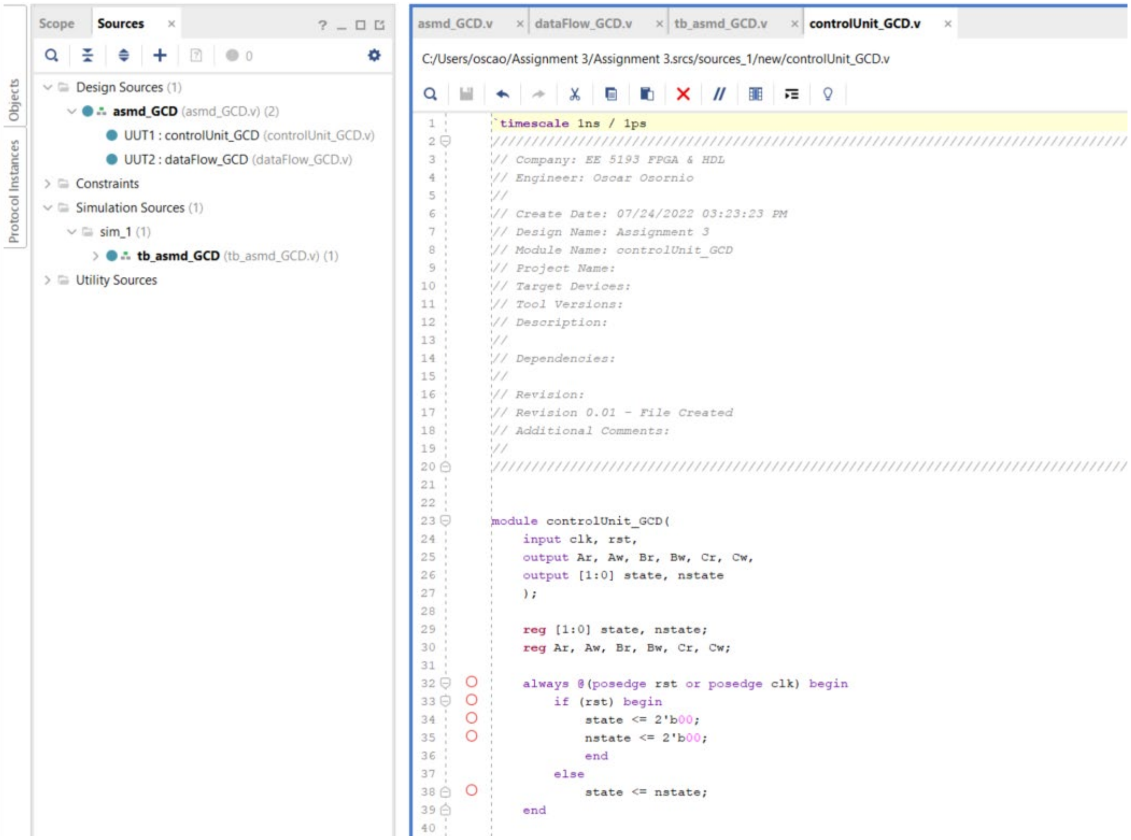
Verilog Code:

(Top Module)


Objects
Protocol Instances
Scope
Sources
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
dataFlow_GCD.v
tb_asmd_GCD.v
controlUnit_GCD.v
C:/Users/oscao/Assignment 3/Assignment 3.srcs/sources_1/new/asmd_GCD.v

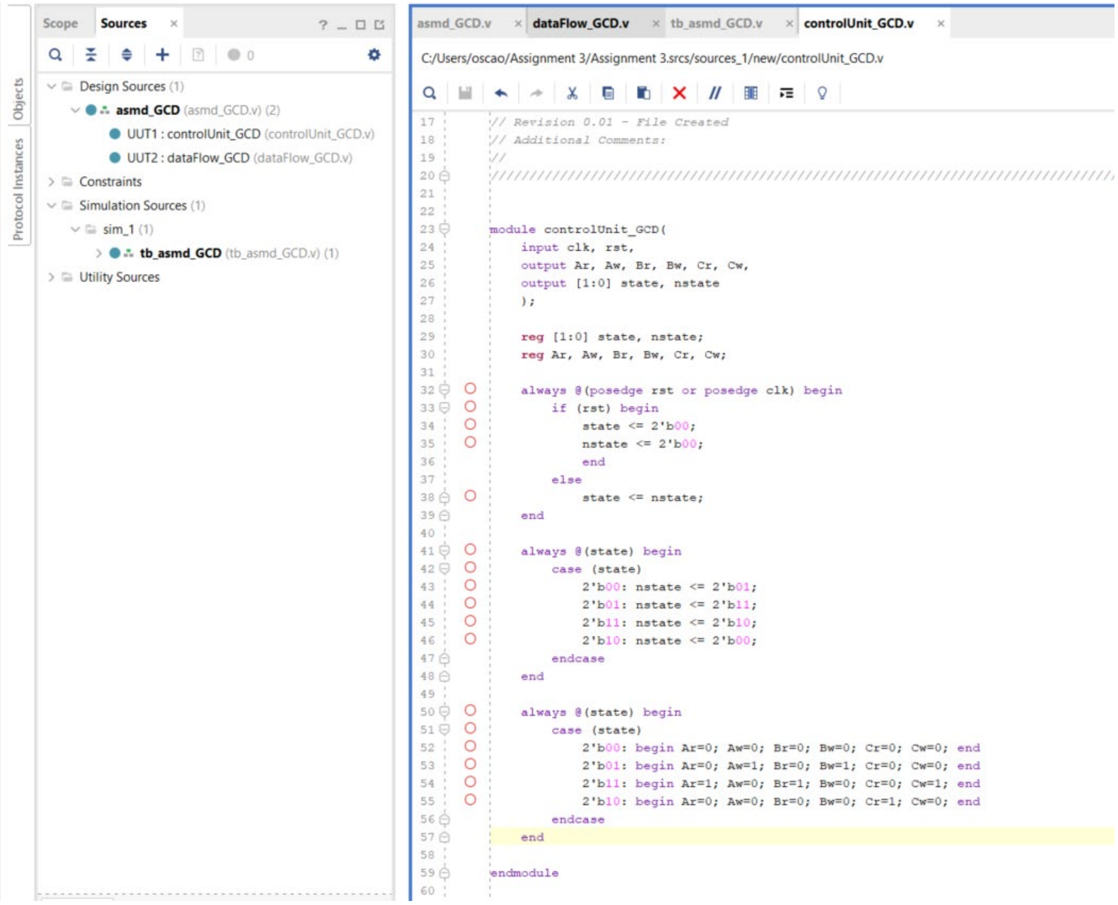
```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: EE 5193 FPGA & HDL
// Engineer: Oscar Osornio
//
// Create Date: 07/24/2022 03:23:23 PM
// Design Name: Assignment 3
// Module Name: asmd_GCD
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module asmd_GCD(
    input clk, rst,
    input [15:0] a, b,
    output [15:0] c,
    output [1:0] state, nstate,
    output Ar, Aw, Br, Bw, Cr, Cw
    );

    controlUnit_GCD UUT1(clk, rst, Ar, Aw, Br, Bw, Cr, Cw, state, nstate);
    dataFlow_GCD UUT2(Ar, Aw, Br, Bw, Cr, Cw, a, b, c);

endmodule

```

(Control Unit Module)


Objects
Protocol Instances
Scope
Sources
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
dataFlow_GCD.v
tb_asmd_GCD.v
controlUnit_GCD.v
C:/Users/oscao/Assignment 3/Assignment 3.srcs/sources_1/new/controlUnit_GCD.v

```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company: EE 5193 FPGA & HDL
// Engineer: Oscar Osornio
//
// Create Date: 07/24/2022 03:23:23 PM
// Design Name: Assignment 3
// Module Name: controlUnit_GCD
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module controlUnit_GCD(
    input clk, rst,
    output Ar, Aw, Br, Bw, Cr, Cw,
    output [1:0] state, nstate
    );

    reg [1:0] state, nstate;
    reg Ar, Aw, Br, Bw, Cr, Cw;

    always @(posedge rst or posedge clk) begin
        if (rst) begin
            state <= 2'b00;
            nstate <= 2'b00;
            end
        else
            state <= nstate;
    end

```

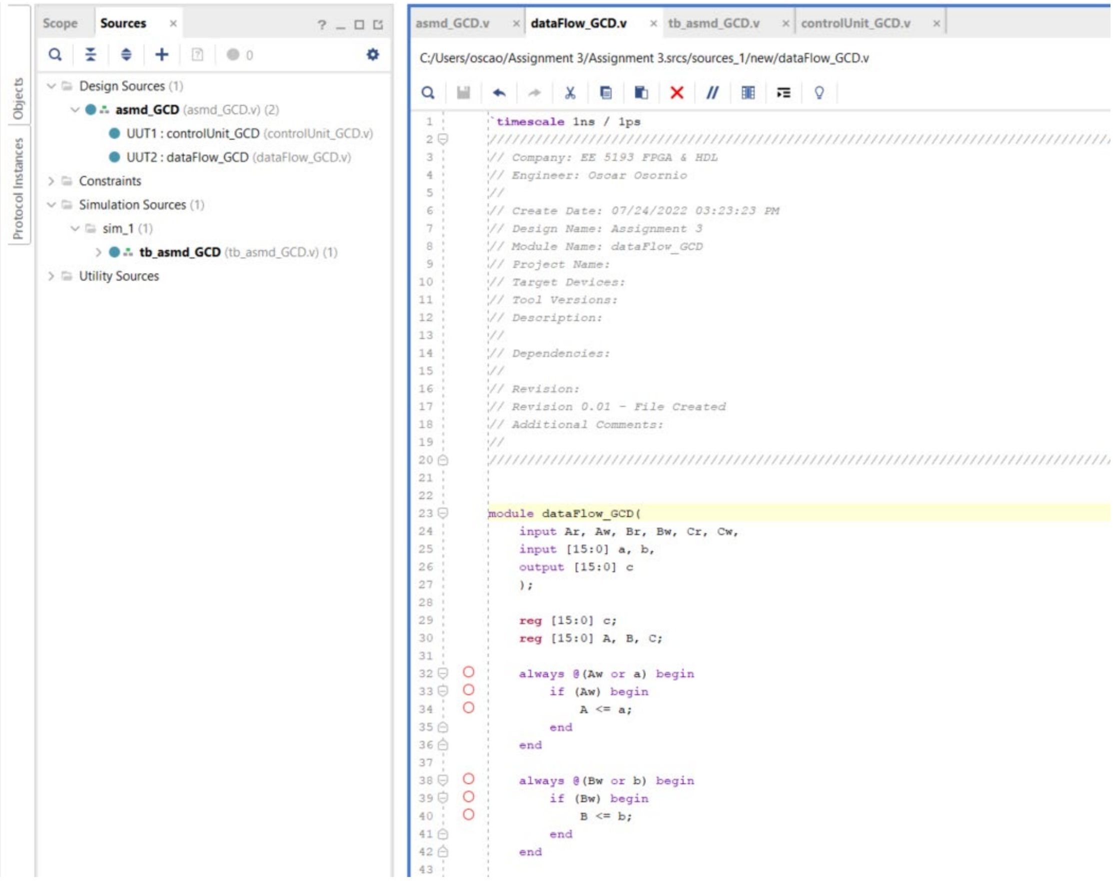
Oscar Osornio



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

```
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////


module controlUnit_GCD(
    input clk, rst,
    output Ar, Aw, Br, Bw, Cr, Cw,
    output [1:0] state, nstate
    );

    reg [1:0] state, nstate;
    reg Ar, Aw, Br, Bw, Cr, Cw;

    always @(posedge rst or posedge clk) begin
        if (rst) begin
            state <= 2'b00;
            nstate <= 2'b00;
            end
        else
            state <= nstate;
    end

    always @(state) begin
        case (state)
            2'b00: nstate <= 2'b01;
            2'b01: nstate <= 2'b11;
            2'b11: nstate <= 2'b10;
            2'b10: nstate <= 2'b00;
        endcase
    end

    always @(state) begin
        case (state)
            2'b00: begin Ar=0; Aw=0; Br=0; Bw=0; Cr=0; Cw=0; end
            2'b01: begin Ar=0; Aw=1; Br=0; Bw=1; Cr=0; Cw=0; end
            2'b11: begin Ar=1; Aw=0; Br=1; Bw=0; Cr=0; Cw=1; end
            2'b10: begin Ar=0; Aw=0; Br=0; Bw=0; Cr=1; Cw=0; end
        endcase
    end

endmodule
```

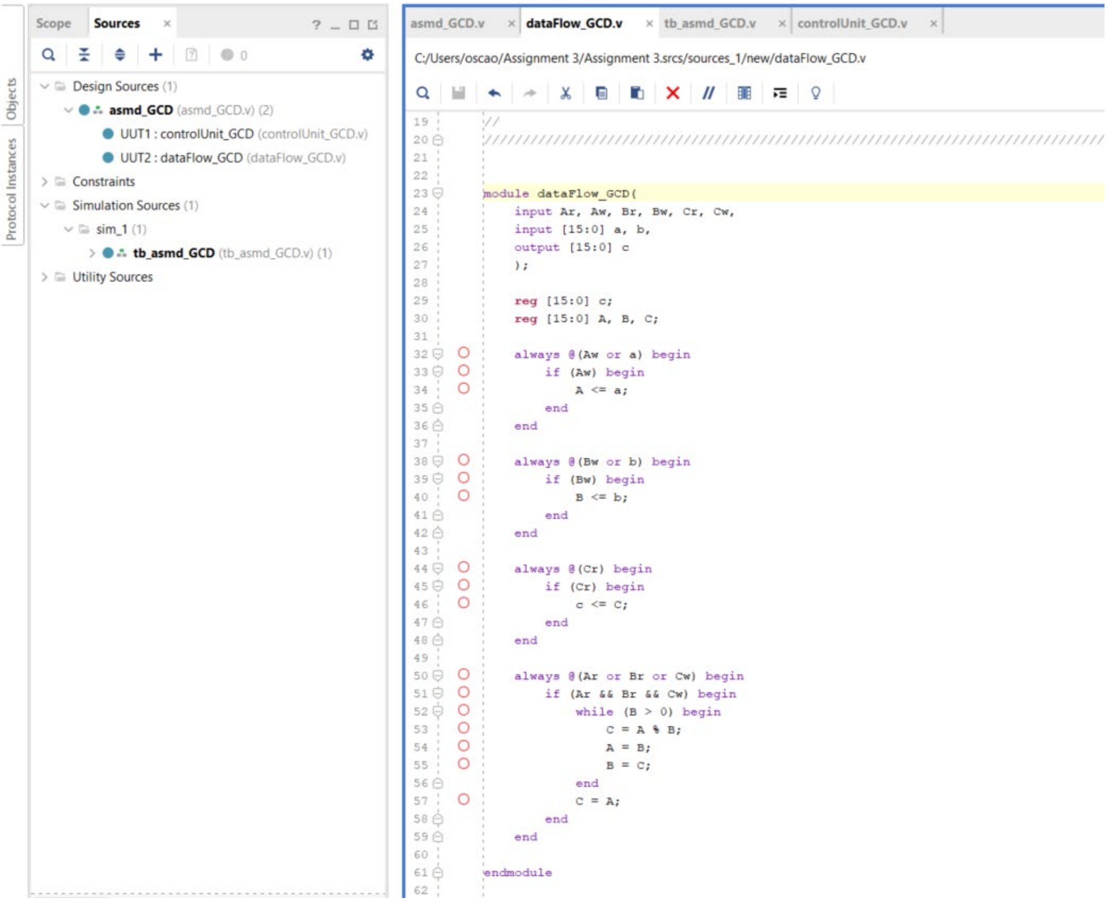
(Dataflow Module)



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

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////
// Company: EE 5193 FPGA & HDL
// Engineer: Oscar Osornio
//
// Create Date: 07/24/2022 03:23:23 PM
// Design Name: Assignment 3
// Module Name: dataFlow_GCD
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////


module dataFlow_GCD(
    input Ar, Aw, Br, Bw, Cr, Cw,
    input [15:0] a, b,
    output [15:0] c
    );

    reg [15:0] c;
    reg [15:0] A, B, C;

    always @(Aw or a) begin
        if (Aw) begin
            A <= a;
        end
    end

    always @(Bw or b) begin
        if (Bw) begin
            B <= b;
        end
    end

```

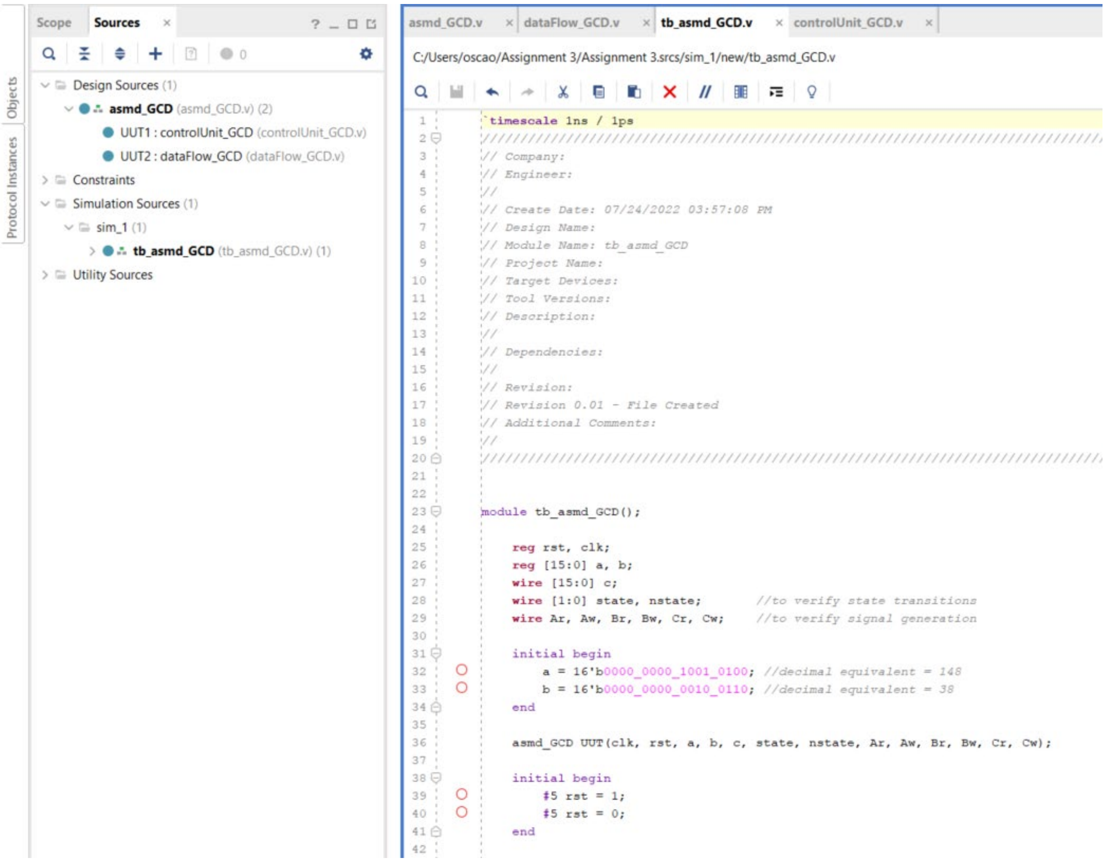
Oscar Osornio



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

```
//
////////////////////////////////////////////////////////////////////////////////


module dataFlow_GCD(
    input Ar, Aw, Br, Bw, Cr, Cw,
    input [15:0] a, b,
    output [15:0] c
    );

    reg [15:0] c;
    reg [15:0] A, B, C;

    always @(Aw or a) begin
        if (Aw) begin
            A <= a;
        end
    end

    always @(Bw or b) begin
        if (Bw) begin
            B <= b;
        end
    end

    always @(Cr) begin
        if (Cr) begin
            c <= C;
        end
    end

    always @(Ar or Br or Cw) begin
        if (Ar && Br && Cw) begin
            while (B > 0) begin
                C = A % B;
                A = B;
                B = C;
            end
            C = A;
        end
    end

endmodule
```

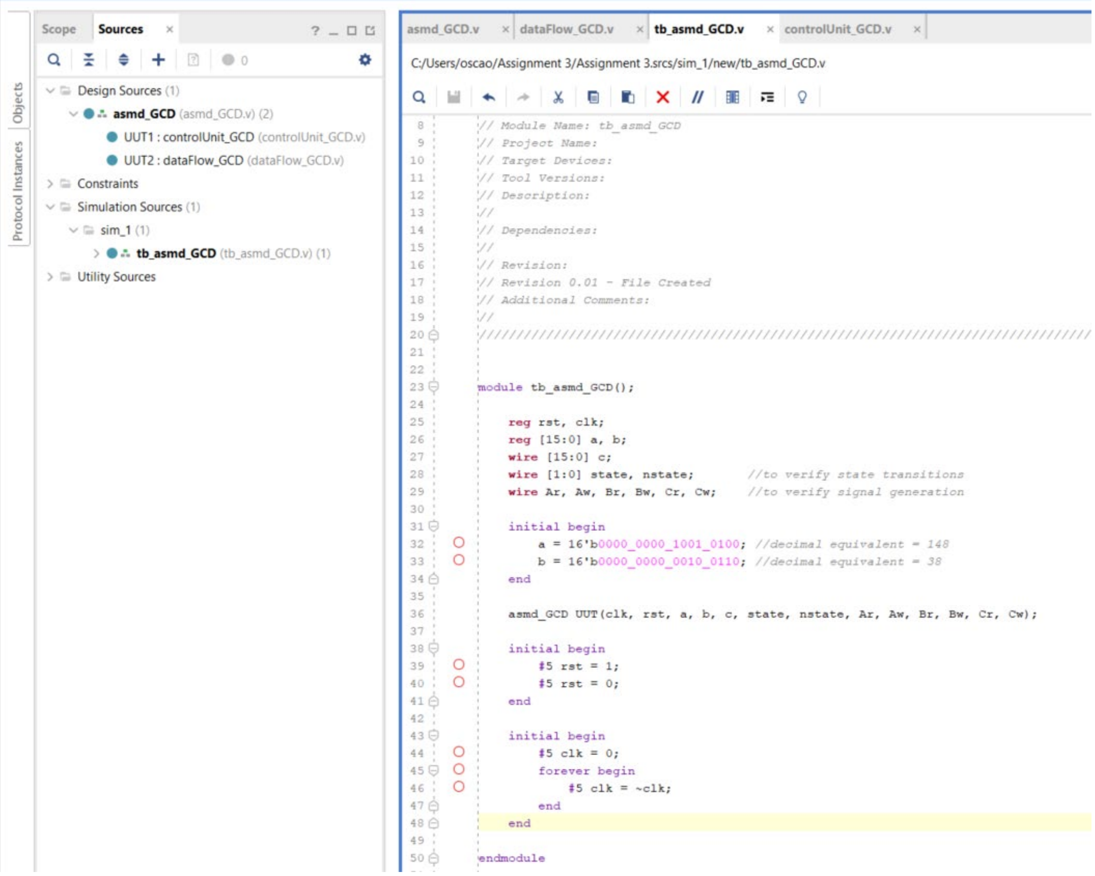
(Test Bench)



C:/Users/oscao/Assignment 3/Assignment 3.srcs/sim_1/new/tb_asmd_GCD.v

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 07/24/2022 03:57:08 PM
// Design Name:
// Module Name: tb_asmd_GCD
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////////////////////


module tb_asmd_GCD();

    reg rst, clk;
    reg [15:0] a, b;
    wire [15:0] c;
    wire [1:0] state, nstate;       //to verify state transitions
    wire Ar, Aw, Br, Bw, Cr, Cw;    //to verify signal generation

    initial begin
        a = 16'b0000_0000_1001_0100; //decimal equivalent = 148
        b = 16'b0000_0000_0010_0110; //decimal equivalent = 38
    end

    asmd_GCD UUT(clk, rst, a, b, c, state, nstate, Ar, Aw, Br, Bw, Cr, Cw);

    initial begin
        #5 rst = 1;
        #5 rst = 0;
    end

```

Oscar Osornio

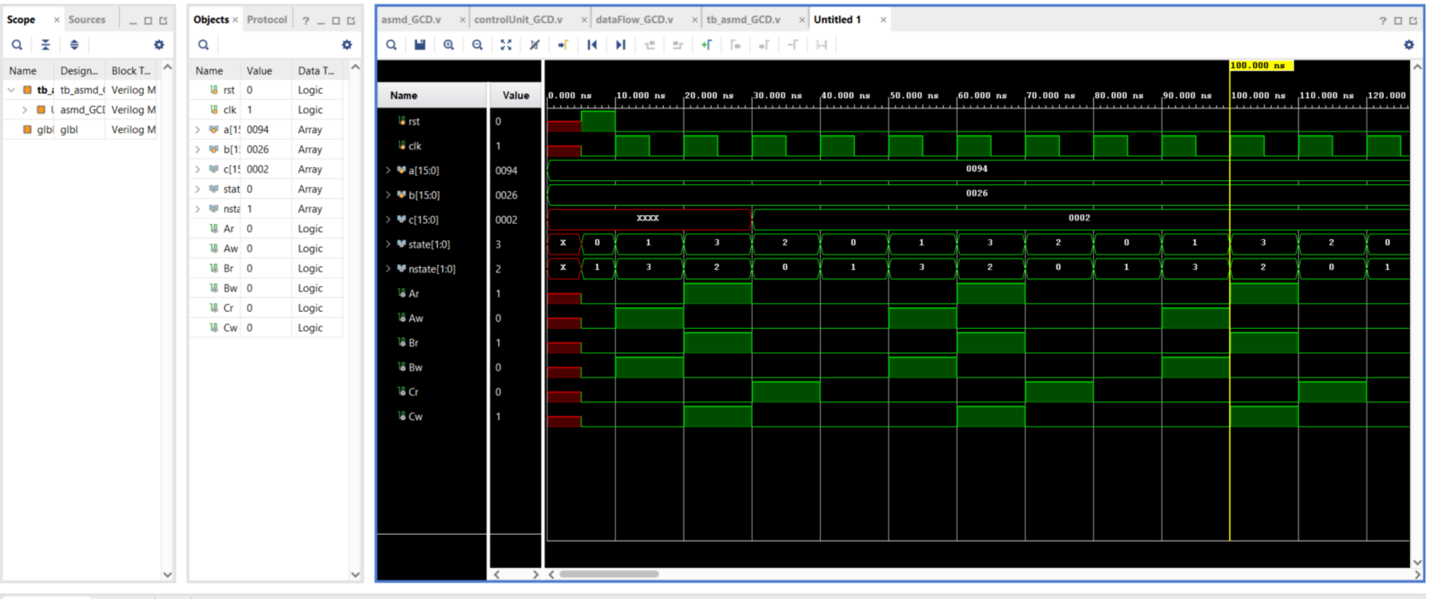


- Design Sources (1)
  - asmd_GCD (asmd_GCD.v) (2)
    - UUT1 : controlUnit_GCD (controlUnit_GCD.v)
    - UUT2 : dataFlow_GCD (dataFlow_GCD.v)
- Constraints
- Simulation Sources (1)
  - sim_1 (1)
    - tb_asmd_GCD (tb_asmd_GCD.v) (1)
- Utility Sources

C:/Users/oscao/Assignment 3/Assignment 3.srcs/sim_1/new/tb_asmd_GCD.v

```
// Module Name: tb_asmd_GCD
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module tb_asmd_GCD();

    reg rst, clk;
    reg [15:0] a, b;
    wire [15:0] c;
    wire [1:0] state, nstate;       //to verify state transitions
    wire Ar, Aw, Br, Bw, Cr, Cw;    //to verify signal generation

    initial begin
        a = 16'b0000_0000_1001_0100; //decimal equivalent = 148
        b = 16'b0000_0000_0010_0110; //decimal equivalent = 38
    end

    asmd_GCD UUT(clk, rst, a, b, c, state, nstate, Ar, Aw, Br, Bw, Cr, Cw);

    initial begin
        #5 rst = 1;
        #5 rst = 0;
    end

    initial begin
        #5 clk = 0;
        forever begin
            #5 clk = ~clk;
        end
    end

endmodule
```

Simulation Screenshot:



Name	Design...	Block T...
tb_a	tb_asmd_(	Verilog M
U	asmd_GCI	Verilog M
glbl	glbl	Verilog M

Name	Value	Data T...
rst	0	Logic
clk	1	Logic
a[1:	0094	Array
b[1:	0026	Array
c[1:	0002	Array
stat	0	Array
nsta	1	Array
Ar	0	Logic
Aw	0	Logic
Br	0	Logic
Bw	0	Logic
Cr	0	Logic
Cw	0	Logic

Name	Value
rst	0
clk	1
a[15:0]	0094
b[15:0]	0026
c[15:0]	0002
state[1:0]	3
nstate[1:0]	2
Ar	1
Aw	0
Br	1
Bw	0
Cr	0
Cw	1

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.