Digital Systems : Assignment2

Nimitt (22110169)

Pratham Sharda (22110203)

This document demonstrates the Assignment 2 of the Digital Systems course.

Port Names

1 Bit Adder

Module Name : FullAdder

1. A : First Number
2. B : Second Number
3. Cin : Carry in
4. Sum, Cout

8 Bit Adder/ Subtractor

Module Name : AdderSubtractor8b

1. A : 8 Bit First Number
2. B : 8 Bit Second Number
3. OUT : 8 bit Answer (Sum/ Difference)
4. Cout : Carry overflow

Gray Converter

1. A : 9 Bit Number
2. OUT : 9 Bit Output

Combined :

1. A : First 8 bit Number

2. B : Second No
3. OUT : Final Output

1 Bit Full Adder

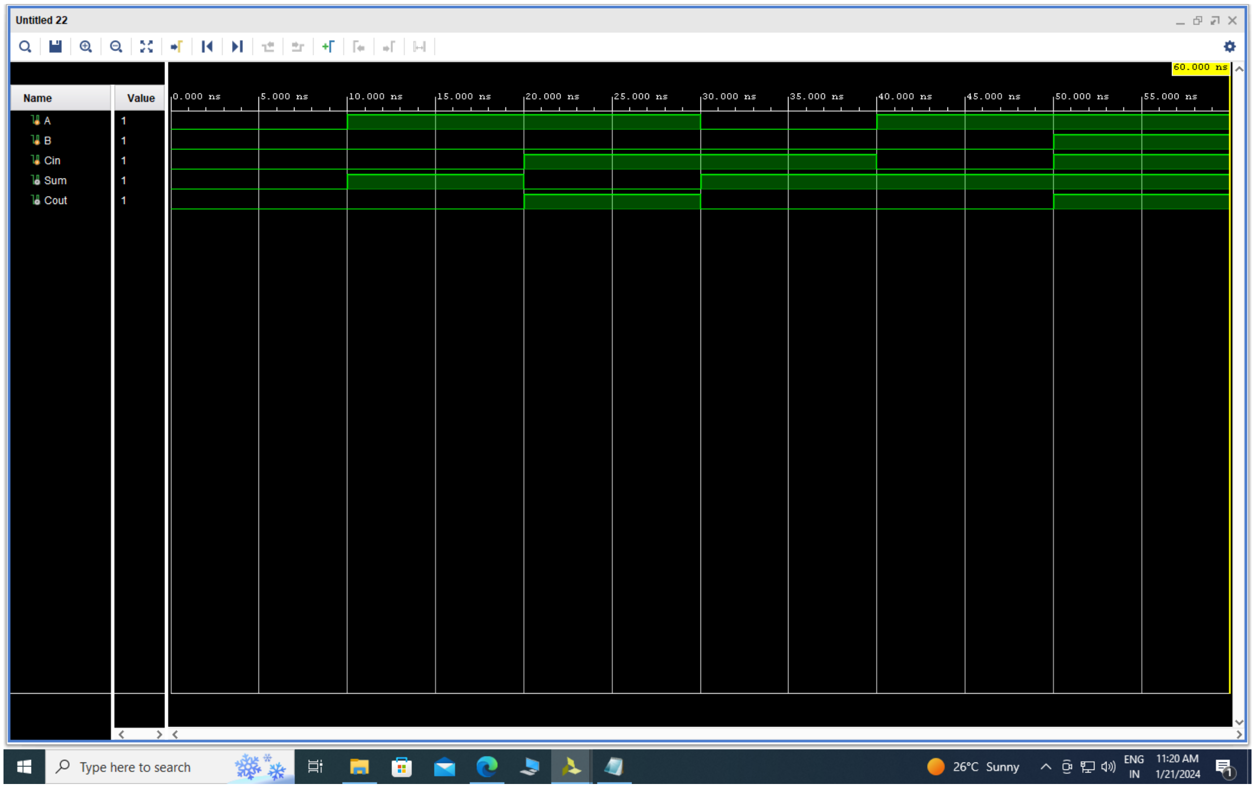
First, we implement the 1 bit Full Adder :

```
Full Adder
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 09:07:38 AM
// Design Name:
// Module Name: FullAdder
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module FullAdder(
    input A,
    input B,
    input Cin,
    output Sum,
    output Cout
    );
    and a1(f, Cin,A);
    and a2(g, A, B);
    and a3(h, B, Cin);
    or a4(Cout, f, g, h);

    xor a5(i, A, B);
    xor a6(Sum,i,Cin);

endmodule
```

```
TestBench for full Adder
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 10:18:59 AM
// Design Name:
// Module Name: FullAdder_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module FullAdder_tb();
```

```
    reg A, B, Cin;
    wire Sum, Cout;
    FullAdder uut(A, B, Cin, Sum, Cout);
    initial begin
    A = 0; B = 0; Cin = 0;
    #10;
    A = 1; B = 0; Cin = 0;
    #10;
    A = 1; B = 0; Cin = 1;
    #10;
    A = 0; B = 0; Cin = 1;
    #10;
    A = 1; B = 0; Cin = 0;
    #10;
    A = 1; B = 1; Cin = 1;
    #10;

    $finish();
    end
endmodule
```

Simulation Results :



Clearly, Simulation shows the expected output.

8 Bit Adder Subtractor

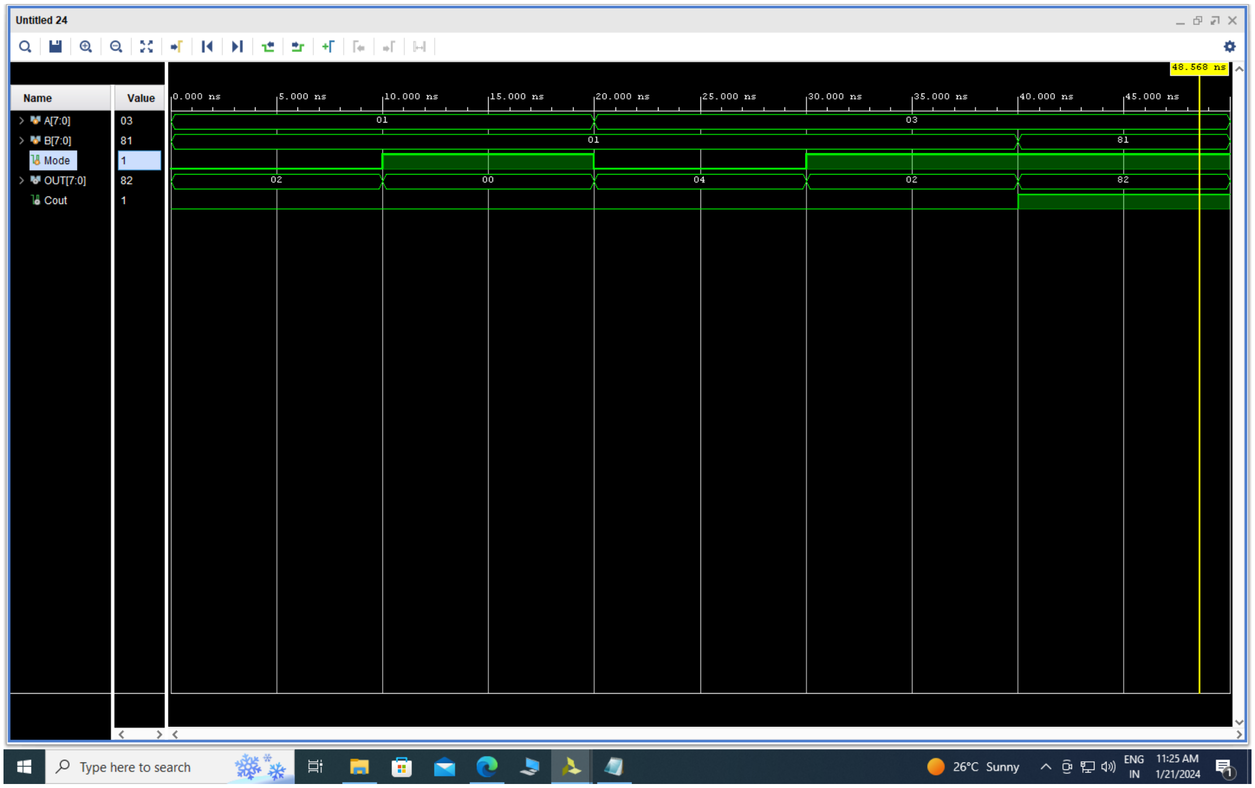
Now, we implement 8 Bit Adder Subtractor using the Full Adder module :

```
adder subtractor
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 09:13:45 AM
// Design Name:
// Module Name: AdderSubtractor8b
// Project Name:
// Target Devices:
```

```
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module AdderSubtractor8b(
    input [7:0] A,
    input [7:0] B,
    input Mode,
    output [7:0] OUT,
    output Cout
    );
    wire [7:0] Cout1;
    wire [7:0] Cout2;
    wire [7:0]B_;
    wire [7:0]B__;
    wire [7:0]Bf;
    wire [7:0] one;



    not n1(B_[0], B[0]);
    not n2(B_[1], B[1]);
    not n3(B_[2], B[2]);
    not n4(B_[3], B[3]);
    not n5(B_[4], B[4]);
    not n6(B_[5], B[5]);
    not n7(B_[6], B[6]);
    not n8(B_[7], B[7]);
    FullAdder FA1_ (.A(1), .B(B_[0]), .Cin(0), .Sum(B__[0]), .C
    FullAdder FA2_ (.A(0), .B(B_[1]), .Cin(Cout1[0]), .Sum(B__[
    FullAdder FA3_ (.A(0), .B(B_[2]), .Cin(Cout1[1]), .Sum(B__[
    FullAdder FA4_ (.A(0), .B(B_[3]), .Cin(Cout1[2]), .Sum(B__[
    FullAdder FA5_ (.A(0), .B(B_[4]), .Cin(Cout1[3]), .Sum(B__[
    FullAdder FA6_ (.A(0), .B(B_[5]), .Cin(Cout1[4]), .Sum(B__[
    FullAdder FA7_ (.A(0), .B(B_[6]), .Cin(Cout1[5]), .Sum(B__[
    FullAdder FA8_ (.A(0), .B(B_[7]), .Cin(Cout1[6]), .Sum(B__[

    assign Bf = (Mode == 0) ? B : B__;


    FullAdder FA1 (.A(A[0]), .B(Bf[0]), .Cin(0), .Sum(OUT[0]),
    FullAdder FA2 (.A(A[1]), .B(Bf[1]), .Cin(Cout2[0]), .Sum(OU
    FullAdder FA3 (.A(A[2]), .B(Bf[2]), .Cin(Cout2[1]), .Sum(OU
    FullAdder FA4 (.A(A[3]), .B(Bf[3]), .Cin(Cout2[2]), .Sum(OU
    FullAdder FA5 (.A(A[4]), .B(Bf[4]), .Cin(Cout2[3]), .Sum(OU
    FullAdder FA6 (.A(A[5]), .B(Bf[5]), .Cin(Cout2[4]), .Sum(OU
    FullAdder FA7 (.A(A[6]), .B(Bf[6]), .Cin(Cout2[5]), .Sum(OU
    FullAdder FA8 (.A(A[7]), .B(Bf[7]), .Cin(Cout2[6]), .Sum(OU

    assign Cout = (Mode == 0) ? Cout2[7] : ~Cout2[7];

endmodule
```

```
TestBench Adder Subtractor
`timescale 1ns / 1ps
///////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 10:12:41 AM
// Design Name:
// Module Name: AdderSubtractor8b_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module AdderSubtractor8b_tb();
    reg [7:0] A;
    reg [7:0] B;
    reg Mode;
    wire [7:0] OUT;
    wire Cout;
    AdderSubtractor8b uut(A, B, Mode, OUT, Cout);
    initial
    begin
    A = 8'b00000001; B = 8'b00000001; Mode = 0;
    #10;
    A = 8'b00000001; B = 8'b00000001; Mode = 1;
    #10;
    A = 8'b00000011; B = 8'b00000001; Mode = 0;
    #10;
    A = 8'b00000011; B = 8'b00000001; Mode = 1;
    #10;
    A = 8'b00000011; B = 8'b10000001; Mode = 1;
    #10;
    $finish();
    end
endmodule
```

Simulation Results :



We can see the module has been implemented correctly.

Gray Code Converter

Now, we move on to building the 9 Bit gray code converter :

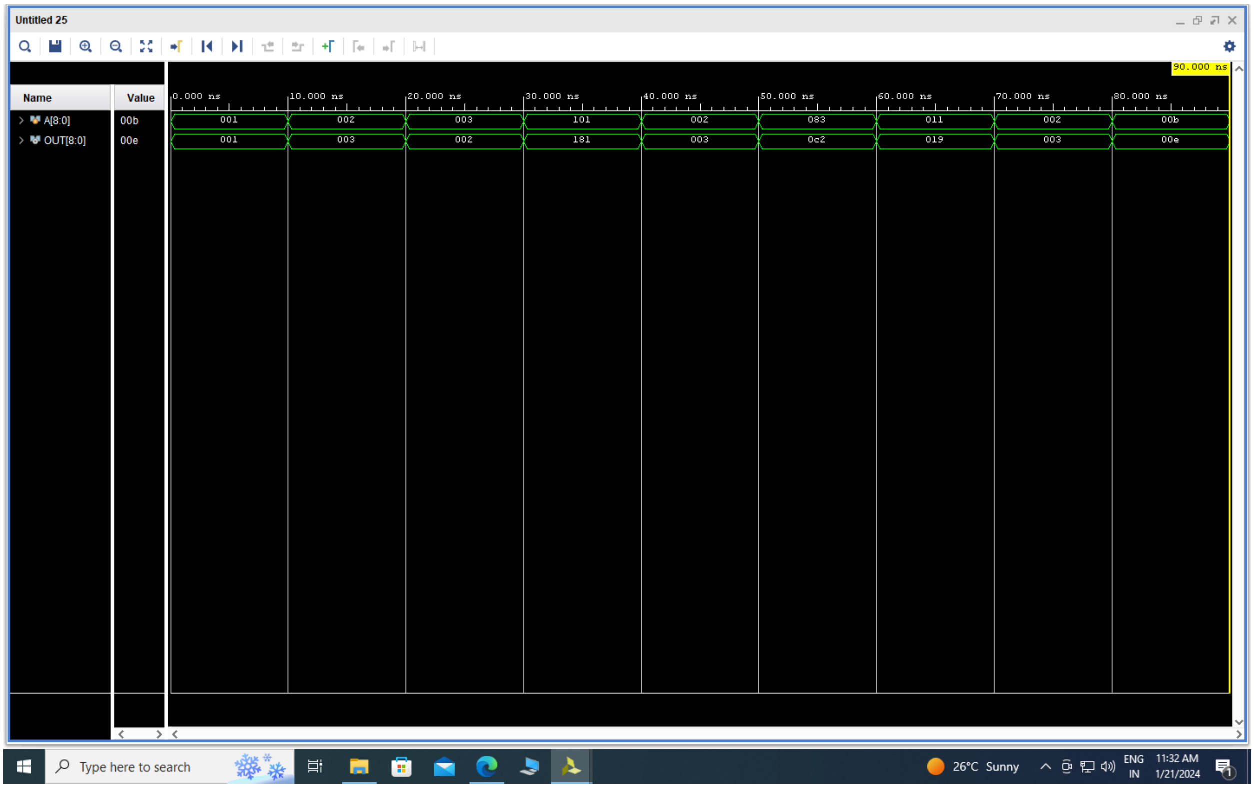
```
Gray Converter
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
```

```
// Create Date: 01/21/2024 09:18:01 AM
// Design Name:
// Module Name: GrayConverter
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module GrayConverter(
    input [9:0] A,
    output [9:0] OUT
    );
    assign OUT[8] = A[8];
    xor a1(OUT[7], A[8], A[7]);
    xor a2(OUT[6], A[7], A[6]);
    xor a3(OUT[5], A[6], A[5]);
    xor a4(OUT[4], A[5], A[4]);
    xor a5(OUT[3], A[4], A[3]);
    xor a6(OUT[2], A[3], A[2]);
    xor a7(OUT[1], A[2], A[1]);
    xor a8(OUT[0], A[1], A[0]);
endmodule
```

```
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
```

```
// Engineer:
//
// Create Date: 01/21/2024 10:07:19 AM
// Design Name:
// Module Name: GreyConverter_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module GreyConverter_tb();
    reg [8:0] A;
    wire [8:0] OUT;
    GrayConverter uut(A, OUT);
    initial
    begin
    A = 9'b000000001;
    #10;
    A = 9'b000000010;
    #10;
    A = 9'b000000011;
    #10;
    A = 9'b100000001;
    #10;
    A = 9'b000000010;
    #10;
    A = 9'b010000011;
    #10;
    A = 9'b000010001;
    #10;
    A = 9'b000000010;
    #10;
    A = 9'b000001011;
    #10;
    $finish();
    end
endmodule
```

Simulation Results :



Simulation results are as expected.

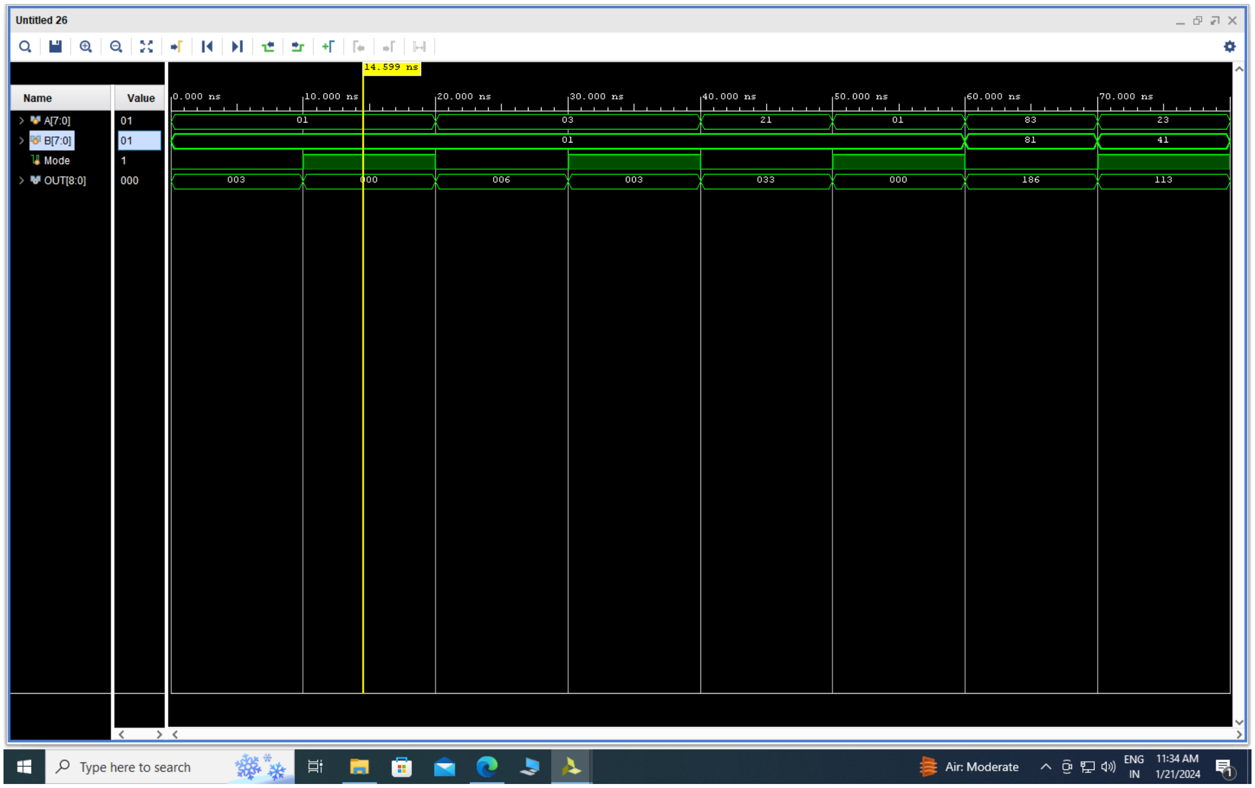
Combined Adder Subtractor Gray Converter

Now, we combine 8 Bit adder subtractor and gray converter to implement the final module

```
Combined Adder Subtractor Gray Converter
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 09:21:13 AM
// Design Name:
// Module Name: Combined
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module Combined(
    input [7:0] A,
    input [7:0] B,
    input Mode,
    output [8:0] OUT
    );
    wire Cout;
    wire [7:0] adderOut;
    wire [8:0]adderOut9;
    AdderSubtractor8b a1(.A(A), .B(B), .Mode(Mode), .OUT(adderOu
    assign adderOut9[7:0] = adderOut;
    assign adderOut9[8] = Cout;
    GrayConverter g1(.A(adderOut9), .OUT(OUT));

endmodule
```

```
TestBench
`timescale 1ns / 1ps
////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 01/21/2024 09:55:00 AM
// Design Name:
// Module Name: Combined_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
////////////////////////////////////////////////////////////////


module Combined_tb();
    reg [7:0] A;
    reg [7:0] B;
    reg Mode;
    wire [8:0] OUT;
    Combined uut( A, B, Mode, OUT);
    initial
    begin
    A = 8'b00000001; B = 8'b00000001; Mode = 0;
    #10;
    A = 8'b00000001; B = 8'b00000001; Mode = 1;
    #10;
    A = 8'b00000011; B = 8'b00000001; Mode = 0;
    #10;
    A = 8'b00000011; B = 8'b00000001; Mode = 1;
    #10;
    A = 8'b00100001; B = 8'b00000001; Mode = 0;
    #10;
    A = 8'b00000001; B = 8'b00000001; Mode = 1;
    #10;
    A = 8'b10000011; B = 8'b10000001; Mode = 0;
    #10;
    A = 8'b00100011; B = 8'b01000001; Mode = 1;
    #10;

    $finish();
    end
endmodule
```

Simulation Results :



We can see the final module works correctly. This concludes the documentation.