LAB REPORT
CSE4010 Computer Architecture
Instructor: Lawrence Orijuela

Name: Daniel Delgado Acosta SCORE: _ __/30 Student ID: 006896598 DUE: 4/18/23 LAB: Lab3

Report

o ***How can addition and subtraction of large numbers be performed by computers if they are only aware of numbers 0 and 1?***

Using logic gates in a specific sequence in the form of circuitry, computers are able to add or subtract using binary. On the physical level, computers are a multitude of transistors and resistors used to keep track of on and off switches. Logic gates are used to create functions allowing computers to do arithmetic such as adding and subtracting.

Source Code for Parts A and B

Part A

fullAdder.v

```
//initialize wires
module halfAdder(op1, op2, sum, carry);


    //input wires
    input op1, op2;
    //output wires
    output sum, carry;


    //wire sum will be the same as the exclusive or of wires op1 and op2
    assign sum = op1 ^ op2; // "^" = XOR operator
    //wire carry will be the same as the and of wires op1 and op2
    assign carry = op1 & op2; // "&" = AND operator


endmodule


module fullAdder(A, B, carryIn, sum, carryOut);


    //input and output wires
    input A, B, carryIn;
    output sum, carryOut;
```

```
    //intermediary wires
    wire c, d, e;

    //half adder wire definitions for full adder
    halfAdder u1(A, B, c, d);
    halfAdder u2(carryIn, c, e, f);

    //wire carryOut will be the same as the or of wires f and d
    assign carryOut = f | d;
    //wire sum will be the same as wire e
    assign sum = e;

endmodule
```

fullAdder_tb.v

```
//creates time limit and include fullAdder.v file
`timescale 1 ns / 1 ns
`include "fullAdder.v"

//creates test bench
module fullAdder_tb;
```

```
//denoted wires
reg A, B, carryIn;
wire sum, caryOut;

//creates instance of fullAdder
fullAdder uut(A, B, carryIn, sum, carryOut);

initial begin

    //creates a separate .vvp file and .vcd file for outputs
    $dumpfile("fullAdder_tb.vcd");
    $dumpvars(0, fullAdder_tb);

    //fullAdder gate
    {A, B, carryIn} = 3'd0; #20;
    {A, B, carryIn} = 3'd1; #20;
    {A, B, carryIn} = 3'd2; #20;
    {A, B, carryIn} = 3'd3; #20;
    {A, B, carryIn} = 3'd4; #20;
    {A, B, carryIn} = 3'd5; #20;
    {A, B, carryIn} = 3'd6; #20;
```

```
    {A, B, carryIn} = 3'd7; #20;
    //displays message to terminal
    $display("Finished additions!");
end
endmodule
```

Part B

fullSubtractor.v

```
//initialize wires
module halfSubtractor(op1, op2, diff, borrow);



    //input wires
    input op1, op2;
    //output wires
    output diff, borrow;



    //wire diff will be the same as the exclusive or of wires op1 and op2
    assign diff = op1 ^ op2; // "^" = XOR operator
    //wire borrow will be the same as the and of wires not op1 and op2
    assign borrow = !op1 & op2; // "&" = AND operator, "!" = NOT operator



endmodule


module fullSubtractor(A, B, Bin, diff, Bout);
```

```
    //input and output wires
    input A, B, Bin;
    output diff, Bout;



    //intermediary wires
    wire c, d, e;



    //half subtractor wire definitions for full subtractor
    halfSubtractor u1(A, B, c, d);
    halfSubtractor u2(Bin, c, e, f);



    //wire Bout will be the same as the or of wires f and d
    assign Bout = f | d;
    //wire diff will be the same as wire e
    assign diff = e;

endmodule
```

fullSubtractor_tb.v

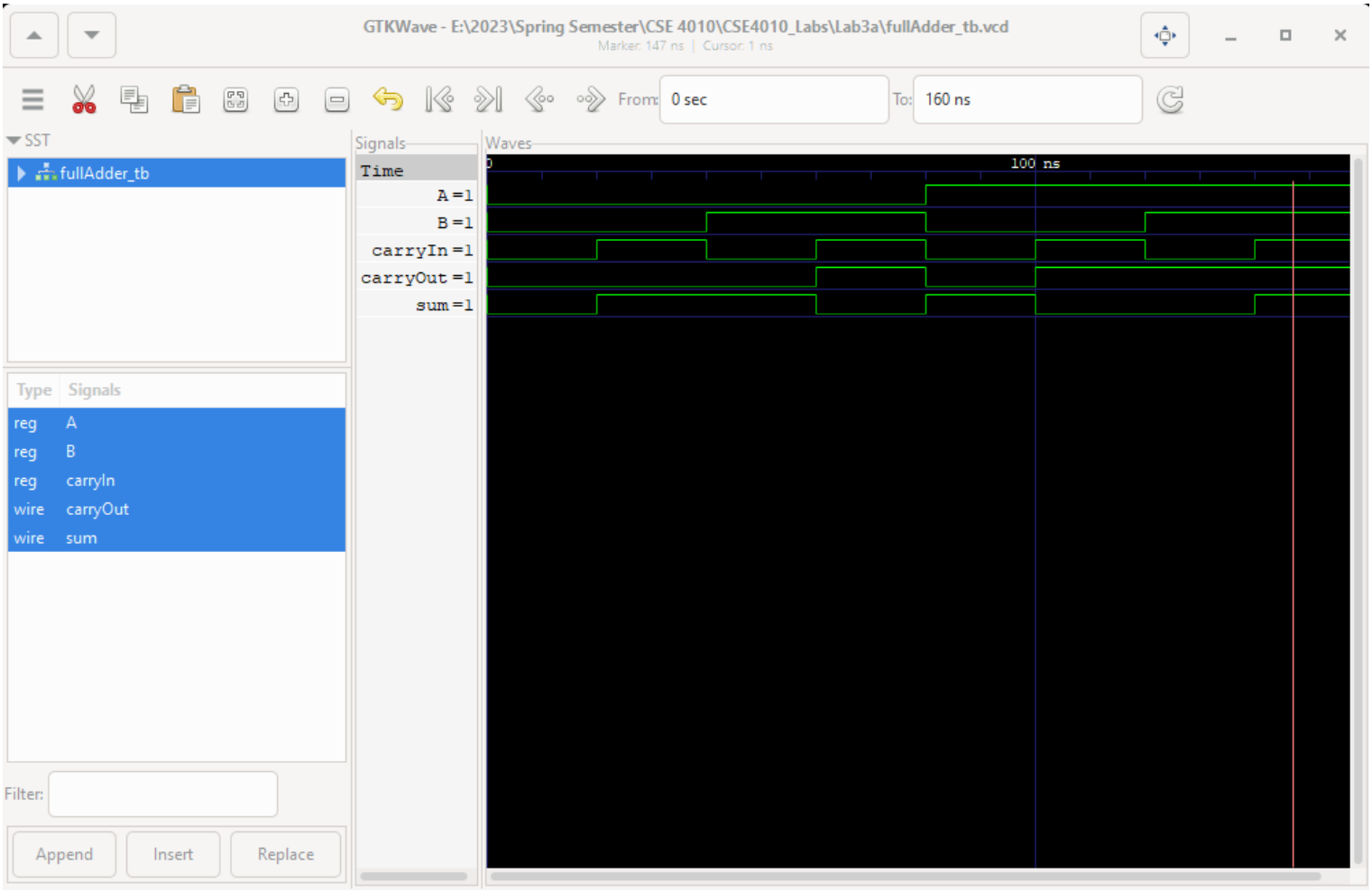
```
//creates time limit and include fullSubtractor.v file

`timescale 1 ns / 1 ns

`include "fullSubtractor.v"



//creates test bench

module fullSubtractor_tb;



//denoted wires

reg A, B, Bin;

wire diff, Bout;



//creates instance of fullSubtractor

fullSubtractor uut(A, B, Bin, diff, Bout);



initial begin



    //creates a separate .vvp file and .vcd file for outputs

    $dumpfile("fullSubtractor_tb.vcd");

    $dumpvars(0, fullSubtractor_tb);
```

```
    //fullSubtractor gate
    {A, B, Bin} = 3'd0; #20;
    {A, B, Bin} = 3'd1; #20;
    {A, B, Bin} = 3'd2; #20;
    {A, B, Bin} = 3'd3; #20;
    {A, B, Bin} = 3'd4; #20;
    {A, B, Bin} = 3'd5; #20;
    {A, B, Bin} = 3'd6; #20;
    {A, B, Bin} = 3'd7; #20;
    //displays message to terminal
    $display("Finished subtractions!");
end

endmodule
```

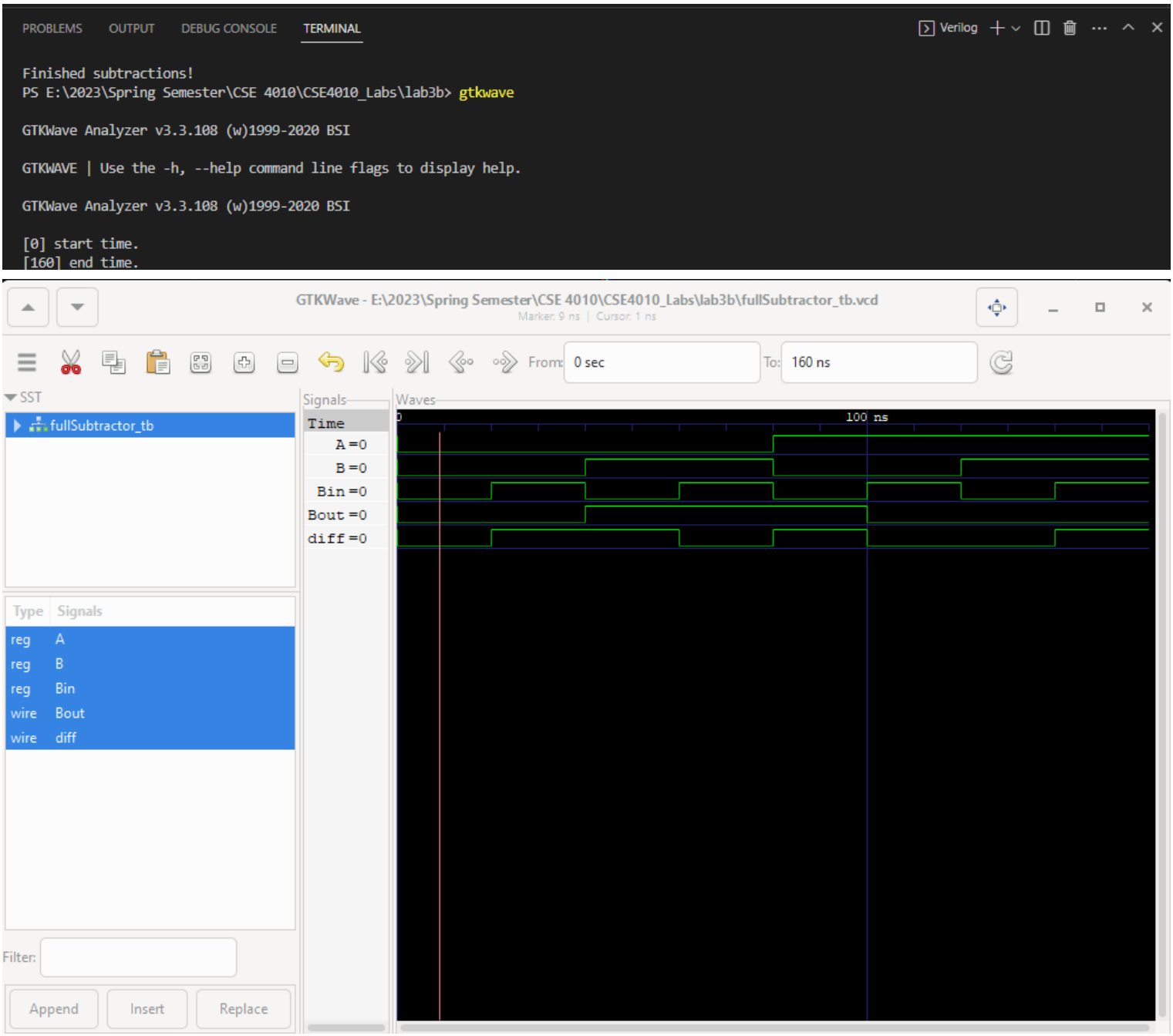
Screenshots for Parts A and B

Part A

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                    Verilog

Finished additions!
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\Lab3a> gtkwave

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

GTKWAVE | Use the -h, --help command line flags to display help.

GTKWave Analyzer v3.3.108 (w)1999-2020 BSI

[0] start time.
[160] end time.
```



Part B


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
Verilog
Finished subtractions!
PS E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\lab3b> gtkwave
GTKWave Analyzer v3.3.108 (w)1999-2020 BSI
GTKWAVE | Use the -h, --help command line flags to display help.
GTKWave Analyzer v3.3.108 (w)1999-2020 BSI
[0] start time.
[160] end time.
GTKWave - E:\2023\Spring Semester\CSE 4010\CSE4010_Labs\lab3b\fullSubtractor_tb.vcd
From: 0 sec
To: 160 ns
SST
fullSubtractor_tb
Type Signals
reg A
reg B
reg Bin
wire Bout
wire diff
Filter:
Append
Insert
Replace
Signals
Time
A =0
B =0
Bin =0
Bout =0
diff =0
Waves
100 ns