CSE4010 – Computer Architecture

Lab 3 – Half/Full Adder(s) and Subtractor(s)

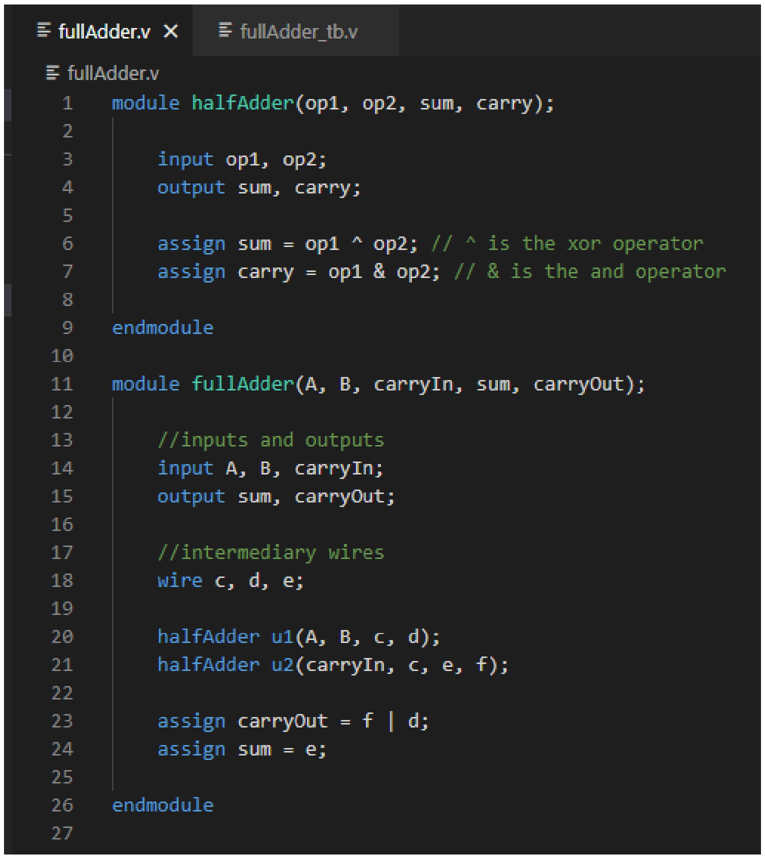
Introduction

Have you ever added two plus two on a calculator and it gave you 'four' as the answer? Have you ever wondered how the calculator is able to do this?Calculators, as well as all computers, only understand binary numbers (1 and 0), so how can a computer perform complex math?

The answer lies in its circuitry.We construct half-adders and half-subtractors which are the simplest circuits we can construct for performing addition and subtraction of two binary bits. We then use several of these connected to one another to create adders and subtractors of much larger numbers.

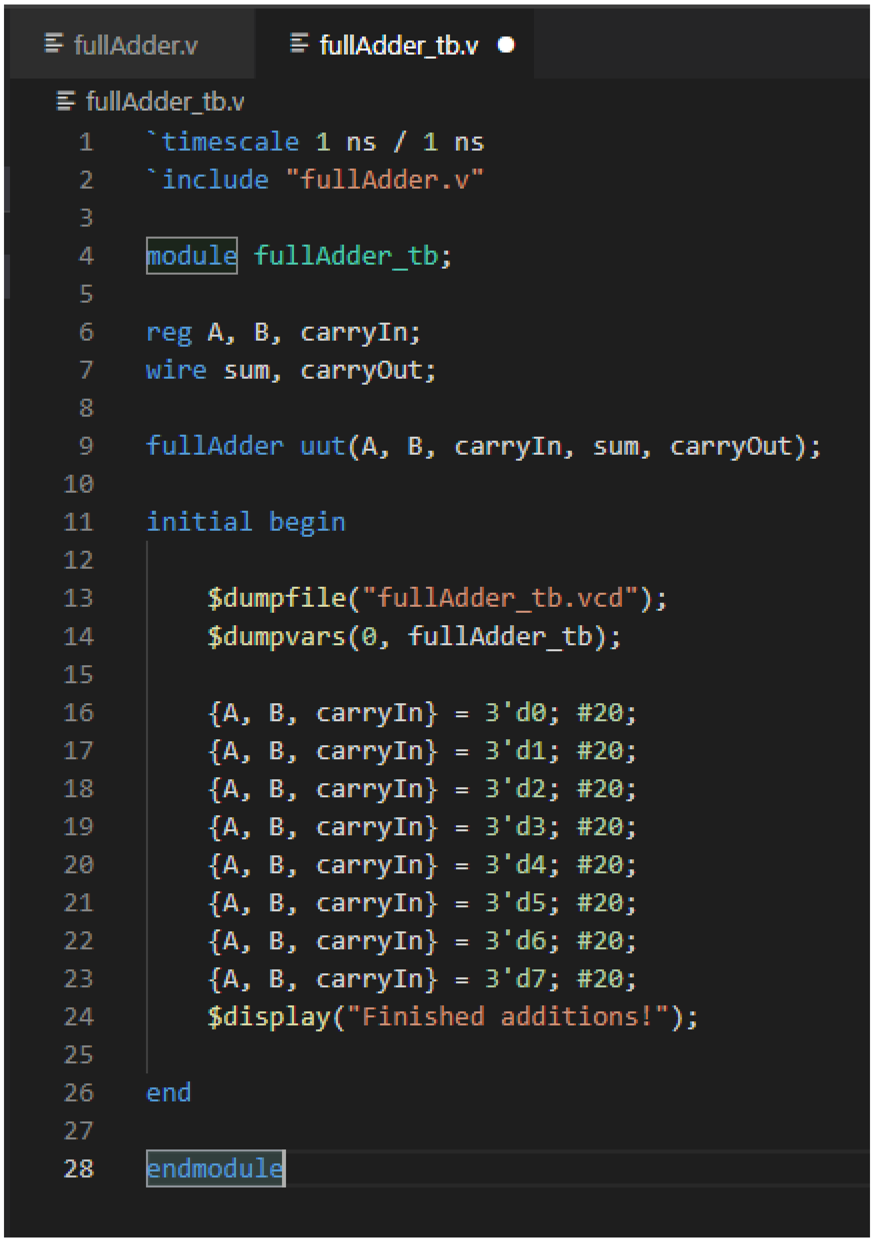
Instructions

1. Review the accompanying PowerPoint for this lab. Observe how the half adder and a full adder are constructed. From the diagrams you can convert it into Verilog code.
2. Open Visual Studio Code. Click *Open Folder* and select the *Lab3a* folder (that you created in Lab 1).
3. Create the module fullAdder.v.
    a. Hover your cursor over the *Lab3a* folder in the Explorer Pane and click the *New File* button. Name the new file 'fullAdder.v'.
4. Type the following code. Enter comments explaining what the code does. **CODE WITH NO COMMENTS GET NO CREDIT.**



```
module halfAdder(op1, op2, sum, carry);

    input op1, op2;
    output sum, carry;

    assign sum = op1 ^ op2; // ^ is the xor operator
    assign carry = op1 & op2; // & is the and operator

endmodule

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

    //inputs and outputs
    input A, B, carryIn;
    output sum, carryOut;

    //intermediary wires
    wire c, d, e;

    halfAdder u1(A, B, c, d);
    halfAdder u2(carryIn, c, e, f);

    assign carryOut = f | d;
    assign sum = e;

endmodule
```

5. Create the testbench.
    a. Hover your cursor over the *Lab3a* folder in the Explorer Pane and click the *New File* button. Name the new file ‘fullAdder_tb.v’
    b. Type the following code. Enter comments explaining what the code does. **CODE WITH NO COMMENTS GET NO CREDIT.**



```
`timescale 1 ns / 1 ns
`include "fullAdder.v"

module fullAdder_tb;

reg A, B, carryIn;
wire sum, carryOut;

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

initial begin

    $dumpfile("fullAdder_tb.vcd");
    $dumpvars(0, fullAdder_tb);

    {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;
    $display("Finished additions!");

end

endmodule
```

6. Compile, Run, and Examine the waveform.
    a. Open VSC’s terminal using ‘Ctrl + `’.
    b. Enter the command iverilog -o fullAdder_tb.vvpfullAdder_tb.v
    c. Enter the command vvpfullAdder_tb.vvp
7. Enter the command gtkwave and examine the waveform. Does it match the truth-table of a full-adder?

Part B

1. If your *Lab3a* folder is still open, you can close it by selecting the folder and clicking File > Close Folder.
2. Create a new folder in your *CSE4010_Labs* directory and name it *Lab3b*. Open this new folder in VS Code.
3. In Part A, you used the diagrams in the powerpoint to construct a half-adder and a full-adder. This time you will use the diagrams in the same powerpoint to construct a half-subtractor and a full-subtractor.
    a. Use the diagrams. Create a module for a half-subtractor taking two inputs (*op1* and *op2*) and two outputs (*difference* and *borrow*). Assign *difference* to be the XOR of op1 and op2. Assign*borrow* to be the AND of !op1 and op2.
    b. Use the diagrams. Create a module for a full-subtractor taking inputs A, B, and BIn and outputs Difference and Bout. Construct it using two half-subtractors and an OR gate.
    c. Create a testbench similar to the one in Part A. Run and examine the waves, does it behave like a full-subtractor?
    d. These links might be helpful:
        i. https://www.geeksforgeeks.org/full-adder-in-digital-logic/?ref=lbp
        ii. https://www.geeksforgeeks.org/full-subtractor-in-digital-logic/?ref=lbp

Lab Report

Use the 'Lab Report Template' found on Blackboard/Canvas. Your lab report must contain the following:

- Report
    - How can addition and subtraction of large numbers be performed by computers of they are only aware of numbers 0 and 1?
- Source Code
    - fullAdder.v
    - fullAdder_tb.v
    - fullSubtractor.v
    - fullSubtractor_tb.v
- Screenshots
    - Screenshot from part A
    - Screenshot from part B