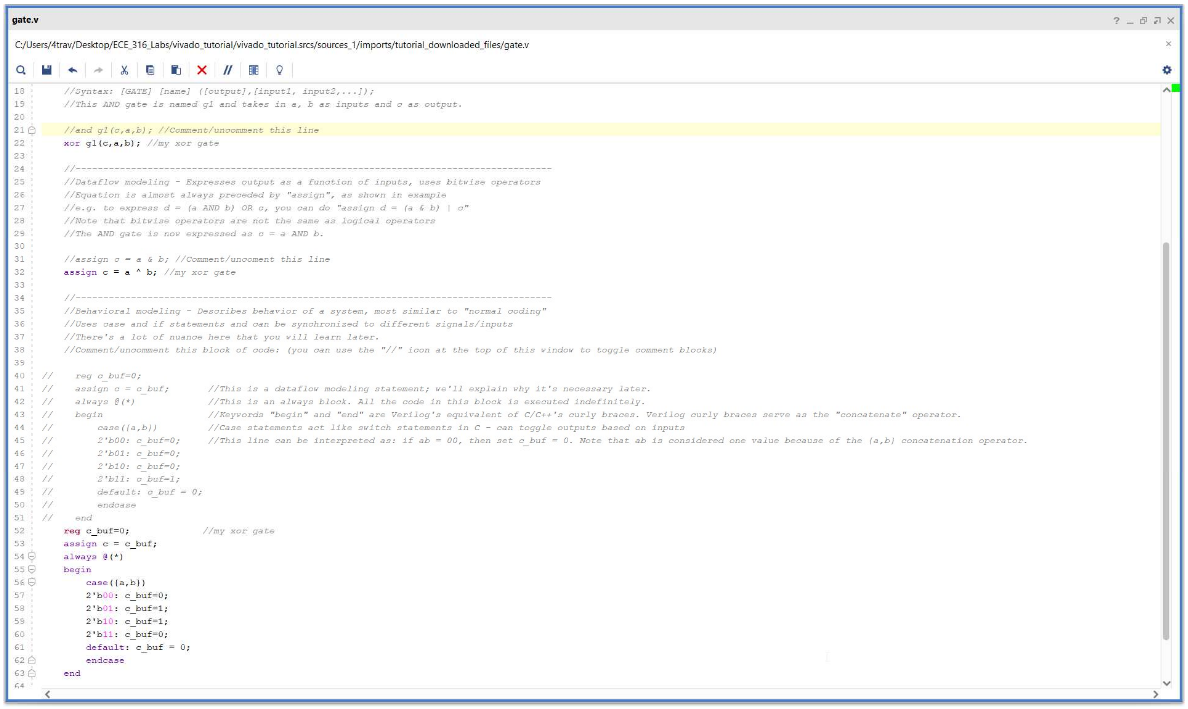
Vivado Tutorial Lab 1 – Travis Beach & Sam Lin

Edited code for xor gate

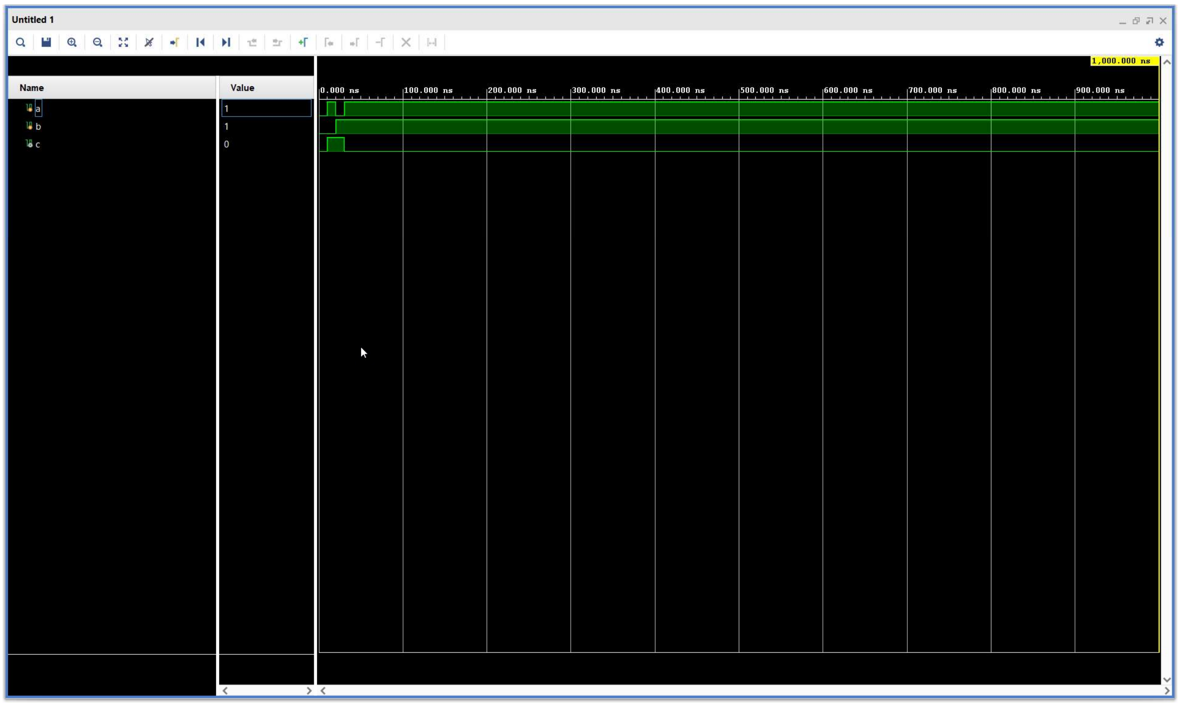


gate.v

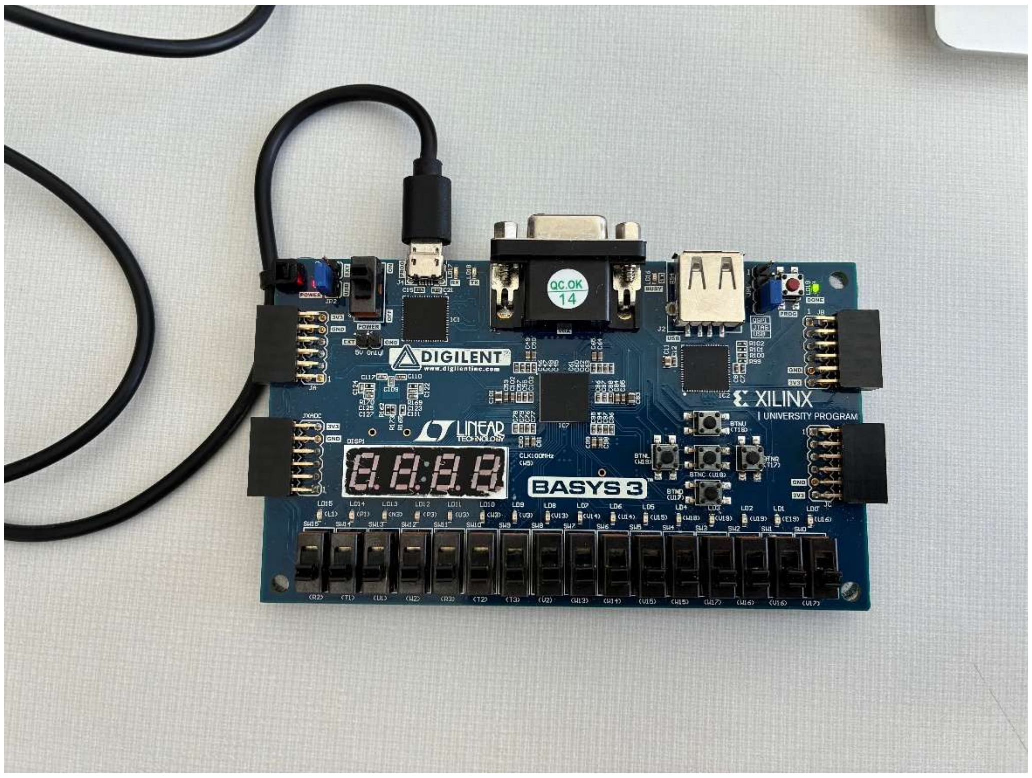
C:/Users/4trav/Desktop/ECE_316_Labs/vivado_tutorial/vivado_tutorial.srcs/sources_1/imports/tutorial_downloaded_files/gate.v

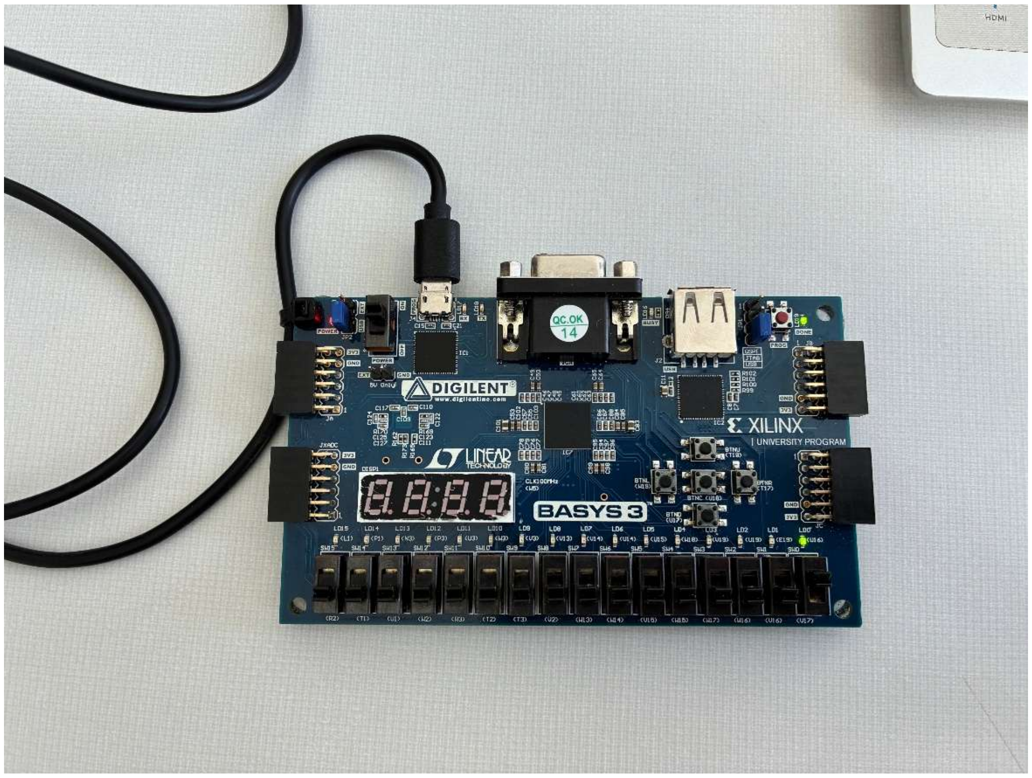
```verilog
    //Syntax: [GATE] [name] ([output],[input1, input2,...]);
    //This AND gate is named g1 and takes in a, b as inputs and c as output.

    //and g1(c,a,b); //Comment/uncomment this line
    xor g1(c,a,b); //my xor gate

    //-------------------------------------------------------------------------------------
    //Dataflow modeling - Expresses output as a function of inputs, uses bitwise operators
    //Equation is almost always preceded by "assign", as shown in example
    //e.g. to express d = (a AND b) OR c, you can do "assign d = (a & b) | c"
    //Note that bitwise operators are not the same as logical operators
    //The AND gate is now expressed as c = a AND b.

    //assign c = a & b; //Comment/uncoment this line
    assign c = a ^ b; //my xor gate

    //-------------------------------------------------------------------------------------
    //Behavioral modeling - Describes behavior of a system, most similar to "normal coding"
    //Uses case and if statements and can be synchronized to different signals/inputs
    //There's a lot of nuance here that you will learn later.
    //Comment/uncomment this block of code: (you can use the "//" icon at the top of this window to toggle comment blocks)

//    reg c_buf=0;
//    assign c = c_buf;       //This is a dataflow modeling statement; we'll explain why it's necessary later.
//    always @(*)             //This is an always block. All the code in this block is executed indefinitely.
//    begin                   //Keywords "begin" and "end" are Verilog's equivalent of C/C++'s curly braces. Verilog curly braces serve as the "concatenate" operator.
//        case({a,b})         //Case statements act like switch statements in C - can toggle outputs based on inputs
//        2'b00: c_buf=0;     //This line can be interpreted as: if ab = 00, then set c_buf = 0. Note that ab is considered one value because of the {a,b} concatenation operator.
//        2'b01: c_buf=0;
//        2'b10: c_buf=0;
//        2'b11: c_buf=1;
//        default: c_buf = 0;
//        endcase
//    end
    reg c_buf=0;            //my xor gate
    assign c = c_buf;
    always @(*)
    begin
        case({a,b})
        2'b00: c_buf=0;
        2'b01: c_buf=1;
        2'b10: c_buf=1;
        2'b11: c_buf=0;
        default: c_buf = 0;
        endcase
    end
```

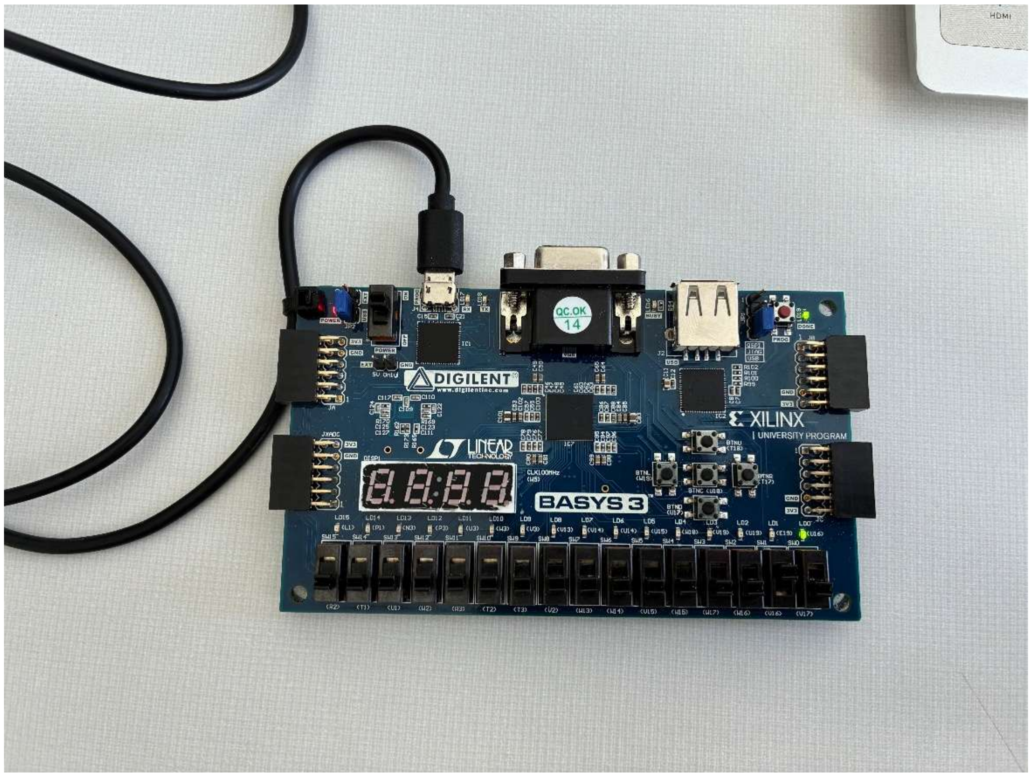
xor gate simulation

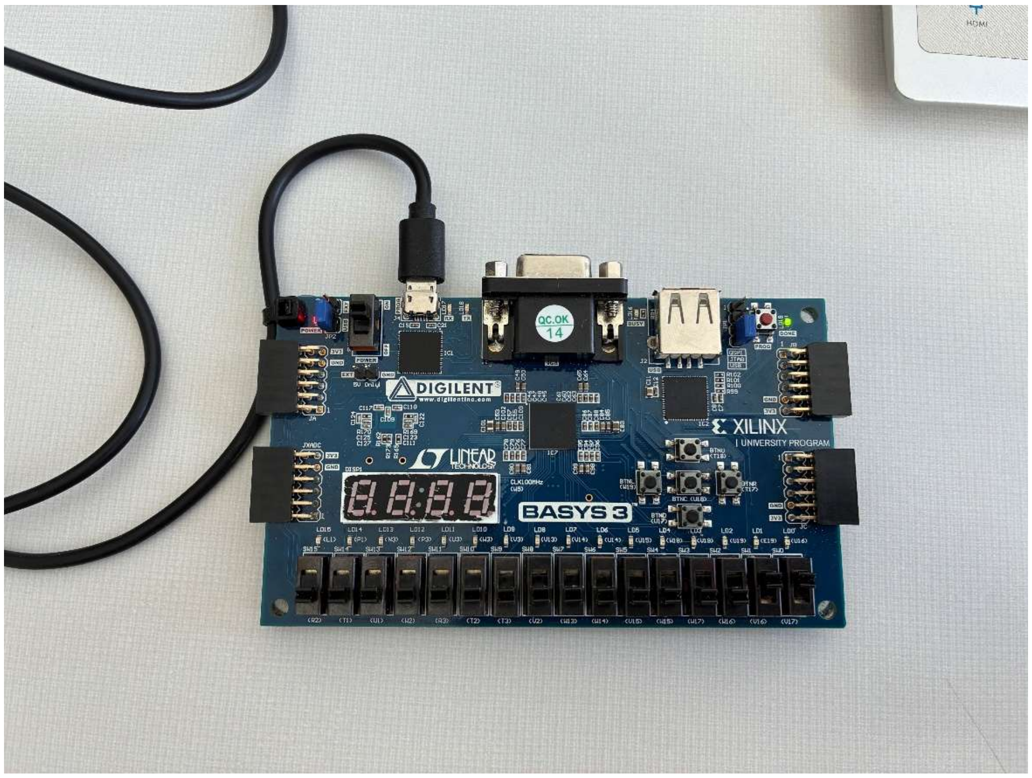


4 xor gate switch positions






DIGILENT
LINEAR TECHNOLOGY
BASYS 3
XILINX
UNIVERSITY PROGRAM
QC.OK 14


DIGILENT
LINEAR TECHNOLOGY
BASYS 3
XILINX
UNIVERSITY PROGRAM
QC.OK 14