ECE 449 - OBJECT ORIENTED COMPOTER SIMULATION

FALL 2014

PROJECT 3: NETLIST CONSTRUCTION

NAME: HARSHITA RAVI SHANKAR

CWID: A20327797

INTRODUCTION

The objective of Project three is to design a netlist. As of now, we have extracted tokens, sorted them into statements and lastly, created a netlist which contains signal nets that passes the signals around the component. Here, we have to simulate the circuit into ".sim_out" files that contain the output gate. Visual C++ is used to write the program and the golden simulator is used to test the result.

IMPLEMENTATION

This includes 4 header files, 3 cpp files and the main program file. They are as follows:

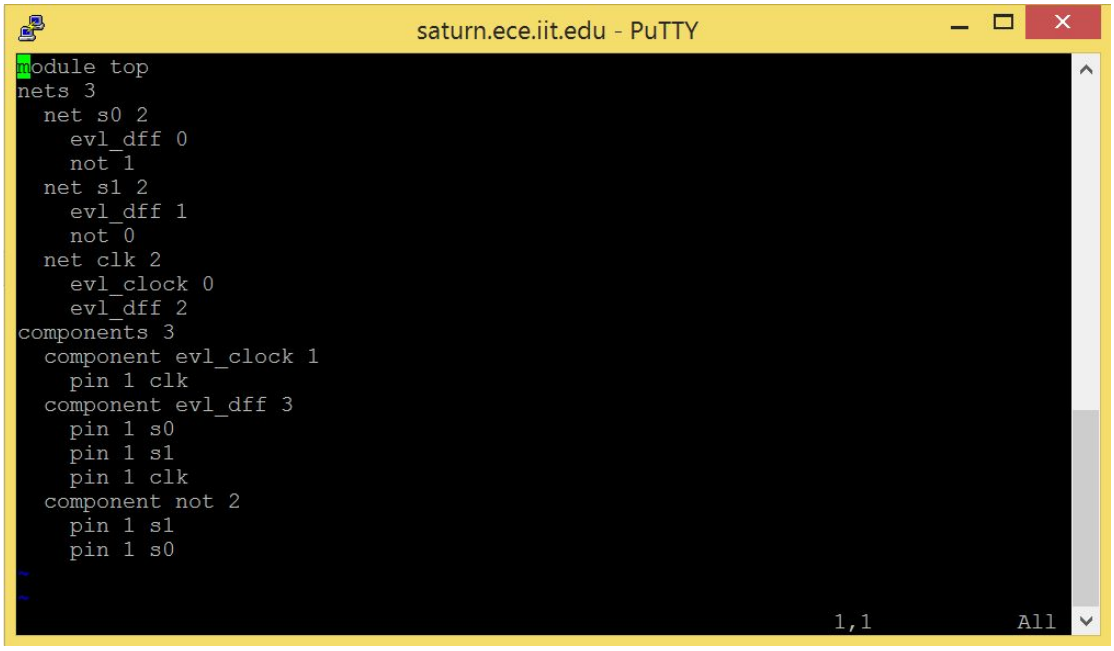
1. **Event header and source file**: Within the event header, priority scheduler is defined. Bucker priority scheduler is declared within this source file.
2. **Token header and source file**: The token header has been defined in project1. The structure evl_token has been defined with members. All the four functions of it are defined and the header just has to be included in the main cpp file.
3. **Statement header and source file**: The statement declaration has been programmed in project2. The only change made from project2 is that the function name declaration is done in header file and implemented in cpp files.
4. **Netlist header and source file**: The functions that were implemented in netlist.cpp files are declared within header file which has been included in cpp file. All the files are included in the .netlist file.
   - Combinational gates: They include AND, OR, XOR, NOT, BUF, ONE, ZERO. The boolean relations specifies the functionality between the inputs and outputs.
   - Circuit outputs: Output is declared as below: output name(in_1,in_2,...,in_k); Output file is written with extension file as ".sim_out".
   - Circuit inputs: If we are using input gate within test file we have to provide sim_in file as input.

5. **Main Program**: Within main program All header files are included within the main program and performed in project section all files are included.

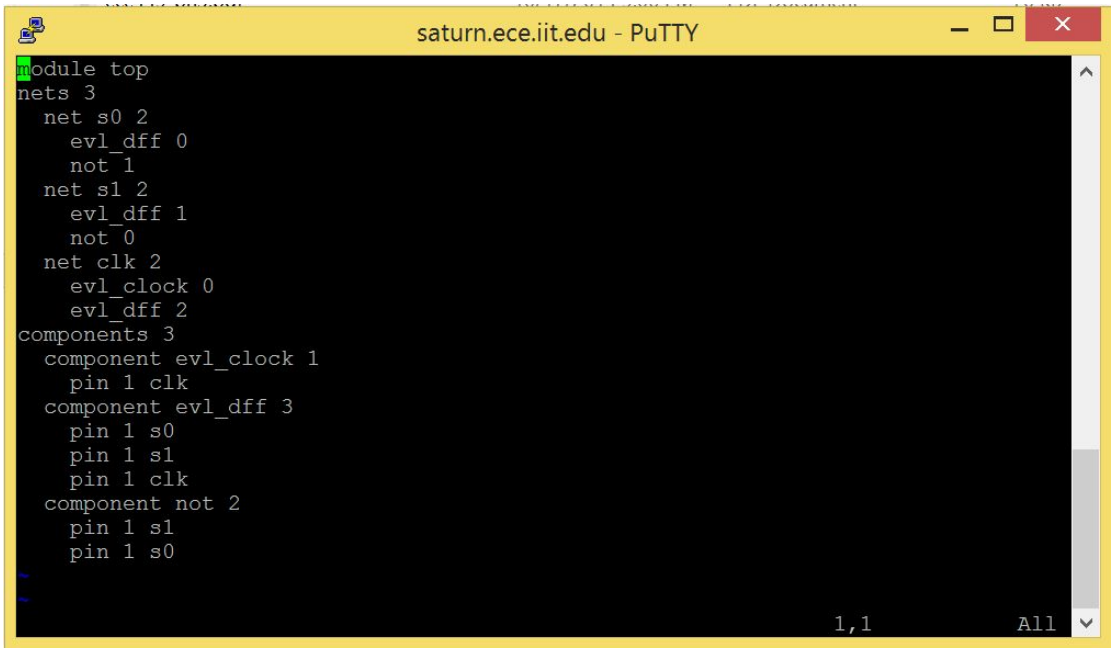
Testcases and the Respective Outputs

1. **Testcase1:** Testcase 1 along with the netlist created and the golden simulator output.

Testcase1



Netlist



Golden Simulator Output

2. **Testcase2**

```
module mux;
      wire s, a, b;
      wire sbar, a1, b1, y;
      input sim_in(s, a, b);
      output sim_out(sbar, a1, b1, y);
      not(sbar, s);
      and(sbar, a, a1);
      and(s, b, b1);
      or(y, a1, b1);
endmodule
```

Expected output is as follows:

```
4                       // No of Pins
pin 1 select            // Pin 1 width and net name
pin 1 in0               // Pin 2 width and net name
pin 1 in1               // Pin 3 width and net name
pin 1 out               // Pin 4 width and net name
0 1 0 1
0 1 0 1
***Continue goes to 1000 Cycles*********
```

CONCLUSION

The output is obtained as expected and it is verified using the golden simulator. The outputs obtained for all the testcases is similar to the output generation as given. All the project requirements have been met. All the testcases have been verified against the golden simulator and the result obtained was true.