Homework 5: Regular Expressions and Command-Line Perl

In this assignment, you will use the netlist rb_flat_synthesized.v that is in “/cad_area/vlsi/design/rb”. In your home directory, create a link to this file called “rb.v”.

1. Using the “ls -l” command followed by a grep, create a list of only links in your home directory (there may be only one: rb.v). Copy and paste the output in your submission document.

2. Verilog statements end in semicolons (;). Many of the Verilog statements in the netlist run over multiple lines. The line with a semicolon at the end represents last piece of the statement. Two consequent statements from a Verilog netlist are listed below to illustrate this point:

```
AOI22_X1 \u0_rb_offset/U6339  (.A1(\u0_rb_offset/n3119 ),
  .A2(\u0_rb_offset/n10641 ), .B1(\u0_rb_offset/cfg_reg [1229]),
  .B2(\u0_rb_offset/n10642 ), .ZN(\u0_rb_offset/n10645 ));
INV_X2 \u0_rb_offset/U1640  (.A(\u0_rb_offset/n10888 ),
  .ZN(\u0_rb_offset/n5975 ));
```

   a. Modify the netlist so that there is no line break in any Verilog statement. Hint: you can replace the new line character (“\n”) with a space, unless the line ends in a semicolon. Also, look out for “endmodule” statements. Save the modified netlist as “rb_flat_single_line.v”. Save this file in a directory called “rb” in your home directory.

   b. Using “grep” to get all the Verilog statements that instantiate flip-flops (flip-flop library cells have the string “DFF” in their names), generate a list of flip-flop instances and the name of the net or port connected to it’s clock pin. Save this in a file called “flop_inst_clock_node.txt” in the “rb” directory in your home directory. Insert a line with your name at the top of this file.

   c. Using sort and uniq commands (with appropriate options), list each node (net or port) connected to flip-flop clock pins with a count of the number of clock pins it is connected to.

   d. List nodes connected to flip-flop reset pins (each node must appear only once in the list) with a count of the number of flip-flop reset pins connected to the node.

3. Write two Verilog modules that instantiate AND and NOR gates to implement the AOI22 function. Call the two modules “SPAOI_X1 and SPAOI_X2. The port names for the modules should be the same as the corresponding pin names in AOI gate. Use X2 drive strength for the AND gates in both modules. Use X1 drive strength for the OR gate in SPAOI_X1, and use X4 drive strength for the OR gate in SPAOI_X2.

   a. Copy the netlist “rb_flat_single_line.v “ in your “rb” directory, naming the new file “rb_flat.v”. Append the two modules SPAOI_X1 and SPAOI_X2 to the bottom of “rb_flat_mode.v”.

   b. Using command-line Perl with appropriate regular expressions, substitute all AOIX1 cells with SPAOI_X1, and all AOI cells with drive-strength greater than X1 with SPAOI_X2. Save this modified netlist as “rb_flat_mod.v”.

   c. Submit the entire pipelined or unpipelined Unix command that you used to generate rb_flat_mod.v.