Computer Architecture ECE314
Winter 2004 Test 1

1. The fundamental gates are NOT, AND, OR, XOR. Draw and label the corresponding logic symbols and Verilog operators? (8 points)

2. Identify the following terms or abbreviations (2 points each)

   (a) ASCII

   (b) BCD

   (c) MSB

3. Convert 116_{10} to octal, hexadecimal and binary. (6 points)

4. Convert the binary number 00101011 to an 8-bit 2's-complement negative number. (3 points)

5. What are the differences between a flip-flop and a latch? (2 points)

6. Explain the hierarchical nature of a computer system. (5 points)

7. What common computer operation or principle is expressed by a 2:1 multiplexor? (2 points)

8. Define or explain the following terms (3 points each)

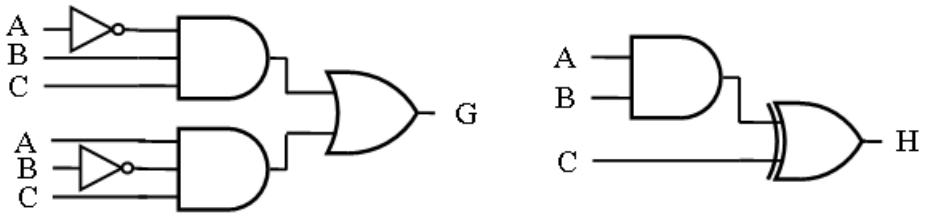
   (a) tri-state buffer

   (b) glitch

   (c) master-slave flip-flop

   (d) finite state machine

9. Write a single Verilog module to perform the following concurrent operations? (6 points)

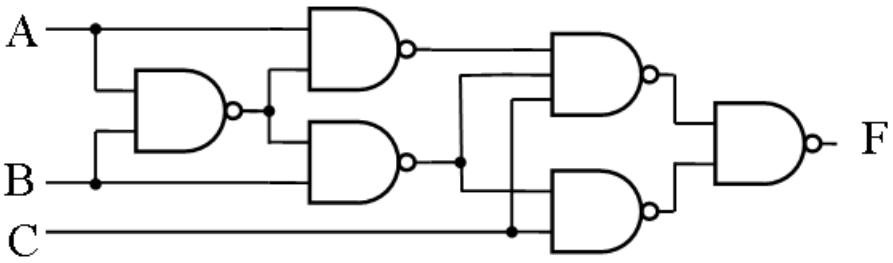


Computer Architecture ECE314
Winter 2004 Test 1

In each of the following problems, include comment lines at the start of each module with your name as author.

10. Write a Verilog module for a 4-bit subtractor ($F = A$ - B). Simulate and verify the following cases: 9 - 3, 14 - 8, and 4 - 11. Paste your module code and a snapshot of the simulation waveform into one page of a Word document and attach a print-out to your exam. (15 points)

11. Write a structural Verilog module for the following logic operation. Use MAXPLUS to compile your module and submit a one-page screen shot showing the Verilog source and compilation window. (15 points)



12. Write a behavioral Verilog module for a JK flip-flop using an **always** statement. Use MAXPLUS to insure that your module compiles without errors. Submit a one-page screen shot showing the Verilog source and compilation window. (15 points)

13. Rewrite the following module to correct the syntax errors, and produce a successful compilation with MAXPLUS. Submit a one-page screen shot as before. (5 points)

```
module Exam-1(A,B,C,D,F)
inputs A, B, C;
Output D,F;
and g1(A,B,D);
not g2(D,B,A);
OR  g3(F,B,C);
endmodule;
```