**CS 3173 Basic Computer Architecture Exam 3 100 points**

**Email your completed exam to me at** [**harringp@nsuok.edu**](mailto:harringp@nsuok.edu)

**Due 11/2 One exam submission per student**

**No late exams accepted Microsoft Word format only accepted**

**11 Questions**

**Part 1: Programming in Java:**

1. *(25 points)*

Write a Java program to get input from the user for the present state, input, and then show the next state based upon the excitation table for the bit counter shown below:

**Present State Input Next State**

**A x A**

--------------------------------------------------------------------

0 0 0

0 1 0

1 0 1

1 1 1

1. **import** java.util.Scanner;
3. **public** **class** Exam3 {
4. **public** **static** **void** main(String[] args) {
5. Scanner scan = **new** Scanner(System.in);
6. **boolean** a = **false**;
7. **boolean** input = **false**;
9. System.out.println("\nPlease enter \"true\" or \"false\"");
10. **for** (**int** i=0;i<2;i++) {
11. **if** (i==0) System.out.print("Present state of A = ");
12. **else** **if** (i==1) System.out.print("Input = ");
13. String in = scan.nextLine().toLowerCase();
14. **if** (in.equals("false")) {
15. **if** (i==0) a = **false**;
16. **else** **if** (i==1) input = **false**;
17. }
18. **else** **if** (in.equals("true")) {
19. **if** (i==0) a = **true**;
20. **else** **if** (i==1) input = **true**;
21. }
22. **else** {
23. System.out.println("Please enter \"true\" or \"false\"");
24. i--;
25. }
26. }
28. // Based on the excitation table, the next state always equal the present state.
29. System.out.println("\nPresent state: "+a+"\nInput: "+input+"\nNext state: "+a);
30. }
31. }

**Part 2: Questions and Problem Solving:**

1. *(5 points)* Solve the following bitwise AND operation: 0110 XOR 1010

**0110 XOR 1010 = 1100**

1. *(5 points)* Use a bitwise odd/even mask on the following binary number: 1010

**1010 AND 0001 = 0000**

1. *(5 points)* Use a bitwise lower nibble mask to isolate the lower four bits (lower nibble) of the following number: 10100111

**10100111 AND 00001111 = 00000111**

1. *(10 points)* Use a bitwise shift and lower nibble mask to isolate the upper four bits (upper nibble) of the following number: 10100111

**Shift: 10100111 > 00001010**

**Mask: 00001010 AND 00001111 = 00001010**

1. *(10 points)* Use bitwise OR with a bitwise mask to set bits 3 and 4 to one in the following number: 10100111

**10100111 OR 00011000 = 10111111**

1. *(10 points)* Use bitwise XOR with a bitwise mask to toggle bits 0 and 2 in the following number: 10100111

**10100111 XOR 00000101 = 10100010**

1. *(10 points)* Use the 1’s complement checksum to calculate the resulting value to check for errors in the following binary number: 10100111

**10100111 + 01011000 = 11111111**

1. *(10 points)* Use the 2’s complement checksum to calculate the resulting value to check for errors in the following binary number: 10100111

**10100111 + 01011001 = (1)00000000**

1. *(5 points)* What is the difference between a latch and a flip-flop?

**A latch is active-low while a flip-flop is active-high**

1. *(5 points)* What is the difference between and SR and JK flip flop?

**While an SR Flip Flop has undefined values, a JK Flip Flop does not.**