**CDA 3103 Exam 1 Practice**

**Problem 1. Multiple Choice**

Use module 1 and 2 quizzes for practice.

**Problem 2. Fill in Blanks**

1. Computer building components
2. Left and right shift operations
3. Sign extension operations
4. Data range
5. Data representations
6. Moore’s law and Rock’s law

**Problem 3. TRUE/FALSE**

Basic concepts are tested.

**Problem 4: Short Answer Questions**

Converting between bases, show detail conversion procedures

1. Convert unsigned binary number (1001 1001.101)2 to decimal number. Show your work.
2. Convert the **signed** octal number (312)8 to decimal. Show your work.
3. Convert the **signed** hexadecimal number (5AC9)16 to decimal. Show your work.
4. Convert the Octal number (512.713)8 to binary number
5. Convert the Hexadecimal number (BF16.79D)16 to binary number
6. Convert the binary number (1000110010010.1011101)2 to **Octal** number. Show your work.
7. Convert the binary number (1000110010010.1011101)2 to **Hexadecimal** number. Show your work.
8. How many bytes are 2 GB?
9. One processor runs at 1.9 GHz. How many nanoseconds is one clock cycle for this processor?
10. Perform the following additions of unsigned binary numbers. Indicate whether the sum overflows an 8-bit result.

1001 10012 + 1100 01002

1. The numbers shown below are in 2’s complement binary representation. Determine whether the indicated arithmetic operations produce an overflow.

1000 10002 – 0110 10102

1. Perform the addition of hexadecimal numbers: 9A5 + 27D
2. The sign-extend of 8 bits binary number 0 1 1 0 1 1 0 1 to 16 bits is \_\_\_\_\_\_\_\_\_\_\_.
3. The sign-extend of 8 bits binary number 1 0 0 1 1 0 0 1 to 16 bits is \_\_\_\_\_\_\_\_\_\_\_.
4. The result of left shift shifting 0 0 0 1 1 1 0 1 by 2 positions is \_\_\_\_\_\_\_\_\_\_\_\_\_.
5. The result of right shift shifting 0 0 0 1 1 1 0 1 by 3 positions is \_\_\_\_\_\_\_\_\_\_\_\_\_.
6. The result of right shift shifting 1 0 0 1 1 1 0 1 by 2 positions is \_\_\_\_\_\_\_\_\_\_\_\_\_.
7. The data range of n bits unsigned binary number representations is \_\_\_\_\_\_\_\_\_\_\_\_\_.
8. The data range of n bits sign-magnitude binary number representations is \_\_\_\_\_\_\_\_\_\_\_\_\_.
9. The data range of n bits 1’s complement binary number representations is \_\_\_\_\_\_\_\_\_\_\_\_\_.
10. The data range of n bits 2’s complement binary number representations is \_\_\_\_\_\_\_\_\_\_\_\_\_.

**Problem 5 Sign Number Representation and Arithmetic**

1. Express the following decimal numbers , , and as an 8-bit binary number sign-magnitude form, 1’s complement form, and 2’s complement form.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Sign Magnitude |  |  |  |  |
| 1’s Complement |  |  |  |  |
| 2’s Complement |  |  |  |  |

1. Complete the following two operations.

|  |  |
| --- | --- |
| Compute the result for using 8-bit 2's complement operation. Verify if the result is correct. If not, explain why. | Compute the result for using 8-bit 1's complement operation. Verify if the result is correct. If not, explain why. |

**Problem 6: Binary Arithmetic**

1. Perform the following additions of two **unsigned** binary numbers. Indicate whether the sum overflows an 8-bit result.

1101 11012 + 0110 01102

1. Perform the addition of **hexadecimal** numbers: D9FC + 4B7E
2. Perform the following additions of two signed **one’s complement** binary numbers.

1101 10012 + 0110 01002

1. The numbers shown below are in 2’s complement binary representation. Determine whether the indicated arithmetic operations produce an overflow.
2. 0011 01102 + 1110 00112
3. 1001 10002 – 0010 00102
4. 0011 01102 – 1110 00112

**Problem 7 Boolean Identity**

Using Boolean identities to simplify the following Boolean function, show that (List the identity used at each step)

**Problem 8 Boolean Functions**

Write the Boolean expression in canonical sum-of-product form and canonical product-of-sum form for the following truth table.

|  |  |  |  |
| --- | --- | --- | --- |
| **x** | **y** | **z** | **F** |
| **0** | **0** | **0** | **1** |
| **0** | **0** | **1** | **0** |
| **0** | **1** | **0** | **1** |
| **0** | **1** | **1** | **0** |
| **1** | **0** | **0** | **1** |
| **1** | **0** | **1** | **1** |
| **1** | **1** | **0** | **0** |
| **1** | **1** | **1** | **0** |

