COMP 222 Assessment/Review Quiz Solutions

1 Convert the following numbers to decimal (base 10)

01101 (binary) _____

277 (octal) _____

3F (hexadecimal) _____

In binary, each position represents a power of 2

01101 = (0 X 2^4) + (1 X 2^3) +(1 X 2^2) + (0 X 2^1) + (1 X 2^0) = 0 + 8 + 4 + 0 + 1 = 13

In octal, each position represents a power of 8

277 = (2 X 8^2) + (7 X 8^1) + (7 X 8^0) = 128 + 56 + 7 = 191

In hex, each position represents a power of 16

Plus, additional symbols A-F are needed to represent values 10-15 as a single digit:

3F = (3 X 16^1) + (F X 16^0) = (3 X 16) + (15 X 1) = 48 + 15 = 63

5/37 = 13%

2 Integer arithmetic is usually implemented on computers using the ___2__'s complement format.

21/37 = 57%

3 Floating point arithmetic is usually implemented on computers using the IEEE __754____ format.

1/37 = 3%

4 Name some of the steps in the von Neuman instruction execution cycle: _____
_____fetch_decode_execute_store_____
_____

11/37 = 30%

5 What are the names of some common addressing modes for a typical assembly language? _____
_immediate_direct_indirect_indexed_____
_____

6/37 = 16%

6 The program that converts high level language to assembly language is__a compiler_____

18/37 = 49%

7 The program that converts assembly language to machine language is___an assembler_____

4/37 = 11%

8 Fill in the truth table for the AND function:

A	B	A AND B
True	True	True
True	False	False
False	True	False
False	False	False

33/37 = 89%

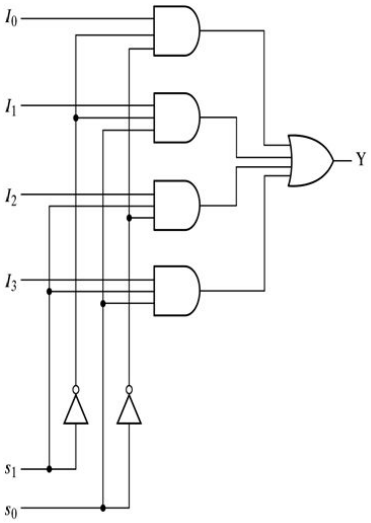
9 Fill in the truth table for the OR function:

A	B	A OR B
True	True	True
True	False	True
False	True	True
False	False	False

36/37 = 97%

10 The following circuit is a (circle one):

- Encoder
- Decoder
- Counter
- **Multiplexer**
- Demultiplexer



The function of the multiplexer (MUX) circuit is to take multiple inputs from the upper left, labeled I_0 through I_3, and choose only 1 to be passed through to the output Y to the right. Which input to pass through is controlled by the 2 select bits s_0 and s_1 at the bottom left. Since there are 4 inputs to select, there must be 2 select bits. More generally, n select bits are needed to control 2^n inputs. Note that the circuit makes available both the select bits and their negated versions.

The AND gates in the middle of the figure act as enable gates. They each have 3 inputs. The top input is the actual input, and the remaining 2 inputs control or enable/disable the first input by passing it through or blocking it by setting the output of the gate to 0. The control inputs are organized so that at most only one of the AND gates can be enabled at the same time. The output of the gate that is enabled will equal its input value. One gate will be enabled, and all other gates will be blocked at 0.

For the final step, the outputs of all the AND gates are ORed together. Since at most one of the AND gate outputs can be enabled (and the others blocked at 0), the output of the OR gate will a copy of the enabled AND gate output, which in turn is a copy of the input from the left that was selected by the select bits.

As a concrete example of how the control bits enable one AND gate and block all others, consider the case where we wish to enable input I_2. The index of the input to enable is 2, so the select bits are 10, with $s_1 = 1$ and $s_0 = 0$. Looking at the control inputs to the AND gate for I_2, they are s_1 and the negation of s_0. This is exactly the combination of select bit values that enables the AND gate for I_2. Similar analysis will show that all other AND gates are blocked.

23/37 = 62%