Philip Lenox, Devlin Junker
October 14, 2013
CS 472 HW 2

(2.5) Calculations are to be performed to a precision of 0.001. How many bits does this require? In order to represent numbers to this precision, 5 digits of decimal are needed. $\log_2\left(10^5\right) = 16.7$ this must be rounded up to 17 bits in order to represent numbers at that precision.

(2.13) A $00110111_2 + 01011011_2 = 10010010_2$

B $00111111_2 + 01001001_2 = 10001000_2$

C 00120121_{16} $+0A015031_{16}$ $= 0A135152_{16}$

D $00ABCD1F_{16}+$ $0F00800F_{16}$ $= 0FAC5D2E_{16}$

(2.14) What is arithmetic overflow? When does it occur and how can it be detected.

Arithmetic overflow occurs when number of the same sign are added together, and the resulting number is of the opposite sign. It occurs only when the same signed number are added togethed, either both negative or both positive. It can be detected by comparing the sign bit of the two operands to the sign bit of the result. If the result differs from both operands, then overflow occured.

(2.16) Convert 1234.125 into 32 bit IEEE floating point format.

sign	exponent	fraction
0	10001001	00111011111011111001110

(2.17) What is the decimal equivalent of the 32bit IEEE floating point value CC4C0000?

sign	exponent	fraction
1	10011000	10011000000000000000000
1	25	1.59375

-53477376

(2.22) What is the difference between a *truncation* error and a *rounding* error?

A *truncation* error is an error caused by lopping off nonsignificant bits, *truncation* errors always ends up with a value below the actual value. A *rounding* error is an error that is caused by the computer's rounding policy. Depending on the rounding policy, the value after the error can either be above the actual value or below the actual value.

(2.40) Draw a truth table for the given circuit (P2.40). Explain what it does

A	B	P	Q	R	C
0	0	1	1	1	0
1	0	1	0	1	1
1	1	0	1	1	0
0	1	1	1	0	1

Output's a 1 when the two inputs (A, B) are different. XOR gate.

(2.45) It is possible to have n-input AND, OR, NAND, and NOR gates, where $n > 2$. Can you have an n-input XOR gate for n > 2 Explain your answer with a truth table.

Whether or a multi-input (n > 2) XOR depends on the definition of XOR. With an odd number inputs, the output of when all inputs are true will be 1, with an even number of inputs, the output will be 0 (the typical definition of an XOR gate). This is because the final gate in the XOR gate structure will and the result of the first two inputs with the third. So it is not possible to build a standard XOR gate with n % 2 == 1 inputs.

A	B	C	XOR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1