Toby Chappell

CPSC 250

October 19, 2019

Homework 3

Digital Arithmetic

1. Find the following differences using twos complement arithmetic: (4 pts)
2. 111000 – 110011

111000 + 001101 = ~~1~~000101 = 000101

1. 11001100 – 101110

11001100 – 00101110 = 11001100 + 11010010 = ~~1~~10011110 = 10011110

1. 111100001111 - 110011110011

111100001111 + 001100001101 = ~~1~~001000011100 = 001000011100

1. 11000011 – 11101000

11000011 + 00011000 = 11011011

1. Assume numbers are represented in 8-bit twos complement representation. Show the calculation of the following: (4 pts)
2. 6 + 13

00000110 + 00001101 = 00010011

1. -6 + 13

11111010 + 00001101 = 00000111

1. 6 – 13

00000110 + 11110011 = 11111001

1. -6 – 13

11111010 + 11110011 = 11101101

1. Given x = 0101 and y = 1010 in twos complement notation (i.e., x = 5, y = -6), compute the product *p = x \* y* (2 pts)

0101\*1010 = 0000 0101\*1111 1010 = 0000 0000 + 0 0000 1010 + 00 0000 0000 + 000 0010 1000 + 0000 0101 0000 + 0 0000 1010 0000 + 00 0001 0100 0000 + 000 0010 1000 0000 = ~~0100~~ 1110 0010 = 1110 0010

1. Divide 145 by 13 in binary twos complement. (3 pts)

0000 1011

1. 0001/1101 =

010 100

01 110

0 0010

1011 Remainder : 10

1. Assume that the exponent e is constrained to lie in the range 0 … e … X, with a bias of q, that the base is b, and that the significand is p digits in length. (6 pts)
   1. What are the largest and smallest positive values that can be written?

Largest: p\*(b^(X-q))

Smallest: p\*(b^(0-q))

* 1. What are the largest and smallest positive values that can be written as normalized floating-point numbers?

Largest: Sign = 0 | Exponent = X + q | Fraction = p – 1.0

Smallest: Sign = 0 | Exponent = 0 + q | Fraction = p – 1.0

1. Express the following numbers in IEEE 32-bit floating-point format: (5 pts)
   1. -5

Sign: 1

Integral: 5 = 101

Normalize: 1.01 \* 2^2

Mantissa: 01

Exponent: (2^(8-1) – 1) + 2 = 127 + 2 = 129 = 10000001

1 10000001 01000000000000000000000

* 1. -6

Sign: 1

Integral: 6 = 110

Normalize: 1.1 \* 2^2

Mantissa: 1

Exponent: 127 + 2 = 129 = 10000001

1 10000001 10000000000000000000000

* 1. -1.5

Sign: 1

Integral: 1 = 1

0.5\*2 = 1: 1

1.1

Normalize: 1.1 \* 2^0

Mantissa: 1

Exponent: 127 + 0 = 127 = 01111111

1 01111111 10000000000000000000000

* 1. 384

Sign: 0

Integral: 384 = 110000000

Normalize: 1.1 \* 2^8

Mantissa: 1

Exponent: 127 + 8 = 135 = 10000111

0 10000111 10000000000000000000000

* 1. 1/16

1/16 = 0.0625

Sign: 0

Integral: 0 = 0

0.0625 \* 2 = 0.125: 0

0.125 \* 2 = 0.25: 0

0.25 \* 2 = 0.5: 0

0.5 \* 2 = 1: 1

0.0001

Normalize: 1.0 \* 2^-4

Mantissa: 0

Exponent: 127 – 4 = 123 = 01111011

0 01111011 00000000000000000000000

1. The following numbers use the IEEE 32-bit floating-point format. What is the equivalent decimal value? (6 pts)

a. 1 10000011 11000000000000000000000

Sign: -

Exponent: 10000011 = 131

Mantissa: 11

Bias: 131 – 127 = 4

De-normalize: 1.11 \* 2^4 = 11100

Convert: 2^4 + 2^3 + 2^2

-28

b. 0 01111110 10100000000000000000000

Sign: +

Exponent: 01111110 = 126

Mantissa: 101

Bias: 126 – 127 = -1

De-normalize: 1.101 \* 2^-1 = 0.1101

Convert: 2^-1 + 2^-2 + 2^-4

0.8125

c. 0 10000000 00000000000000000000000

Sign: +

Exponent: 10000000 = 128

Mantissa: 0

Bias: 128 – 127 = 1

De-normalize: 1.0 \* 2^1 = 10

Convert: 2^1

2