**Note:** Please post your homework to ICS232 D2L on or before the due date.

If you do not post your homework on or before the due data, please post your late homework when complete. (Late Homework: -15% Penalty).

1. (3 pts) Consider an unsigned fixed point decimal (Base10) representation with 8 digits, 5 to the left of the decimal point and 3 to the right.

a.      What is the range of the expressible numbers?

Answer: [00000.000, 99999.999]

b.      What is the precision?

Answer: 0.001 the precision is the distance between successive numbers

c.       What is the error?

Answer: the error is ½ the precision. The error is 0.01/2 = 0.0005

1. (3 pts) Convert this unsigned base 2 number, **1001 10112**, to each base given below

(Note: the space in the binary string is purely for visual convenience)

Show your work.

* 1. Using the Polynomial method convert the number above from base 2 to base 10 (decimal)

Answer: 128 + 16 + 8 + 2 + 1 = (155)10

* 1. Using the grouping method convert number above from base 2 to base 16… (hex)

Answer: (1001) (1011) = (9b)16

* 1. Using the grouping method convert number above from base 2 to base 8… (octal)

Answer: (010) (011) (011) = (233)8

1. (3 pts) Convert this unsigned base 2 number, **11011.100112**, to each base given below

(Note: The placement of the decimal point. Correct padding must be used)

Show your work.

* 1. Using the grouping method convert number above from base 2 to base 16… (hex)

Answer: (0001) (1011).(1001)(1000) = 1b.9816

* 1. Using the grouping method convert number above from base 2 to base 8… (octal)

Answer: (011) (011) . (100) (110) = 33.468

* 1. Using the grouping method convert number above from base 2 to base 4… ( )

Answer: (01) (10) (11) . (10) (01) (10) = 123.2124

1. (3 pts) Convert 597.2210 (decimal number) to **unsigned binary** using the remainder and multiplication methods:. Stop at the 6th digit to the right of the decimal place. (Show your work)

Answer: 10 0101 0101.001110

|  |  |
| --- | --- |
| 597/2 = 298 r 1 | .22 x 2 = 0.44 |
| 298/2 = 149 r 0 | .44 x 2 = 0.88 |
| 149/2 = 74 r 1 | .88 x 2 = 1.76 |
| 74/2 = 37 r 0 | .76 x 2 = 1.52 |
| 37/2 = 18 r 1 | .52 x 2 = 1.04 |
| 18/2 = 9 r 0 | .04 x 2 = 0.08 |
| 9/2 = 4 r 1 |  |
| 4/2 = 2 r 0 |  |
| 2/2 = 1 r 0 |  |
| 1/2 = 0 r 1 |  |

1. (3 pt) Using the overflow concept and its identifying methods introduced in the class, solve the problem: suppose that we have a computing device that has only 8 bits. Using this device, add the following two unsigned 8 bit binary numbers and give the final result. If there is an overflow, point that out (you must clearly indicate whether your final result has overflow).

1 1 0 1 1 1 0 0 carry bits

0 1 1 0 1 1 1 0

+ 1 1 1 0 1 0 1 1

============

Answer: (1) 0 1 0 1 1 0 0 1 , the leftmost 1( ninth bit) is overflow

1. (3 pts) Convert -12310 to an 8-bit binary number using the representations given below.

Show all 8 bits.

* 1. signed magnitude number

Answer: 1111 10112

* 1. one's complement number

Answer: 1000 01002  (# negative so compliment the number 123)

* 1. two's complement number

Answer: 1000 01012

* 1. excess 128 number (bias the two’s complement number)

Answer: 0000 01012

1. (6 pts) Consider the bit pattern 1101 10102. **Provide the equivalent value in Base10 (decimal)** for this bit pattern based on the following assumptions: (if the bit pattern represents a negative number under the assumed context, then give its negative value).

Assume the original number is expressed using:

* 1. signed magnitude representation

Answer:

-9010 ( sign bit ‘1’… negative number, weighted method 1\*26 + 0\*25 + 1\*24 + 1\*23 + 0\*22 + 1\*21 + 0\*20 = -9010)

* 1. unsigned representation

Answer:

21810 (no sign bit, weighted method 1\*27 + 1\*26 + 0\*25 + 1\*24 + 1\*23 + 0\*22 + 1\*21  + 0\*20 = 21810)

* 1. one's complement representation

Answer:

-3710

The number is negative so: Complement to get the magnitude of the number while keeping the sign 1 010 0001 ( sign bit ‘1’… negative number, weighted method 0\*26 + 1\*25 + 0\*24 + 0\*23 + 1\*22 + 0\*21 + 1\*20 = -3710)

* 1. two's complement representation

Answer:

-3810

The number is negative so: Subtract one from the two’s complement number to get the one’s complement number ; then complement the magnitude of the number while keeping the sign.

1101 1010 -> 1101 1001 -> compliment 0 010 0110 -> - ( original sign bit was ‘1’… negative number, weighted method 0\*26 + 1\*25 + 0\*24 + 0\*23 + 1\*22 + 1\*21 + 0\*20 = -3810)

* 1. excess 128 representation (biased two’s complement number)

Answer:

9010

128 – 38 = 90

Remove the 128 bias to get the two’s complement number which is positive (0101 1010). Therefore no subtraction of 1 nor complement of the number.

1. (3 pts) This problem tests your knowledge about coding schemes. What is the binary bit pattern for the letter 'h' using?

**The answers should give the whole bit string (including leading 0s).**

* 1. ASCII encoding (7-bits)

110 1000 (68)16

* 1. EBCDIC encoding (8-bits)

1000 1000 (88)16

* 1. UNICODE encoding (16 bits)

0000 0000 0110 1000 (0068)16

1. (3 pts) Show how each of the following floating point values would be stored using IEEE-754 single precision (be sure to indicate the sign bit, the exponent, and the significand fields):
   1. 12.5

Ans. a) 12.5 = 1.1001 × 23 0 10000010 1001000...0

Positive sign bit

Exponent 3+127 = 130 = 10000010

Hidden Bit 1.

Significant .1001

* 1. −1.5

Ans.  b)  −1.5=−1.1×20 1 01111111 1000000...0

Negative sign bit

Exponent 0+127=127=01111111

Hidden Bit 1.

Significant .1

* 1. 0.75

Ans.  c)  0.75 = 1.1 × 2−1 0 01111110 1000000...0

Positive sign bit

Exponent  −1 + 127 = 126 = 01111110

Hidden Bit 1.

Significant .1

* 1. 26.625

Ans.  26.625=1.1010101×24 0 10000011 1010101...0

Positive sign bit

Exponent 4+127=131=10000011

Hidden Bit 1.

Significant . 1010101

1. (6 pts) Show how each of the following floating point values would be stored using IEEE-754 double precision (be sure to indicate the sign bit, the exponent, and the significand fields):
   1. 13.5

Ans: 0 10000000010 1011000...0

a) 13.5 = 1.1011 × 23

Positive sign bit

Exponent 3+1023 = 1026 = 10000000010

Hidden Bit 1.

Significant .1011….

* 1. −102.25

Ans. 0 10000000101 100110010….0

b)  −102.5=−1100110.01= -1.10011001×26

Negative sign bit

Exponent 6 + 1023 = 1029 = 100 0000 0101

Hidden Bit 1.

Significant .100110010….

* 1. 0.0078125

Ans. 0 01111111000 0...0

c)  0.0078125= 0.1×25

Positive sign bit

Exponent -7 + 1023 = 1016 = 011 1111 1000

Hidden Bit 1.

Significant .0

1. (6 pts) Perform the following binary multiplications using Booth’s algorithm, assuming signed two’s complement integers:
   1. 1011 × 0101

Ans. a) 11100111

* 1. 1011 × 0101

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | 1 | 0 | 1 | 1 |
|  |  |  |  |  |  |  |  |  | x | 0 | 1 | 0 | 1 |
|  |  |  |  |  | 0 | 0 | 0 | 0 | 0 | 0+1 | 1 | 0 | 1 |
| + |  |  |  |  | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |  |
| Sum |  |  |  | +1 | 1+1 | 1+1 | 1 +1 | 1 + 1 | 1 | 1 | 0 | 1 | 1 |
| + |  |  |  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |  |
| Sum |  |  |  | 1 | 0 | 0 | 0 | 0+1 | 0+1 | 1 | 1 | 1 | 1 |
| + |  |  | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |  |  |  |
| Sum |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 |

+00000101 (10=subtract 1101= add 00000101)

+11111011 (01= add 11111011 to product, extend sign)

+00000101 (10=subtract 1101= add 00000101)

+11111011 (01= add 11111011 to product, extend sign)

100111100111 (We only consider the 8 rightmost bits, thus, we have -5 \* 5 = -25

* 1. 0011 × 1011

Ans. b) 11110001

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | 0 | 0 | 1 | 1 |
|  |  |  |  |  |  |  |  |  | x | 1 | 0 | 1 | 1 |
|  |  |  |  |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| Shift |  |  |  | +1 | 1 + 1 | 1 + 1 | 1 + 1 | 1 + 1 | 1 +1 | 1 | 0 | 1 |  |
| + |  |  |  | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  |  |
| Sum |  |  | +1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |  |
| + |  |  | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |  |  |  |
|  |  | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

~~11101110~~

+11111101 (10=subtract 0011= add 11111101)

+00000000 (11= simple shift)

+00000011 (01=add 0011)

+11111101 (10=subtract 0011= add 11111101)

1000 11110001 (We only consider the 8 rightmost bits, thus, we have +3 \* -5 = -15

* 1. 1011 × 1100

Ans. c) 00010100

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |  | 1 | 0 | 1 | 1 |
|  |  |  |  |  |  |  |  |  | X | 1 | 1 | 0 | 0 |
| Shift |  |  |  |  | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |  |
| Shift |  |  | +1 | 1+1 | 1+1 | 1+1 | 1+1 | 1 | 0+1 | 1 | 1 |  |  |
| + |  |  | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |  |  |  |
| Sum |  |  | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |

+00000000 (00=simple shift)

+00000000 (00=simple shift)

+00000101 (10=subtract 1011= add 00000101)

+00000000 (11= simple shift)

000000010100 (We only consider the 8 rightmost bits, thus, we have -5 \* -4 = +20.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. (3 pts) Using arithmetic shifting, perform the following on the two’s complimant numbers:
   1. double the value 000101012

Ans. a) 00101010

* 1. quadruple the value 011101112

Ans. b) error (sign bit changes)

* 1. divide the value 110010102 in half

Ans. c) 01100101

1. (3 pt) Find the quotients and remainders for the following division problems modulo 2
   1. 10011112 ÷ 11012

Ans. a) 1111 Remainder 100

1111

1101 ) 1001111

1101

01001

1101

01001

1101

01001

1101

0100

Quotient 1111, Remainder 100

* 1. 10111102 ÷ 11002

b) 1101 Remainder 10

1101

1100 ) 1011110

1100

01111

1100

001110  
 1100

0010  
  
quotient 1101, remainder 10

* 1. 10011011102 ÷ 110012

c) 111001 Remainder1111

1110001

11001 ) 1001101110

11001

010100

11001

011011

11001

00010110

11001

01111  
  
quotient 1110001, remainder 1111

* 1. 1111010102 ÷ 100112

d) 11100 Remainder 1110

11100  
10011 ) 111101010

10011

011011

10011

010000

10011

0001110

quotient 11100, remainder 1110

1. (6 pts) Using the CRC polynomial 1101, compute the CRC code word for the information word, 01001101. Check the division performed at the receiver.

Ans. In information word is: 01001101. The CRC is 100

Check: The code word is 01001101100. Dividing this by 1101 modulo2 should yield a zero remainder.

1101 is a 3rd order polynomial so: Append three 0s to the end of the information word and divide:

1101 01001101000

1101

1001

1101

1000

1101

1011

1101

1100

1101

100

100 → remainder The information word (with appended zeros) + remainder = codeword

so we have: 01001101000 100 = 01001101100 To check the division:

01001101100

1101

1001

1101

1000

1101

1011

1101

1101

1101

000

1. (6 pt) Using the CRC polynomial 1101, compute the CRC code word for the information word, 01011101. Check the division performed at the receiver.

Ans. The code word is: 01011101101 Dividing this by 1101 modulo2 will yield a zero remainder. Note: Ignore the leading zero in the information word when setting up the dividend, as illustrated in Problem 76.

Ans. In information word is: 01011101. The CRC is 101… Total code word is: 01011101 101

1101 is a 3rd order polynomial so: Append three 0s to the end of the information word and divide:

1101 01011101000

1101

1101

1101

00001000

1101

101

To check the code word answer:

1101 01011101101

1101

1101

1101

00001101

1101

000