Homework #1 Problems

Quiz in Recitations on Monday, September 21 & Wednesday, September 23

Problem 1: Convert the following decimal numbers to binary, octal, and hexadecimal.

(a) 1984 (b) 4000 (c) 8192

Answer:

(a) **To binary, by division by 2**:

1984/2 Remainder 0
992/2 Remainder 0
496/2 Remainder 0
284/2 Remainder 0
124/2 Remainder 0
62/2 Remainder 0
31/2 Remainder 1
15/2 Remainder 1
7/2 Remainder 1
3/2 Remainder 1
1/2 Remainder 1

Reading from bottom up, left to right → 11111000000_2

To octal: First, convert to binary → 11111000000. Then group by every 3 bits starting from the right (octal is base 8, 8 symbols are represented by 3 binary bits [range 0-7])
11 111 000 000 → 3700_8

To Hexadecimal: First, convert to binary → 11111000000. Then group by every 4 bits starting from the right (hex is base 16, 16 symbols are represented by 4 binary bits [range 0-9,A,B,C,D,E,F])
111 1100 0000 → $7C0_{16}$

(b) **To binary, by division of 2**.

4000/2 Remainder 0
2000/2 Remainder 0
1000/2 Remainder 0
500/2 Remainder 0
250/2 Remainder 0
125/2 Remainder 1
62/2 Remainder 0
31/2 Remainder 1
15/2 Remainder 1
7/2 Remainder 1
3/2 Remainder 1
1/2 Remainder 1

Reading from bottom up, left to right → 111110100000_2

To octal: First convert to binary → 111110100000_2 . Then group by every 3 bits starting from the right (octal is base 8, 8 symbols are represented by 3 binary bits [range 0-7])
111 110 100 000 → 7640_8

To Hexadecimal: First convert to binary → 111110100000_2. Then group by every 4 bits starting from the right (hex is base 16, 16 symbols are represented by 4 binary bits [range 0-9,A,B,C,D,E,F])
$1111\ 1010\ 0000_2$ → $FA0_{16}$

(c) **To binary, by division by 2.**

8192/2 Remainder 0
4096/2 Remainder 0
2048/2 Remainder 0
1024/2 Remainder 0
512/2 Remainder 0
256/2 Remainder 0
128/2 Remainder 0
64/2 Remainder 0
32/2 Remainder 0
16/2 Remainder 0
8/2 Remainder 0
4/2 Remainder 0
2/2 Remainder 0
1/2 Remainder 1

Reading from bottom up, left to right → 10000000000000_2

To octal: $10\ 000\ 000\ 000\ 000_2$ → 20000_8

To Hex: $10\ 0000\ 0000\ 0000_2$ → 2000_{16}

<u>Problem 2:</u> Which of the following are valid

(a) hexadecimal numbers?

(b) base 13 numbers?

BED, CAB, DEAD, DECADE, ACCEDED, BAG, DAD

Answer: Hexadecimal numbers have symbols from 0-9, A,B,C,D,E,F.
Base 13 numbers have symbols 0-9, A,B,C.

BED – Hex;
CAB – Hex or Base 13;
DEAD – Hex;
DECADE – Hex;
ACCEDED – Hex;
BAG – neither
DAD – Hex.

Problem 3: Express the decimal number 100 in all radices from 2 to 9 .

Answer: Use the method of division for each radix.

100/2 Remainder 0
50/2 Remainder 0
25/2 Remainder 1
12/2 Remainder 0
6/2 Remainder 0
3/2 Remainder 1
1/2 Remainder 1
Reading from bottom up, left to right → 1100100_2

100/3 Remainder 1
33/3 Remainder 0
11/3 Remainder 2
3/3 Remainder 0
1/3 Remainder 1
Reading from bottom up, left to right → 10201_3

100/4 Remainder 0
25/4 Remainder 1
6/4 Remainder 2
1/4 Remainder 1
Reading from bottom up, left to right → 1210_4

100/5 Remainder 0
20/5 Remainder 0
4/5 Remainder 4.
Reading from bottom up, left to right → 400_5

100/6 Remainder 4
16/6 Remainder 4
2/6 Remainder 2
Reading from bottom up, left to right → 244_6

100/7 Remainder 2
14/7 Remainder 0
2/7 Remainder 2
Reading from bottom up, left to right → 202_7

100/8 Remainder 4
12/8 Remainder 4
1/8 Remainder 1
Reading from bottom up, left to right → 144_8

100/9 Remainder 1
11/9 Remainder 2
1/9 Remainder 1
Reading from bottom up, left to right → 121_9

Problem 4: Convert the following numbers. Show all steps.

(a) A3F in hex to a number in base 4.

(b) -27 in decimal to 8-bit binary number in 1's complement form.

(c) 22.2 in base 3 to a decimal number. (You must have at least two digits after decimal point.)

(d) E4.A9 in hex to a number in base 8.

Answer:

(a) Convert A3F to binary → 1010 0011 1111,
Group by 2's for base 4 → 0 10 00 11 11 11 → 220333_4

(b) -27 in decimal to 8-bit binary number in 1's complement form.
Sign bit will be 1 since sign should be negative in final answer; 7 bits to represent the magnitude.
27 in 8-bit binary → 00011011_2. Flip all the bits to get 1's complement -27.
So, in 8-bit 1's complement → 11100100_2

(c) 22.2 in base 3 to a decimal number.
22.2_3 to decimal → $2*3^1 + 2*3^0 + 2*3^{-1} = 6 + 2 + .67 = 8.67$

(d) Convert $E4.A9_{16}$ to binary → $1110\ 0100.\ 1010\ 1001_2$.
Now group by 3's, starting from decimal point to convert to base 8:
$11\ 100\ 100\ .\ 101\ 010\ 01_2$ → 344.522_8

Problem 5: Let numbers be six (6) bits wide including the sign. Negative numbers are stored in 2's complement form. What is the range for negative numbers that can be stored?

Answer: The range of 2's complement is $-(2^{(n-1)})$ to $(2^{(n-1)} - 1)$, where n is the number of bits used to represent the number.
Therefore, a 6 bit number in 2's complement can represent -32 to 31.
So the range of negative numbers that can be stored is -1 to -32.

Problem 6: Assume that negative numbers are stored in 2's complement form. Perform the three operations, (A - B), (B - A), and (A + B) for the given values of A and B. Obtain the results, and state if we have an overflow/underflow.

(a) Let A = 100011 and B = 001100.

(b) Let A = 110001 and B = 100111.

Answer:

(a) A is a negative number. B is a positive number. For (A-B), convert B to a negative number and add → A+ (-B); to convert B, flip the bits and add 1.
For (B-A), convert A to a positive number and add.
For (A+B), just add.

A = -29 , B = 12 .
–A = 011101_2 , –B = 100011_2 .
A – B = -41 (underflow).
B – A = 41 (overflow).
A + B = 101111_2 = -17

(b) A= -15 , B = -25 .
–A = 001111_2 , –B = 011001_2 .
A – B = 001010 .
B – A = 110110 .
A + B = -40 (underflow).

<u>Problem 7:</u> Consider two decimal numbers A and B, where A = 42 and B = 12 (show steps).

(a) First convert A and B to 8-bit wide unsigned binary numbers.

(b) Calculate A + B in sign-magnitude binary.

(c) What is the 1's complement of B?

(d) What is the 2's complement of B?

(e) Calculate A - B by adding A and the 2's complement of B.

(f) Calculate B - A in binary. Negative numbers must be in 2's complement form.

Answer:

(a) A= 42; to unsigned binary → 00101010_2.
B = 12; to unsigned binary → 00001100_2

(b) A+B in sign-magnitude: 00101010 + 00001100 = 00110110_2 .

(c) 1's complement for B, means to write the number as –B in 1's complement form = 11110011_2 .

(d) 2's complement for B, means to write the number as –B in 2's complement form = 11110100_2 .

(e) A-B = A + (-B) = 00101010 + 11110100 = 1 00011110 ;
drop the extra carry-out bit → 00011110 .

(f) B-A = 00001100 – 00101010 = 00001100 + 11010110 = 11100010 .

Problem 8: Consider the following addition problems for 3-bit binary numbers. Negative numbers are in 2's complement form. For each sum, state:

(i) Whether the sign bit of the result is 1.

(ii)Whether an overflow/underflow has occurred.

000 + 000 ; 010 + 010 ; 100 + 001 ; 101 + 110 ; 111 + 100 .

Answer:
(1) sign bit is 0, no overflow/underflow.

(2) sign bit is 1, overflow occured.

(3) sign bit is 1, no overflow/underflow.

(4) sign bit is 0, underflow occurred.

(5) sign bit is 0, underflow occurred.

Problem 9: Given the bit pattern: 1010 1100 1011 0101 0011 0000 0011 1000, what value does it represent, assuming that it is,

(a) a 2's complement integer?

(b) an unsigned integer?

(c) a single precision floating-point number?

Answer:

(a) The number given is negative. To find its magnitude, flip the bits and 1 in order to convert it to a positive number.
Flip bits: 0101 0011 0100 1010 1100 1111 1100 0111
Add 1: 0101 0011 0100 1010 1100 1111 1100 1000
Value: $2^{30}+ 2^{28}+ 2^{25}+2^{24} + 2^{22}+ 2^{19}+2^{17}+2^{15}+2^{14}+2^{11}+2^{10}+2^{9}+2^{8}+2^{7}+2^{6}+2^{3}$
$= 1,397,411,784_{10}$ → add the sign: - 1,397,411,784 .

(b) an unsigned integer = $2^{31}+2^{29}+2^{27}+2^{26}+2^{23}+2^{21}+2^{20}+2^{18}+2^{16}+2^{13}+2^{12}+2^{5}+2^{4}+2^{3}$
$= 2,897,555,512_{10}$.

(c) a single precision floating-point number:
Sign = 1
Exponent = 010 11001 = 64+16+8+1 = 89 (note that this is a value in Excess-127);
converting the exponent value back from Excess-127: 89 - 127 = -38 .
Fraction = 011 0101 0011 0000 0011 1000 .
Putting the pieces together → $-1.01101010011000000111000_2 * 2^{-38}$

Problem 10: The following binary floating-point numbers consist of a sign bit; an excess-64, radix-2 exponent; and a 16-bit fraction. Assume that the fractional parts have an implicit leading "0." that is not included in the number (rather than the usual "normalized" implicit "1."). Now normalize the numbers and write them back in the same excess-64 with 16-bit fraction form.

(a) 010000000001010100000001

(b) 001111110000001111111111

(c) 010000111000000000000000

Answer: Note that the exponent field is 7 bits wide since the numbers are 24 bits with 1 bit for the sign and 16 bits for the fraction.

(a) 0 1000000 0001010100000001
Sign bit is 0 ; Exponent = 64 - 64 = 0 ; Fraction = 0001010100000001 → $0.0001010100000001_2 * 2^0$.
Normalizing yields $1.010100000001_2 * 2^{-4}$ → 0 0111100 0101000000010000

(b) 0 0111111 0000001111111111
Sign bit is 0 ; Exponent = 63 - 64 = -1 ; Fraction = 0000001111111111 → $0.0000001111111111_2 * 2^{-1}$.
Normalizing yields $1.111111111_2 * 2^{-8}$ → 0 0111000 1111111110000000

(c) 0 1000011 1000000000000000
Sign bit is 0 ; Exponent = 67 - 64 = 3 ; Fraction = 1000000000000000 → $0.1000000000000000_2 * 2^3$.
Normalizing yields $1.0_2 * 2^2$ → 0 1000010 0000000000000000

Problem 11: Convert (- 110011.1011) in binary to single precision IEEE format

(a) State the normalized number.

(b) Write the number in IEEE format (sign, exponent and significand).

(c) IEEE format number in HEX.

Answer:
(a) $-1.100111011_2 * 2^5$

(b) Sign bit = 1 ; Exponent = 5+127 = 132 = 10000100 ;
Fraction/Significand = 10011101100000000000000 .

(c) HEX: 1100 0010 0100 1110 1100 0000 0000 0000 → $C24EC000_{16}$

Problem 12: Show the binary representation for the following decimal floating point values in IEEE single and double precision format.

(a) 7.4 (b) -13/64 (c) 6.125

Answer:

(a) In binary: $7 = 111_2$; $0.4 = 0.01100110011001100110\ldots_2$ → $111.011001100110\ldots_2$

Normalized= $1.1101100110011001100110\ldots * 2^2$

In single precision: 0 10000001 11011001100110011001100

In double precision: 0 10000000001 1101100110011001100110011001100110011001100110011000

(b) -13/64 = -0.203125 → In binary = 0.001101_2 → Normalized = $1.101 * 2^{-3}$

In single precision: 0 01111100 10100000000000000000000

In double precision: 0 01111111100 1010000000000000000000000000000000000000000000000000

(c) In binary: $6 = 110_2$; $0.125 = 0.001_2$ → 110.001_2 → Normalized = $1.10001 * 2^2$

In single precision: 0 10000001 10001000000000000000000

In double precision: 0 10000000001 1000100000000000000000000000000000000000000000000000

Problem 13: Convert the hexadecimal numbers to IEEE single precision format, and then to decimal.

(a)B54B7F10 (b) 43D00000 (c) C20C0000 (d) 42055555

Answer:

(a) To binary: 1011 0101 0100 1011 0111 1111 0001 0000

Sign is - ; Exponent = 106 - 127 = -21; Fraction = 10010110111111100010000

→ $-1.10010110111111100010000_2 * 2^{-21} = -1.58981513977050781 25_{10} * 2^{-21}$

$= -2.8124577511334791779518127441406_{10} * 10^{-7}$

(b) To binary: 0100 0011 1101 0000 0000 0000 0000 0000

Sign = + ; Exponent = 135 - 127 = 8 ; Fraction = 10100000000000000000000

→ $1.101_2 * 2^8 = 1.625_{10} * 2^8 = 416.0_{10}$

(c) To binary: 1100 0010 0000 1100 0000 0000 0000 0000

Sign = - ; Exponent = 132 - 127 = 5 ; Fraction = 00011000000000000000000

→ $-1.00011_2 * 2^5 = -1.09375_{10} * 2^5 = -35.0_{10}$

(d) To binary: 0100 0010 0000 0101 0101 0101 0101 0101

Sign = + ; Exponent = 132 - 127 = 5 ; Fraction = 00001010101010101010101

→ $1.00001010101010101010101_2 * 2^5 = 100001.01010101010101010101_2 * 2^0$

$= 33.33333206176757812 5_{10}$

Problem 14: Perform addition (0.0111 + 0.111) for these two binary numbers.

(a) Normalize the binary numbers (normalization only) in single precision IEEE format.

(b) Align the exponents and perform addition.

(c) Normalize the result and convert it to IEEE format. Show all steps. State all bits that are important (biased exponent and fraction in binary)

Answer:
(a) 0.0111 → $1.11 * 2^{-2}$; 0.111 → $1.11 * 2^{-1}$

(b) Align the exponents and add: $0.111 * 2^{-1} + 1.11 * 2^{-1} = 10.101 * 2^{-1}$

(c) $10.101 * 2^{-1}$ → $1.0101 * 2^{0}$
Convert to IEEE single precision format: 0 01111111 01010000000000000000000

Problem 15: Explain how this subtraction (2.75 - 8.625) will be performed? Begin by explaining how these numbers are stored in IEEE single precision format. Show important steps.

Answer:
Convert each number to IEEE single precision:

2.75 in binary = 10.11 → $1.011 * 2^{1}$
In IEEE single precision → 0 10000000 01100000000000000000000

8.625 in binary = 1000.101 → $1.000101 * 2^{3}$
In IEEE single precision → 0 10000010 00010100000000000000000

To subtract, align the exponents (align the number with the smaller-exponent value to that of the larger exponent-value); *i.e.*, shift the 2.75 to the left by 2 positions → $0.01011 * 2^{3}$

Now subtract the smaller number from the larger. Because this involves flipping the order of the subtraction around (8.625 - 2.75), the result will be negative.

$$
\begin{array}{r}
1.000101 * 2^{3} \\
- \quad \underline{0.010110 * 2^{3}} \\
-0.101111 * 2^{3}
\end{array}
$$

Re-normalize the result → $-1.01111 * 2^{2}$
In IEEE single precision → 1 10000001 01111000000000000000000

Problem 16: Consider a machine with a 32-bit word (4 bytes in a word). Bytes are numbered 0, 1, 2, 3, ... , and words are numbered 0, 1, 2, ... , *etc*. Word-0 contains byte-0, byte-1, byte-2 and byte-3. Word-1 contains byte-4, byte-5, byte-6, and byte-7, *etc*.

(a) The byte numbered 406 would belong to what word?_______ (word-number)

(b) This memory is organized using big endian byte order, what would be the least significant byte (one that holds the Least Significant Bit) of Word-20? ______ (byte-number)

Answer:

(a) The byte numbered 406 would be in a word numbered (406)/4 = 101 (integer division).

(b) The number of the LSByte of word 20 is byte-83 (20*4 + 3).