CSE 3430 NAME: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

AU 25

# Homework 1

**Due: Monday, September 8th by 11:30 pm on Carmen (NOTE: If the homework is submitted by 11:59 pm on the due date, it will be counted as being on time, but if submitted at midnight or after, up till 11:59 pm the following day, it will be counted one day late, and the 25% deduction will apply. Please submit by 11:30 pm *at the very latest*, so you do not risk submitting at midnight or after). You can submit a docx or a pdf to Carmen. We do not accept email submissions for homework.**

**You should type your answers into this document or write them legibly by hand, but for problems which require calculations, be sure to show your work; no credit will be given for a final answer without work, even if the final answer is correct.**

**IMPORTANT: Show your work for problems which involve calculations! [For problems which have boxes, you can just fill in the boxes].**

1. Consider the 5-bit binary encoding 10011, and decode the value as an unsigned decimal integer (B2U).

ANSWER:

1. Consider the 5-bit binary encoding 10011, and decode the value as a signed decimal integer (B2T; that is, assume a 2’s complement encoding)

ANSWER:

1. Using the formula given in class, write the range of decimal values which can be encoded by a B2T (2’s complement) 6-bit signed binary number (first, write the range using powers of 2, and then, write the decimal values):

Range for 6-bit signed numbers (B2T) =

ANSWER: to (written using powers of 2)

ANSWER: to (written using decimal number values)

1. Fill in the table given to add the two unsigned (B2U) 4-bit values below (Op1 and Op2), which have been converted to binary. Be sure to show all carries (no matter whether they are 1 or 0) and all sum bits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Carry | (Last carry) |  |  |  |  |
| Op1 |  | 0 | 1 | 1 | 0 |
| Op2 |  | 0 | 1 | 1 | 1 |
| Sum |  |  |  |  |  |

* + - 1. Is there overflow for this result (put an X in one box)?

ANSWER: [ ] Yes [ ] No

* + - 1. How does the CPU hardware adder determine whether there is overflow or not?

ANSWER:

1. Add the two 4-bit signed B2T (2’s complement) values given (Op1 and Op2 below). Be sure to show all carries and all sum bits.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Carry | (Last carry) |  |  |  |  |
| Op1 |  | 1 | 0 | 1 | 1 |
| Op2 |  | 0 | 1 | 1 | 0 |
| Sum |  |  |  |  |  |

* + - 1. Is there overflow for this result (put an X in one box)?

ANSWER: [ ] Yes [ ] No

* + - 1. How does the CPU hardware adder determine whether there is overflow or not?

ANSWER:

1. Using 4-bit 1’s complement (B2O), show the addition of the two operands Op1 and Op2 below, using the methods described in class for adding 1’s complement numbers correctly (Remember that the hardware needs to determine what bit value to use as the first carry, and that in some cases, 2 addition operations are required).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Carry | (Last carry) |  |  |  |  |
| Op1 |  | 0 | 1 | 1 | 0 |
| Op2 |  | 1 | 0 | 1 | 1 |
| Sum |  |  |  |  |  |

* 1. How does the hardware (the addition unit in the CPU) determine if a second addition operation is required?

ANSWER:

* 1. Is a second addition operation required in this case (Put an X in one box)?

ANSWER: [ ] YES [ ] NO

* 1. If a second addition is required, what is added back to the first sum from above to get the final correct result after the second addition?

ANSWER:

If a second addition is required, use the table below to show how it is done:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Carry | (Last carry) |  |  |  |  |
| Result of 1st addition above |  |  |  |  |  |
| Carry from 1st addition |  |  |  |  |  |
| Final Sum |  |  |  |  |  |

* 1. If you interpret the result as a 1’s complement value (after the 2nd addition, if you determined one is required), is the final result the correct result?

ANSWER: [ ] Yes [ ] No

1. For the 32-bit floating point number given below, encoded according to the IEEE 754 standard, determine the value of the encoded number in decimal. If there is a fractional part of the value (a part less than 1), you should write it as x/y, where x and y are integers.

0 1000 0011 1010 0001 1001 0000 0000 000

Be sure to show your work!

ANSWER:

1. For the 32-bit floating point number given below, encoded according to the IEEE 754 standard, determine the value of the encoded number in decimal. If there is a fractional part of the value (a part less than 1), you should write it as x/y, where x and y are integers.

1 0111 1101 1010 1000 0000 0000 0000 000

Be sure to show your work!

ANSWER:

1. Using an 8 bit ASCII encoding for each character (see the ASCII encoding table on Carmen in the Homework folder), encode the following bit strings (please separate sequences of 4 bits with a space, to aid readability):
   * 1. No way! [Notice that a space is a separate character in ASCII.]

ANSWER (Please put it all on one line below):

* + 1. Brutus

ANSWER (Please put it all on one line below):

1. For the UTF-8 encoding given at the end of the question below, which encodes some unknown number of characters, answer the following questions:
   * 1. How many characters are encoded?

ANSWER:

* + 1. Now fill in the table below (add more rows if necessary), and for each character, say how many bytes are used to encode it:

|  |  |
| --- | --- |
| Character | Number of bytes  for character |
| 1st |  |
| 2nd |  |
| 3rd  (Add more rows  below if needed) |  |

**UTF-8 encoding:**

1101 1111 1011 0011 0010 1001 1110 0001 1011 0101 1010 1111