# CS3650 Homework #3 (20 points)

Problem 1: Integer multiplication unit. (4 points)

Multiplication unit

1. The above shows a 32-bit integer multiplication unit. We reduce the size accordingly to multiple two 5-bit unsigned numbers, i.e. 32 bits => 5 bits etc. Trace the multiplication hardware when multiplying two 5-bit **unsigned** numbers **10101 x 01011**.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Iteration | Multiplicand | Multiplier | Product | Action(s) |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |

(b) Will this multiplication operation issue an overflow warning?

Problem 2: IEEE 754 FP representations - Basics (4 points) Please show your work!

1. What FP number does the following bit pattern represent?

**1 0111 1111 0000 0000 0000 0000 0000 000**

1. What FP number does the following bit pattern represent?

**0 1111 1111 0000 0000 0000 0000 0000 000**

1. For **64-bit double precision** FP representation, how many bits for the exponent part? How many bits for the fraction part?

1. Chapter 3 defines overflow and underflow. What is the book’s definition of FP number **underflow**?

A. A positive exponent becomes too large to fit in the exponent field

B. A negative exponent becomes too large to fit in the exponent field

C. When adding two positive numbers, the result is a negative number

D. When adding two negative numbers, the result is a positive number

E. None of above, please write your own definition.

Problem 3: IEEE 754 Representation – Conversion (6 points) Please show your work!

(1) Convert **-126.625** to single precision FP number.

(2) Convert **0.875** to single precision FP number.

(3) Convert FP number **1100 1100 0011 0011 0000 0000 0000 0000** to base-10 decimal number.

Note: Leave the result in the format of 1.x \* 2y format where x and y are decimal numbers, e.g. 1.72 \* 2-5.

Problem 4: FP operations (6 points)

(1) Show the step-by-step actions of ***adding*** the following two base-10 floating point numbers. No need to convert to binary.

**9.8942 \* 104 + 7.9529 \* 103**

Note: Name the step first, i.e. Step x: Rounding, then illustrate the step using the given numbers.

(2) Show the step-by-step actions of ***multiplying*** the following two base-10 floating point numbers. No need to convert to binary.

**-1.2412\*10-5 \* 3.1002 \* 109**

Note: name the steps and then use the given numbers to illustrate.

(3) For FP multiplication, how do we detect if there is an **overflow**? How do we detect if there is an **underflow**?