CDA 4205 Computer Architecture

Assignment 4: MIPS Programming

Kenneth Neads

U3036-9779

3/3/2019

1. (10 pts) What is the decimal value of the following single-precision floating-point numbers?
   1. **1010 1101 0001 0100 0000 0000 0000 0000** (binary)
      1. **Negative bit, exponent = -37, Mantisa = 1.15625**
      2. **8.412826e-12**
   2. **0100 0110 1100 1000 0000 0000 0000 0000** (binary)
      1. **Positive bit, exponent = 14, Mantisa = 1.5625**
      2. **25600**
2. (10 pts) Show the IEEE 754 binary representation for: -75.4 in …
   1. Single Precision
      1. 1100 0010 1001 0110 1100 1100 1100 0000
   2. Double Precision
      1. 1100 0000 0101 0010 1101 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001 1001
3. (10 pts) Single-precision float-point numbers, and are as follows:

*x* = **1100 0110 1101 1000 0000 0000 0000 0000** (binary) and

*y* = **0011 1110 1110 0000 0000 0000 0000 0000** (binary)

Perform the following operations showing all work:

X = negative int, exponent 14, mantissa 1.6875 = -27648

Y = positive int, exponent -2, mantissa 1.75 = 0.4375

* 1. x + y = -27648 + 0.4375 = 27647.5625
     1. -1.101 1000 0000 0000 0000 0000 x 2^14
     2. +1.110 0000 0000 0000 0000 0000 \* 2^-2
     3. =
     4. 1 0.010 1000 0000 0000 0000 0000 complement
     5. 0 0.000 0000 0000 0000 1110 0000 shift 16
     6. =
     7. – **1.101 0111 1111 1111 0010 0000** x2^14 (added together and complement back)
     8. Final answer = 1100 0110 **1101 0111 1111 1111 0010 0000**
  2. x \* y = -27648 \* 0.4375 = -12096

-2 + 14 = 12 – new exponent = 1000 1100

0.000 0000 0000 0000 0000 0000

* + 1. -1.101 1
    2. + 1.11
    3. =
    4. 1.1011
    5. 11.011
    6. 110.11
    7. =
    8. 10.111 1010 0000 0000 0000 0000
    9. Shift over exponent becomes 13
    10. Final answer 1100 0110 0011 1101 0000 0000 0000 0000

1. (15 pts) Single precision IEEE 754 floating-point numbers,, and are as follows:

*x* = **0101 1111 1011 1110 0100 0000 0000 0000** (in binary) and

*y* = **0011 1111 1111 1000 0000 0000 0000 0000** (in binary) and

*z* = **1101 1111 1011 1110 0100 0000 0000 0000** (in binary)

Perform the following operations.

* 1. x + y
     1. The answer cannot be computed, you will get X as the answer because y is too large a difference
  2. Result of (**a**)+ z
     1. All of the numbers are the same except for the first bit, therefore z is -x
     2. (a) gives us x, so the question is asking x – x, which is equal to 0
  3. Why is the result of (**b**) counterintuitive?
     1. It is counterintuitive because theoretically we should have y left over as the answer, but unfortunately it was cut off and rounded in a previous answer.

1. IA-32 offers an 80-bit extended precision option with a 1 bit sign, 16-bit exponent, and 63-bit fraction (64-bit significand including the implied 1 before the binary point). Assume that extended precision is similar to single and double precision.
   1. (2 pts) What is the bias in the exponent?
      1. 2^15 – 1 = 32767
   2. (3 pts) What is the range (in absolute value) of normalized numbers that can be represented by the extended precision option?
   3. |1.0 \* 2^-32766 - 2.0 \* 2^32766|
2. (10 pts) Using the refined division hardware, show the unsigned division of:

Dividend = 11011001 (binary) by Divisor = 00001010 (binary)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | Step | HI (Remainder) | LO  (Quotient) | Divisor | Difference |
| 0 | Initialize | 0000 0000 | 1101 1001 | 0000 1010 | --- |
| 1 | Shift left, Diff = HI - Divisor | 0000 0001 | 1011 0010 | 0000 1010 | -0 |
| 2 | Shift left, Diff = Hi - Divisor | 0000 0011 | 0110 0100 | 0000 1010 | -0 |
| 3 | Shift left, Diff = Hi - Divisor | 0000 0110 | 1100 1000 | 0000 1010 | -0 |
| 4 | Shift left, Diff = Hi - Divisor | 0000 1101 | 1001 0000 | 0000 1010 | 0000 0011 |
| 4 | Remainder = diff, set lsb of lo | 0000 0011 | 1001 0001 | 0000 1010 | --- |
| 5 | Shift left, Diff = Hi - Divisor | 0000 0111 | 0010 0010 | 0000 1010 | -0 |
| 6 | Shift left, Diff = Hi - Divisor | 0000 1110 | 0100 0100 | 0000 1010 | 0000 0100 |
| 6 | Remainder = diff, set lsb of lo | 0000 0100 | 0100 0101 | 0000 1010 | --- |
| 7 | Shift left, Diff = Hi - Divisor | 0000 1000 | 1000 1010 | 0000 1010 | -0 |
| 8 | Shift left, Diff = Hi - Divisor | 0001 0001 | 0001 0100 | 0000 1010 | 0000 0111 |
| 8 | Remainder = diff, set lsb of lo | 0000 0111 | 0001 0101 | 0000 1010 | --- |

The result of the division should be stored in the Remainder and Quotient registers. Eight iterations are required. Show your steps.

1. (10 pts) Using the refined signed multiplication algorithm, show the multiplication of:

Multiplicand = 00101101 by Multiplier = 11010110 (signed)

The result of the multiplication should be a 16 bit signed number in HI and LO registers. Eight iterations are required because there are 8 bits in the multiplier. Show the steps.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Iteration | STEP | MULTIPLICAND | SIGN | HI | LO |
| 0 | Init(HI = 0, LO = multiplier) | 0010 1101 |  | 0000 0000 | 1101 0110 |
| 1 | Shift left to first 0 |  |  | 0000 0000 | 0110 1011 |
| 2 | LO[0] = 1 => ADD | + | 0 | 0010 1101 | 0110 1011 |
| 2 | Shift (Sign, HI, LO) right 1 bt |  |  | 0001 0110 | 1011 0101 |
| 3 | LO[0] = 1 => ADD | + | 0 | 0100 0011 | 1011 0101 |
| 3 | Shift (Sign, HI, LO) right 1 bt |  |  | 0010 0001 | 1101 1010 |
| 4 | Shift (Sign, HI, LO) right 1 bt |  |  | 0001 0000 | 1110 1101 |
| 5 | LO[0] = 1 => ADD | + | 0 | 0011 1101 | 1110 1101 |
| 5 | Shift (Sign, HI, LO) right 1 bt |  |  | 0001 1110 | 1111 0110 |
| 6 | Shift (Sign, HI, LO) right 1 bt |  |  | 0000 1111 | 0111 1011 |
| 7 | LO[0] = 1 => ADD | + | 0 | 0011 1100 | 0111 1011 |
| 7 | Shift (Sign, HI, LO) right 1 bt |  |  | 0001 1110 | * + - * 1. 1 |
| 8 | LO[0] = 1 => SUB | (+2 compliment) | 1 | 1111 0001 | 0011 1101 |
| 8 | Shift (Sign, HI, LO) right 1 bt |  |  | 1111 1000 | 1001 1110 |

* **Submission Requirements**
* Your solutions must be in a single file with a file name yourname-hw1.
* If scanned from hand-written copies, then the writing must be legible, or loss of credits may occur.
* Only submissions via the link on Canvas where this description is downloaded are graded. Submissions to any other locations on Canvas will be ignored.
* Late submissions are accepted for a maximum of 3 late days with 20% assignment credit off as late penalization. Assignments submitted after 3 late days will not be accepted.