CSCI 2400, Fall 2014

Second Midterm Practice Questions Solutions

1. **[10 Points]** The following problem concerns optimizing a procedure for maximum performance on an Intel Pentium III with the following characteristics of the functional units:

Operation	Latency	Issue Time/Rate
Integer Add	1	1
Integer Multiply	4	1
Floating Point Add	3	1
Floating Point Multiply	5	2
Load or Store (Cache Hit)	1	1

Assume there is one of each functional unit, data1 and data2 have the correct types, e.g. int or floating point. Assume acc1, acc2, out1, and out2 can be stored in registers.

(a) **[5 Points]**

```
int acc1, acc2;
    for (i=0; i< length; i++){
        acc1 = acc1 + data1[i];
        acc2 = acc2 * data2[i];
    }
```

What is the CPE of this loop?

Answer:
Answer: 4 (latency of the multiply)

(b) **[5 Points]**

```
float out1, out2, acc1, acc2;
    for (i=0; i< length; i++){
        out1 = acc1 + data1[i];
        out2 = acc2 * data2[i];
    }
```

What is the CPE of this loop?

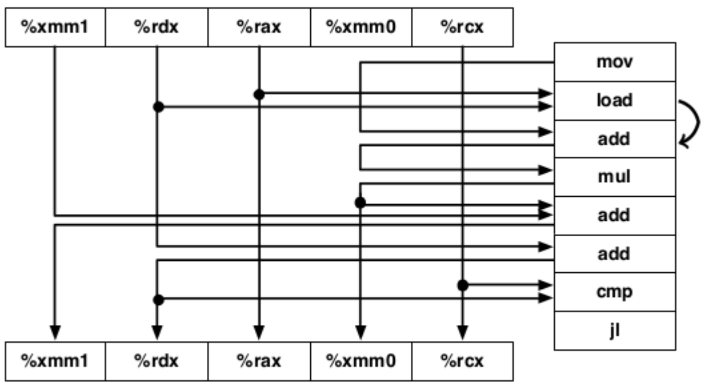
Answer:
Answer: 2 (Issue time of the multiply).

3. **[10 Points]**

Consider the following x86-64 assembly code for an inner loop:

```
L42:
   movs    $0, %xmm0
   adds    (%rax,%rdx,4), %xmm0
   mulss   $2, %xmm0
   adds    %xmm0, %xmm1
   addq    $1, %rdx
   cmpq    %rcx, %rdx
   jl      .L42
```

(a) **[8 Points]** Complete the dataflow diagram below by filling in the blank cells.



(b) **[2 Points]** Which two registers are on the Critical Path for this loop?

Answer:
%rdx, %xmm1

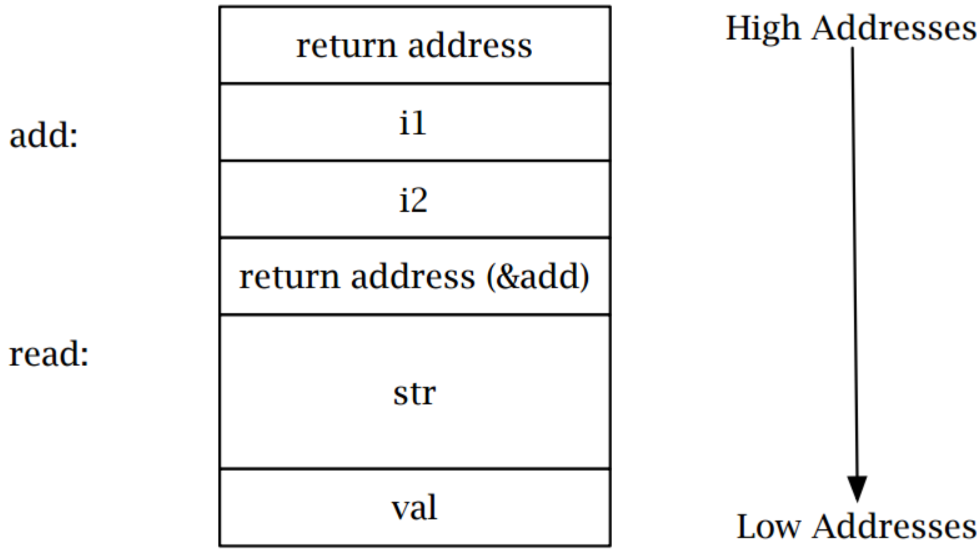


Figure 1:

Figure 2:

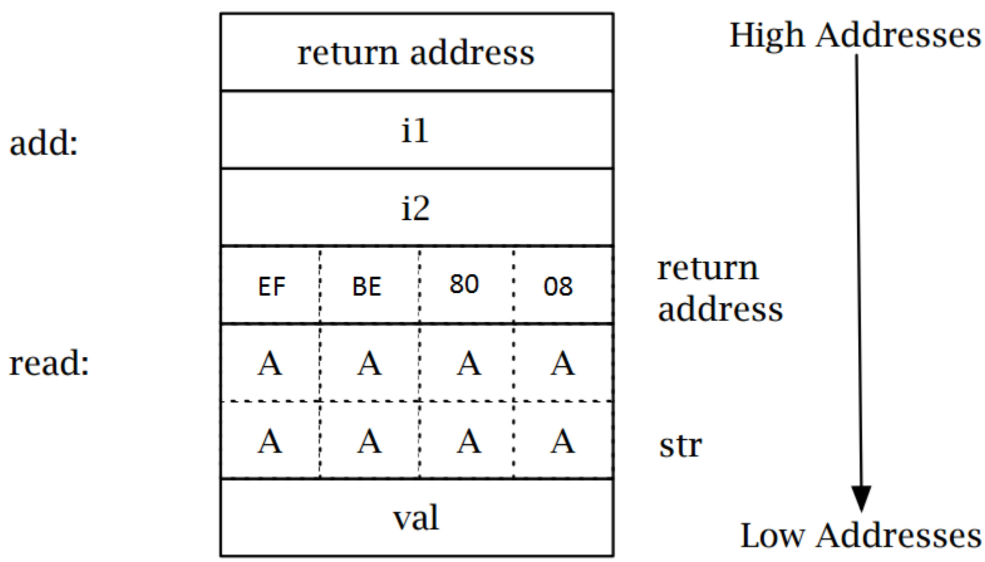


Figure 3:

Figure 4:

3. **[20 Points]** Jane Q. Programmer is working on a weather model program on a computer with 10 bit IEEE floating point numbers that use round-to-even mode. This includes a sign bit, 5-bit exponnent field with bias 16 and a 4 bit mantissa / fractional field).

(a) **[5 Points]** How would you represent the number 200 in this floating point format? You should indicate what portion of the binary digits below are sign, exponent and mantissa components and use the IEEE format as described in the book.

Answer:
0 10111 1001

(b) **[5 Points]** Show what `1000001010` represents in this floating point format? Convert the mantissa portion to a decimal (base 10) number (*e.g.* 1.234) and show the exponent expressed as a power of two. It may be useful to know that $1/2 = 0.5, 1/4 = 0.25, 1/8 = 0.125, 1/16 = 0.0625, 1/32 = 0.03125$. Your resulting number should be formatted something like $1.234 * 2^{56}$, but you should obviously write out the proper value represented by `1000001010`.

______ . ______________________________ $*2^{------}$

Answer:
$-0.625 * 2^{-15}$

(c) **[10 Points]**
The 10-bit float type is specified using `float10`. Jane is concerned about the limited precision of the `float10` data type. She has to add two `float10` numbers together. She wants to have a check to determine if a loss of precision has occured because of the limited range of representable values and the problems that occur when adding very large numbers to very small numbers.
She has a program fragment that appears as below:

```
float10 x = 256;
float10 y = ......;

if ( y <= ______________ ) {
   printf("This value of Y will not change x\n");
} else {
  x = x + y;
}
```

What is the largest value for y that will cause the `printf` to be executed? You should give your value as a decimal number. Remember to account for any needed rounding using the standard round-to-even mode.

Answer:
256 is represented as 0 11000 0000. In binary, this is just $1\ 0000\ 0000_2$. Only the left-most five bits will be represented, or $1\ 0000_2$ with the 1 not being in the final form because this is a normalized number.
Thus, if we add 1111_2 to this, we would get $1\ 0000\ 1111_2$, which would result in the same representation as 256, except that we would round the value to fit to the specified field.
This would be the same as rounding e.g. 0.1111, which would become 1, and change the result. If we use 0.1000 and use the round-to-even rule, the result would be 0, and thus the answer is 8, or the 1000 bit pattern.