CS430 Computer Architecture

Spring 2015

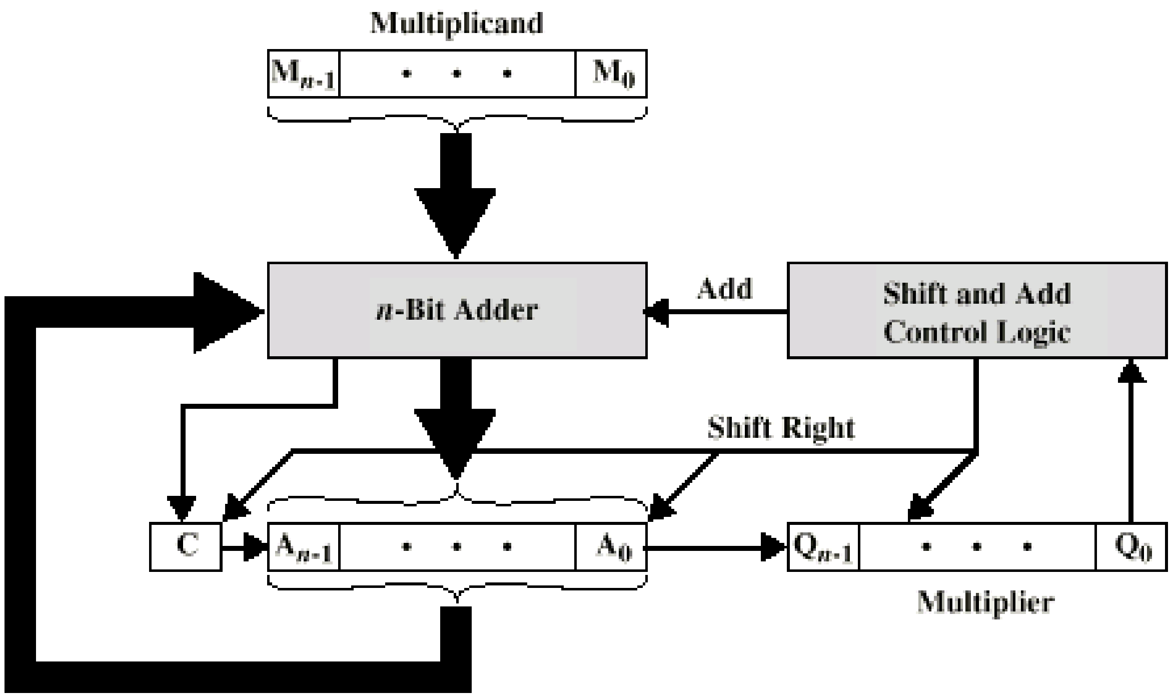
Chapter 10
Computer Arithmetic

- Unsigned Integer Multiplication
  - Read pp. 319-333
- Floating-point numbers
  - Read pp. 341-358

Unsigned Integer Multiplication

```
     1011   Multiplicand (11 dec)
x    1101   Multiplier   (13 dec)
     1011   Partial products
    0000    Note: if multiplier bit is 1 copy
   1011     multiplicand (place value)
  1011      otherwise zero
 10001111   Product (143 dec)
```

Unsigned Integer Multiplication



Unsigned Integer Multiplication

- The multiplier and multiplicand are loaded into Q and M respectively and a third register is needed and initially set to 0.
- The control logic is:
  1. Read multiplier bit one at a time
  2. If Q_0 is 1, then multiplicand is added to A register and the result is stored in A with C used for overflow.
  3. If Q_0 is 0, then no addition is performed.
  4. Shift C, all A, and all Q bits right one bit
  5. Repeat from 1 until each bit in original multiplier is processed

Floating-point Arithmetic

- Very large or very small numbers require floating-point notation.
- Consider 53,760,000 = $.5376 \times 10^8$ or .00005376 = $.5376 \times 10^{-4}$
- This notation is based on the relation $y = a(r^p)$ where:
    - y is the number to be represented
    - a is the mantissa
    - r is the base of the number system (10=decimal; 2=binary)
    - p is the exponent

Floating-point number

- A floating-point number can be divided into three pieces:
  1. sign bit for the entire number
  2. the exponent for the number represented
  3. the mantissa for the number represented

Floating-point number

- Consider a computer with a 12-bit word: s----|s------.
- The first 5 bits from the left represent the exponent (or characteristic) and the remaining 7 bits represent the mantissa (or integer part).
- Both the exponent and mantissa in the above case are represented in signed-magnitude form. They could as well be represented in 2's complement.
- The format of the number expresses a value of: $I \times 2^{C}$

Floating-point number

1. Using this notation, show what the following values might look like:
   a) 4 =
   b) -0.5 =
2. What is the range of the characteristic?
3. What is the range of the mantissa?
4. What is the largest number that can be represented by this 12-bit floating-point number?

Floating-point number

- We can also express the mantissa of the word as a fraction instead of an integer.
- Consider a 12-bit floating-point word as defined above with characteristic C and fraction (mantissa) F. Let's also assume that the binary point is to the left of the magnitude giving us $F \times 2^C$ where F is the binary fraction and C is the characteristic.

```
s----|s.-------
  C        F
```

Floating-point number

- What are the ranges of values that F can have?
  a) `s.-`
  b) `s.--`
  c) `s.---`
  d) `s.------`
- What is the range that the overall number can have? That is, what are the largest and smallest values?

IEEE Standard 754 Floating-point Format

- IEEE 754 adopted in 1985 and revised in 2008
- Let's start with the 1985 version
- There is a 32-bit and 64-bit representation. The principal feature of this representation features the hidden 1 since the numbers are all normalized. We will discuss this in greater detail later on.

IEEE-754 1985

- Single-precision format
- -|--------|-----------------------

S E F

where S is the sign bit, E is a binary integer, and F is a binary fraction of length 23.

- The value of this number is: $(-1)^S x 2^{E-127} x 1.F$
- The representation used for the exponent is called a "biased representation." This means that a fixed value called the bias is subtracted from the binary value of E to get the actual value.

IEEE-754 1985

- What is the bias of the IEEE 754 fp format? What are the ranges that the exponent can represent?

1. Find what value C03E0000 represents using the IEEE-754 FP format.

IEEE-754 1985

- double-precision
- For double-precison, the value of this number is: $(-1)S\text{x}2^{E-1023}\text{x}1.F$

1. Find the value C03E000000000000 represents using the IEEE 754 double-precision FP format.

IEEE-754 1985

- It is important to note that not all bit patterns in the IEEE format are interpreted in the same way. For single-precision:

  1. An exponent of 0 with a fraction of 0 represents +0 or -0 depending on the sign bit.
  2. An exponent of all 1's with a fraction of 0 represents positive or negative infinity.
  3. An exponent of 0 with a nonzero fraction represents a denormalized number.
  4. An exponent of all 1's with a nonzero fraction represents a NaN (not a number) and is used to represent various exceptions.
  5. Exponents in the range of 1-254 with normalized fractions implies the resulting exponent value will be in the range of -126 to +127. Since the number is normalized, we do not need to represent the 1. This bit is implied and called the hidden 1. It is actually a way of adding one more bit of precision to the fraction.

IEEE-754 1985

- Let's go the other way. Find the representation of each of the following in single-precision IEEE-754 FP representation. Express your result in HEX.
  1. -1.0
  2. 1/32
  3. -14.5