CIS 351 - Computer Organization & Assembly Language

Nathan Bowman

Images taken from Harris & Harris book

Binary Numbers and Conversion

The "normal" numbers that we use in everyday life are base-10 (decimal)

Because it is base *10*, we have 10 symbols to represent numbers

0 1 2 3 4 5 6 7 8 9

To go beyond that, we need to use the *location* of a symbol to determine its value

$$56 =$$

$$5 * 10 + 6$$

$$456 =$$

$$4 * 100 + 5 * 10 + 6 =$$

$$4 * 10^2 + 5 * 10^1 + 6 * 10^0$$

So, the fact that we are in base 10 determines that

- we have 10 symbols
- moving left means increasing by a power of 10

We can use any natural number as a base in this way

The ones that will be most useful in this course are base 2 (binary) and base 16 (hexadecimal)

These follow the same simple rules as base 10

Base 2

We have 2 symbols in base 2: 0 and 1

Moving one place to the left means increasing by a power of 2

Because 10, for example, could be either a binary number or a decimal number, we generally use a prefix or suffix to indicate when we use binary

Sometimes you will see b10, more often 10_2 (usually the 2 would be subscripted)

What is the equivalent decimal value to 1011_2?

(Don't forget to start at 2^0 !)

Base 2

$$1011_2$$

$$= 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$$

$$= 8 + 0 + 2 + 1$$

$$= 11$$

Base 2

Note that each 0 or 1 in a binary number is referred to as a **bit**

Bit is short for "binary digit"

Base 16

Base 16 (hexadecimal) works exactly the same way

We have 16 symbols. Because this goes beyond the usual numbers, we start using letters

0 1 2 3 4 5 6 7 8 9 A B C D E

So, 14 in hexadecimal is just D

Moving one place to the left means increasing by a power of 16

The prefix to distinguish hexadecimal numbers is 0x, e.g., 0x1A

Base 16

0x1A

= 1 * 16^1 + A * 16^0

= 16 + 10

= 26

Conversions

Converting *to* base 10 is easy -- just follow the steps from the last slide

Converting *from* base 10 to base 2 (or another base) can be done in two different ways

Learn whichever one is easier for you

We will do all examples in binary because the math is easier, but the same processes apply to hexadecimal -- just be sure to change any 2s in the algorithms to 16s

Base 2 Conversion -- Right-to-Left

Converting from right to left is probably the easier way to perform the conversion

Until you get more comfortable with binary, it's harder to understand *why* this works, so it may be harder for some to remember and easier to make mistakes

Base 2 Conversion -- Right-to-Left

Right-to-left conversion of a number X from base 10 to binary works as follows:

- B <- "" (binary number starts as blank string)
- Loop:
  - R <- X % 2 (remainder of X/2)
  - X <- X/2 (discard any fractional part)
  - prepend R to B (put R on the *left*)
  - if X is 0, you are done
  - otherwise, repeat loop

Base 2 Conversion -- Right-to-Left

Try converting 30 to binary using this method

Base 2 Conversion -- Right-to-Left

X = 30; B = ""

R = 0; X = 15; B = 0

R = 1; X = 7; B = 10

R = 1; X = 3; B = 110

R = 1; X = 1; B = 1110

R = 1; X = 0; B = 11110

B = 11110_2

Base 2 Conversion -- Left-to-Right

Converting from left to right may be more natural for some people

Powers of 2 are 1, 2, 4, 8, 16, 32, ...

Converting X from decimal this way, we are essentially asking "how many 32s are in X?", then "how many 16s?", and so on

We start at large numbers (left), then move to smaller ones (right)

Base 2 Conversion -- Left-to-Right

We will start with an example this time -- convert X=30 to binary

First step is to figure out largest power of 2 we will need

$30 < 32 = 2^5$

So, any bits in the 2^5 place or higher will be 0s

Base 2 Conversion -- Left-to-Right

Our converted number B will have 5 bits (enough for all binary powers from 2^0 to 2^4)

_ _ _ _ _

Base 2 Conversion -- Left-to-Right

_ _ _ _ _

To fill them in, we use the following algorithm:

- Current power is P (starts at 4, in our case)
- repeat until P < 0
    - if X >= 2^P:
        - put a 1 in corresponding blank
        - X <- X - 2^P
    - otherwise, put a 0 in corresponding blank
    - P <- P - 1 (move one blank to the right)

Base 2 Conversion -- Left-to-Right

Try it out yourself

Base 2 Conversion -- Left-to-Right

X = 30

B = _ _ _ _ _

P = 4; 30 >= 2^4; X <- 14 (= 30 - 16)

B = 1 _ _ _ _

P = 3; 14 >= 2^3; X <- 6 (= 14 - 8)

B = 1 1 _ _ _

P = 2; 6 >= 2^2; X <- 2 (= 6 - 4)

B = 1 1 1 _ _

...continued...

$$P = 1;\ 2 >= 2\hat{}1;\ X <- 0\ (= 2 - 2)$$

$$B = 1\ \ 1\ \ 1\ \ 1\ \ \_$$

$$P = 0;\ 0 < 2\hat{}0;\ X <- 0$$

$$B = 1\ \ 1\ \ 1\ \ 1\ \ 0$$

Base 2 Conversion

You can check your answer by converting back to decimal

It doesn't matter which method you choose, but be careful not to mix them up

We will learn more about how to use binary numbers in a future lecture