Number Systems Day 3

Binary addition

$0+0=0$
$0+1=1$
$1+0=1$
$1+1=0$ carry a 1

Binary subtraction

$0-0=0$
$1-0=1$
$1-1=0$
$0-1=1$ but have borrow/regroup a 1 from next column

EX: decimal (base #)

$$\begin{array}{r} 307 \\ -\,134 \\ \hline 173 \end{array}$$

binary (base value)

$$\begin{array}{r} 1101 \\ -\,110 \\ \hline 111 \end{array}$$

$10_2 = 1\times2^1+0\times2^0 = 2_{10}$

Binary Multiplication

$0\cdot0=0$
$0\cdot1=0$
$1\cdot0=0$
$1\cdot1=1$

EX: Multiply in binary.

$$\begin{array}{r} 1101 \\ \times\,110 \\ \hline 0000 \\ 1101 \\ +\,1101 \\ \hline 1001110 \end{array}$$

Binary Division

$0\div1=0$
$1\div1=1$
Cannot divide by zero

EX: Divide in binary.

$$\begin{array}{r} 1.1\,1001\,100 \\ 101\,\overline{)\,1001.00000} \\ -\,101 \\ \hline 1000 \\ -\,101 \\ \hline 110 \\ -\,101 \\ \hline 1000 \\ \vdots \end{array}$$

(repeat)

∴ quotient is $1.\overline{1100}_2$

There is another way to subtract (in all bases) using complements.

First we'll explore the idea of complements in the decimal system before moving on the other systems.

10's complement → subtract each digit from 9 then add 1 to the least digit

e.g. 10's complement of 23:

$$\begin{array}{r} 99 \\ -\,23 \\ \hline 76 \end{array} \qquad \begin{array}{r} 76 \\ +\,1 \\ \hline 77 \end{array}$$

9's complement → subtract each digit from 9

e.g. 9's complement of 23 is

$$\begin{array}{r} 99 \\ -\,23 \\ \hline 76 \end{array}$$

Complements can be used to perform subtraction

To subtract using 10's complements, you add the 10's complement (of the # you are subtracting) to the first number and drop the final carry.

EX:
$$\begin{array}{r} 87 \\ -23 \\ \hline \end{array} \rightarrow \begin{array}{r} \cancel{1}\,87 \\ +77 \\ \hline 64 \end{array}$$

To subtract using 9's complements, you add the 9's complement (of the # you are subtracting) to the first number and then add the final carry to the least digit.

EX:
$$\begin{array}{r} 87 \\ -23 \\ \hline \end{array} \rightarrow \begin{array}{r} 1\,87 \\ +76 \\ \hline 63 \\ +\ \ 1 \\ \hline 64 \end{array}$$

EX: Subtract using both 10's and 9's complements.

$$\begin{array}{r} 2\overset{8}{9}3 \\ -\ 45 \\ \hline 248 \end{array} \qquad \begin{array}{r} \text{10's compl.} \\ \cancel{1}\,293 \\ +\ 55 \\ \hline 248 \end{array} \qquad \begin{array}{r} \text{9's compl} \\ 1\,293 \\ +\ 54 \\ \hline 247 \\ +\ \ \ 1 \\ \hline 248 \end{array}$$

$$\begin{array}{r} 99 \\ -45 \\ \hline 54 \\ +1 \\ \hline 55 \end{array} \begin{array}{l} \\ \\ \leftarrow \text{9's comp.} \\ \\ \leftarrow \text{10's comp.} \end{array}$$

Complements in any base:

We want to avoid having to write the code for subtraction in a programming language because it requires another electrical circuit and circuits are very expensive.

To save money, computers do addition only (they use complements to subtract and work with negative numbers).

In general, there are two types of complements in every number system:

(1) true complement - formed by subtracting each digit of the number from "the radix minus one" and then adding one to the least significant digit of the number formed

In the decimal system, the 10's complement is the true complement.

(2) radix minus one complement - formed by subtracting each digit of the number from "the radix minus one"

In the decimal system, the 9's complement is the radix minus one complement.

In the binary system we have:

1. 2's complement (true compl.)
2. 1's complement (radix minus one compl.)

Binary Complements (much easier to find)

2's complement is formed by changing every 1 to a 0, every 0 to a 1, and adding 1 to the least bit

"bit" = binary digit

EX: Subtract using 2's complements (in binary)

$$\begin{array}{r} 11011 \\ -\,10100 \\ \hline 111 \end{array} \quad \rightarrow \quad \begin{array}{r} 11011 \\ +\,01100 \\ \hline 00111 \end{array}$$

drop final carry

$$\begin{array}{r} 01011 \\ +\quad\ \ 1 \\ \hline 01100 \end{array}$$ 2's compl.

EX: Subtract in binary using 2's complements.

add a zero!

$$\begin{array}{r} 11011 \\ -\,01100 \\ \hline \end{array} \quad \rightarrow \quad \begin{array}{r} 11011 \\ +\,10100 \\ \hline 01111 \end{array}$$

2's compl

$$01100 \rightarrow \begin{array}{r} 10011 \\ +\quad\ \ 1 \\ \hline 10100 \end{array}$$

1's complement is formed by changing each 0 to a 1 and each 1 to a 0

EX:

$$\begin{array}{r} 11001 \\ -\,10110 \\ \hline 00011 \end{array} \quad \rightarrow \quad \begin{array}{r} 11001 \\ +\,01001 \\ \hline 00010 \\ +\quad\ \ 1 \\ \hline 11 \end{array}$$

add final carry to least bit

The Octal (base 8) and Hexadecimal (base 16) number systems help us condense binary numbers into numbers that are more manageable.

Decimal system isn't so helpful (it makes it hard to "see" the switches)

Octal System (base 8)

We want to be able to convert quickly

binary → octal

and octal → binary

without having to go through decimal.

binary → octal

- group binary digits in groups of 3 (starting at radix point and working outwards)
- read each group of 3 as an octal digit

reason:

$\underbrace{1\ 1\ 1}_{2^2\ 2^1\ 2^0}$.

$4+2+1=7 \leftarrow$ largest digit in octal

EX: $111110111_2 \rightarrow$ base 8

$\underbrace{111}_{2^2+2^1+2^0}\ \underbrace{110}_{2^2+2^1}\ \underbrace{111}_{2^2+2^1+2^0}{}_2 = 767_8$

7 6 7

EX: convert 11101_2 to octal.

$\underbrace{011}_{2^1+2^0}\ \underbrace{101}_{2^2+2^0}{}_2 = 35_8$

3 5