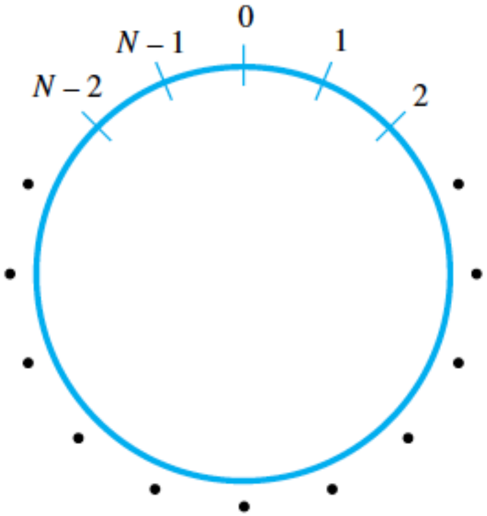
Integer Addition and Subtraction

We will review the algorithms that the ALU must implement

Addition & subtraction

Multiplication & division

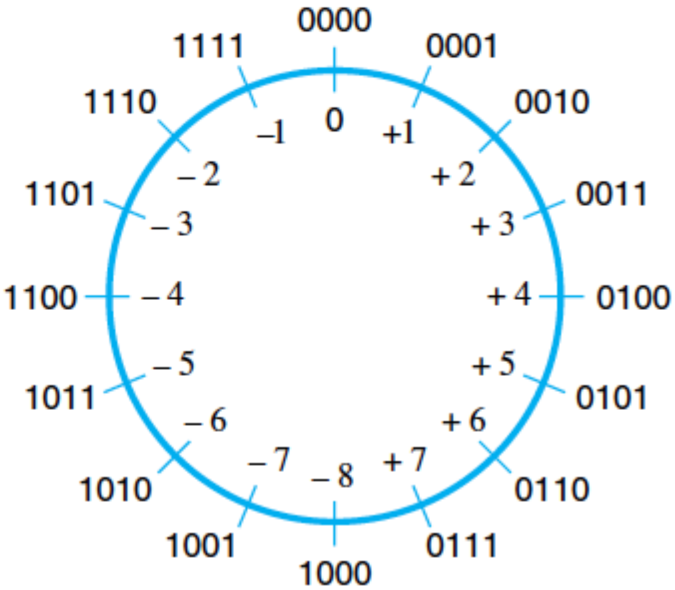
We will also show why two's complement is preferred



An incremented unsigned integer rolls over from the max value back to 0. (modulo N)

Basis for Two’s Complement

Stepping counter-clockwise from 0 goes to -1



Stepping clockwise from 0 goes to +1

In this 4-bit system there are 16 possible values (modulus=16)

To add M to a value, perform M clockwise steps

To subtract M from a value, perform M counter-clockwise steps
this is equivalent to 16-M clockwise steps
the same as adding the two’s complement of -M

Integer Addition and Subtraction

Using hex makes dealing with more bits easier

Each hex digit (0,..,9,A,B,C,D,E,F) corresponds to a group of 4 bits

Example: assume 16-bit numbers:

$$\begin{array}{r} -\ 9 \\ \underline{+22} \\ 13 \end{array} \quad \rightarrow \quad \begin{array}{r} \text{0xFFF7} \\ \underline{+\text{0x0016}} \\ \text{0x000D} \end{array}$$

modulus = 2^{16} = 65536 = 0x10000

-9 in two’s complement = 65536 – 9 = 65527 = 0xFFF7

$$\begin{array}{r} 350 \\ \underline{+922} \\ 1272 \end{array} \quad \rightarrow \quad \begin{array}{r} \text{0x015E} \\ \underline{+\text{0x039A}} \\ \text{0x04F8} \end{array}$$

Integer Addition and Subtraction

In two's complement addition:
  add the two bit patterns
  ignore any carry out of the leftmost bit

Subtract by adding the two's complement of the subtrahend

Only negative values need to be complemented

Two's complement is preferred over one's complement
    One's complement has both +0 and -0
    One's complement addition requires an end-around carry

```
 - 9             0xFFF6        modulus-1 =  2^16 - 1  =  65535 = 0xFFFF
 +22      →     +0x0016
 ---            -------
  13             0x000C
```

The carry must be added into the result to obtain 0x000C + 1 = 0x000D = 13

Integer Addition and Subtraction

Using sign and magnitude representation complicates arithmetic
extra operations are required (sign checking & absolute value)

For addition or subtraction:
check the signs of the two numbers
if they differ, subtract the smaller number from the larger
use the sign of the larger as the sign for the result

For multiplication or division:
check the signs of the two numbers
if they differ, the result is negative
compute using the absolute values of the two numbers

So two's complement representation and arithmetic is preferred