Arithmetic Circuits

(Part I)

Randy H. Katz
University of California, Berkeley

Spring 2007

Motivation

Arithmetic circuits are excellent examples of comb. logic design

- *Time vs. Space Trade-offs*

  Doing things fast requires more logic and thus more space

  Example: carry lookahead logic

- *Arithmetic Logic Units*

  Critical component of processor datapath

  Inner-most "loop" of most computer instructions

Overview

- *Binary Number Representation*

  Sign & Magnitude, Ones Complement, Twos Complement
- *Binary Addition*

  Full Adder Revisted
- *ALU Design*
- *BCD Circuits*
- *Combinational Multiplier Circuit*
- *Design Case Study: 8 Bit Multiplier*
- *Sequential Multiplier Circuit*

Number Systems

Representation of Negative Numbers

Representation of positive numbers same in most systems

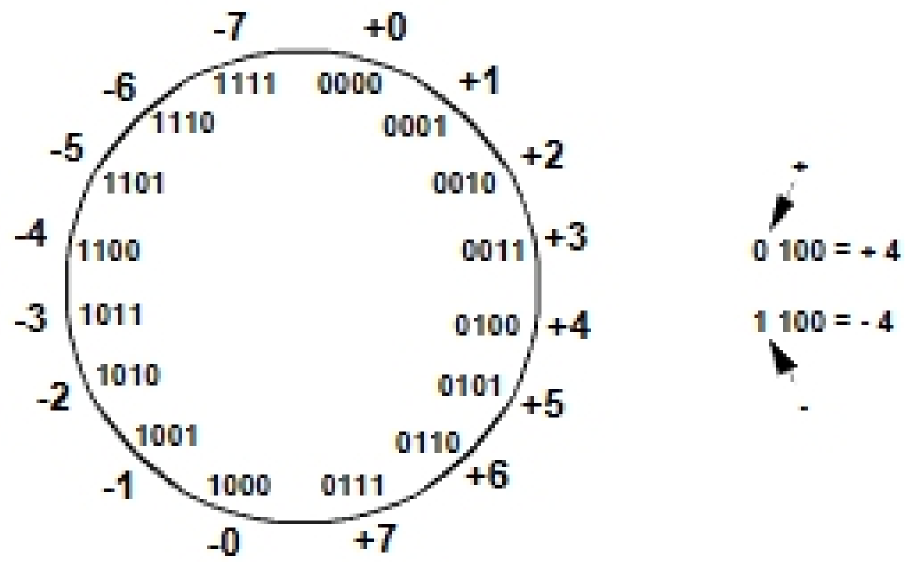
Major differences are in how negative numbers are represented

Three major schemes:
- sign and magnitude
- ones complement
- twos complement

Assumptions:
- we'll assume a 4 bit machine word
- 16 different values can be represented
- roughly half are positive, half are negative

Number Systems

Sign and Magnitude Representation



High order bit is sign: 0 = positive (or zero), 1 = negative

Three low order bits is the magnitude: 0 (000) thru 7 (111)

Number range for n bits = $+/-2^{n-1} -1$

Representations for 0

Number Systems

Sign and Magnitude

Cumbersome addition/subtraction

Must compare magnitudes to determine sign of result

Ones Complement

N is positive number, then $\overline{N}$ is its negative 1's complement

$$\overline{N} = (2^n - 1) - N$$

Example: 1's complement of 7

2^4 = 10000

-1 = 00001

1111

-7 = 0111

1000 = -7 in 1's comp.

Shortcut method:

simply compute bit wise complement

0111 -> 1000