EECS 281, January 20, 2015

Signed - magnitude Representation

$+102$, -42

For binary numbers: use an extra bit position to represent the sign:

sign bit

MSB is used as the sign bit:

0 : plus   1 : minus

signed-magnitude rep.

$$\begin{array}{c|ccccccc} & 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array}$$

$$\underbrace{\boxed{0}}_{\text{sign bit}}\underbrace{1\,0\,1\,0\,1\,0\,1}_{\text{magnitude bits}}{}_2 = +85_{10}$$

$$0000\,0000_2 = +0_{10}$$

$$1000\,0000_2 = -0_{10}$$

has equal number of positive and negative integers.

n-bit signed-magnitude integer range:

$$-(2^{n-1}-1) \text{ to } +(2^{n-1}-1)$$

two possible repres. of 0.

Complement Number Systems

Negates a number by taking its complement.

radix: r, complement of an n-digit number D

$$r^n - D$$

Define complement of a digit: $r-1-d$

$\Rightarrow$ $r^n - 1 - D$ obtained by complementing digits of D.

$\Rightarrow$ radix complement: complement individual digits and add 1.

Two's complement Representation:

MSB: serves as sign bit.

MSB = 1 $\Rightarrow$ negative number

8-bit example.

e.g. $17_{10} = 00010001_2$ $\qquad -17_{10} = ?_2$

complement bits:

$$00010001$$
$$\downarrow$$
$$11101110$$

Add 1 : $+ \qquad\qquad 1$

$$11101111_2$$

$1110111 1_2$.

weight of MSB is -2^{n-1}

$$\begin{array}{cccccccc} -128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1_2 \end{array} =$$

$$-128 + 64 + 32 + 8 + 4 + 2 + 1 = -17_{10}.$$

$$1000\,0001_2 = -127_{10}.$$

complement

$$\begin{array}{r} 0111\,1110 \\ +\qquad\qquad 1 \\ \hline 0111\,1111 \end{array} = 127_{10}.$$

$$0000\,0000$$

complement

$$\begin{array}{r} 1111\,1111 \\ +\qquad\qquad 1 \\ \hline \cancel{1}\,0000\,0000 \end{array} = 0000\,0000$$

carryout from MSB

Zero is considered positive because sign bit is 0.

one extra negative number -2^{n-1}, That does not have a positive counterpart.

$$-128_{10} = 10000000_2$$

$\Downarrow$ complement

$$\begin{array}{r} 01111111 \\ +\quad\quad\quad 1 \\ \hline 10000000 \end{array} \rightarrow -128.$$

Ones' complement:

MSB is the sign: 1: negative.

Just complement the bits!

weight of MSB $-(2^{n-1}-1)$

Table 2-6 Decimal and 4-bit numbers.

Decimal	*Two's Complement*	*Ones' Complement*	*Signed Magnitude*	*Excess 2^{m-1}*
−8	1000	—	—	0000
−7	1001	1000	1111	0001
−6	1010	1001	1110	0010
−5	1011	1010	1101	0011
−4	1100	1011	1100	0100
−3	1101	1100	1011	0101
−2	1110	1101	1010	0110
−1	1111	1110	1001	0111
0	0000	1111 or 0000	1000 or 0000	1000
1	0001	0001	0001	1001
2	0010	0010	0010	1010
3	0011	0011	0011	1011
4	0100	0100	0100	1100
5	0101	0101	0101	1101
6	0110	0110	0110	1110
7	0111	0111	0111	1111

$n = 4$

Range -2^{n-1} to $2^{n-1} - 1$

$-2^{3} = -8$ to $2^{3} - 1 = +7$

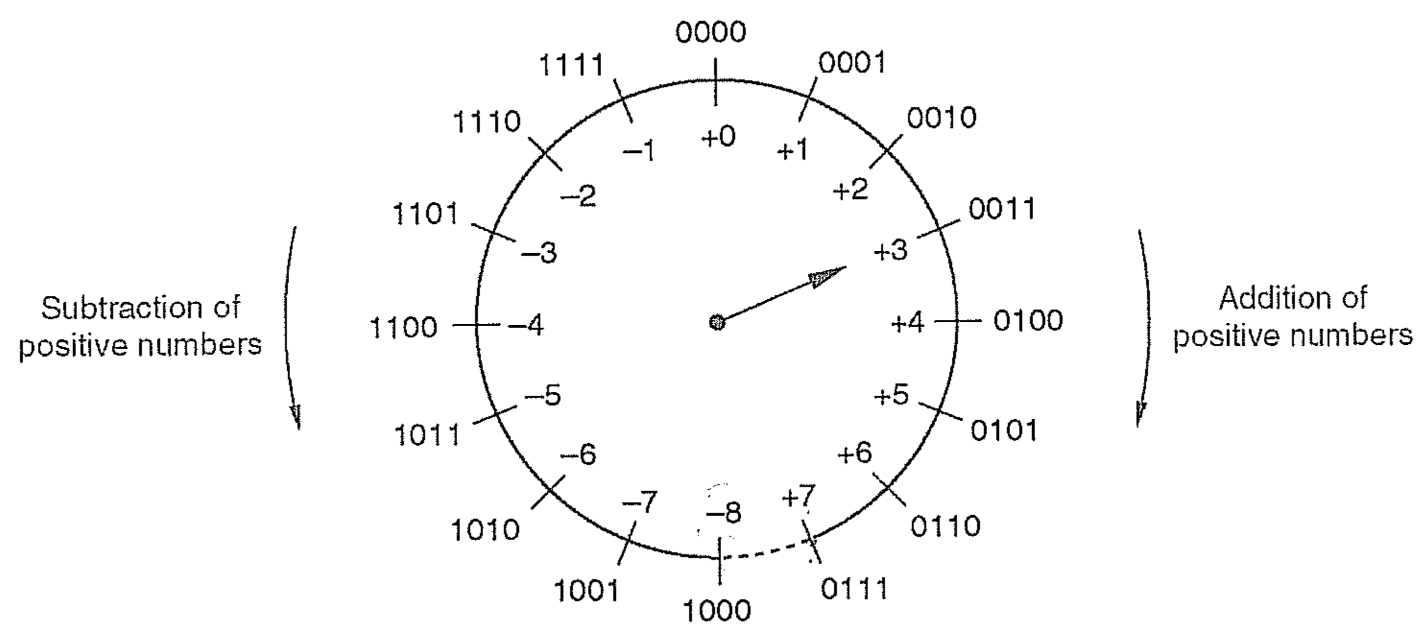

Figure 2-3

A modular counting representation of 4-bit two's-complement numbers.

From *Digital Design: Principles and Practices*, Fourth Edition, John F. Wakerly, ISBN 0-13-186389-4.

$17_{10} = 00010001_2$

$-17_{10} = ?$

11101111_2

$$\begin{array}{cccccccc} -127 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \end{array}_2 = ?_{10}$$

Addition Rules in Two's Complement Number System:

Two's complement numbers can be added by ordinary binary addition. Ignore carries beyond MSB.

Result will be always be the correct sum as long as the range of the number system is not exceeded.

overflow: if addition result exceeds range ⇒ overflow.

If you are adding two numbers with different signs → never produce an overflow.

$$\begin{array}{r} -3 \\ + \;\; -6 \\ \hline -9 \end{array} \rightarrow \text{out of range!}$$

use four bit two's comp. representation:

$$\begin{array}{r} 0011 \leftarrow +3 \\ 1100 \\ + \quad\;\; 1 \\ \hline 1101 \leftarrow -3 \end{array} \qquad -3 : 1101 \qquad -6 : 1010 \qquad \begin{array}{r} 0110 \leftarrow +6 \\ 1001 \\ + \quad\;\; 1 \\ \hline 1010 \leftarrow -6 \end{array}$$

$c_{in} = 0$

$$\begin{array}{r} \boxed{1}101 \\ + \;\; \boxed{1}010 \\ \hline 1\,\underline{\underline{0}}111 \end{array}$$

cout = 1

overflow.

overflow detection: addends' signs are same
Sum's sign is different from the addends.

If c_{in} into and c_{out} out of sign bit are different.

Example:

$$\begin{array}{r} 11010100 \\ +\ 11101011 \\ \hline ? \end{array}$$

$$\begin{array}{r} \textcircled{1} \\ 11010100 \\ +\ 11101011 \\ \hline \cancel{1}\,10111111 \end{array}$$

No overflow.

Example: $+25_{10} = ?$ <u>8 bits.</u>

Signed magn. 00011001

Two's comp. 00011001

Ones' comp. 00011001

$25 \div 2 = 12$ 1.

$12 \div 2 = 6$ 0

0.

$6 \div 2 = 3$ 1.

$3 \div 2 = 1.$ 1.

$-6_{10} = ?_2$ 8-bits.

signed mag. 10000110

Two's comp. 11111010

Ones' comp.

00000110. : +6

signed mag : 10000110 : -6

two's comp : 11111001

$+$ 1

11111010 ← -6.

Ones' comp : 11111001

$\overset{-128}{1}\,\overset{64}{1}\,\overset{32}{0}\;\overset{16}{1}\,\overset{8}{0}\,\overset{4}{1}\,\overset{2}{1}\,\overset{1}{0}_2 = ?_{10}$ represented in two's comp. representation.

$-128 + 64 + 16 + 4 + 2 = -42$

$-(2^{n-1}-1)$

$\overset{-127}{1}\,0\,0\;\overset{16}{1}\,0\,0\,0\,0_2 = ?_{10}$ represented in ones' comp.

-111_{10}.