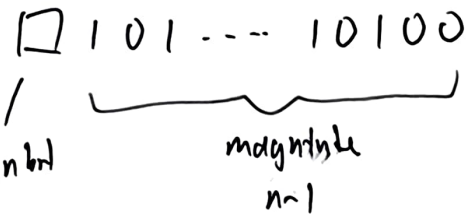
subtraction can be performed by adding a negative number

computer only needs adder to perform all arithmetic operations

everything is add or shift only

signed binary number

most significant bit is reserved as sign bit



0 → positive number

1 → negative number

n-bit signed binary has n-1 magnitude bits

range: $-(2^{n-1}-1)$ to $+(2^{n-1}-1)$

3	011
2	010
1	001
+0	000
-0	100
-1	101
-2	110
-3	111

} problem ⇒ 2 representations for 0

subtraction is performed by adding a negative number

$$\begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array} \equiv \begin{array}{r} 0011 \\ +1110 \\ \hline 10001 \end{array}$$

discarded

$$\begin{array}{r} 3 \\ -1 \\ \hline 2 \end{array} \quad \begin{array}{r} 011 \\ +101 \\ \hline 1000 \end{array} \quad X$$

signed magnitude representation cannot be used for addition of 2 numbers with opposite sign when using a simple adder

complement representation

radix complement
of n-digit integer A with radix r

$$A^* = r^n - A$$

diminished radix complement

$$A^* = r^n - A - 1$$

Decimal $A = 2375 \Rightarrow A^* = 10000 - 1 - 2375 = 7624$

Binary $A = 100|_2 \Rightarrow A^* = \underbrace{10000_2 - 1}_{1111_2} - 100|_2$

$= 1111_2 - 100|_2$

$= 011|_2$

diminished radix 2 complement found by reversing bits.

known as 1's Complement

3	011
2	010
1	001
+0	000
-0	111
-1	110
-2	101
-3	100

$$\begin{array}{r} 3 \\ -\ 2 \\ \hline 1 \end{array} \qquad \begin{array}{r} 10'11 \\ +\ 101 \\ \hline 1000 \\ \hline \end{array} \quad \curvearrowright 1$$

can perform subtraction, but need to shift and add the carry

still multiple representation for +0 & -0

2's complement is the radix complement of binary number.

add 1 to 1's complement

$$A^* = 2^n - A$$
$$= (2^n - A - 1) + 1$$

reverse | add 1

2's complement

pos → neg
flip + 1

neg → pos
−1 + flip
OR
flip + 1 ⇒ negate again

3	011
2	010
1	001
0	000
−1	111
−2	110
−3	101
−4	100

no multiple representation

no problem in subtraction

$$\begin{array}{r} 3 \\ -2 \\ \hline 1 \end{array} \qquad \begin{array}{r} 011 \\ +110 \\ \hline \cancel{1}001 \end{array}$$

carry is discarded

more hardware efficient (no need shift and add carry)

magnitude range -2^{n-1} to $2^{n-1}-1$

MSB 1 = $1 \times -2^{N-1}$ (negative number)

MSB 0 = $0 \times -2^{N-1}$ (positive number)

most negative ⇒ $-2^{N-1} + 0$ 1000...00

most positive ⇒ $0 + 2^{N-1} - 1$ 0111...11

all bits are 1 ⇒ 111...11 = $-2^{N-1} + 2^{N-1} - 1 = -1$ (least -ve)

all bits are 0 ⇒ 000...00 = 0 (least +ve)

Binary coded decimal

each decimal digit is represented by a 4-bit binary numbe

(0-9)

$(5\ 9\ 8)_{10}$

0101 1001 1000

$$\begin{array}{r} 5 \\ +7 \\ \hline 12 \end{array} \equiv \begin{array}{r} 0101 \\ 0111 \\ \hline 1100 \end{array} \longrightarrow$$

$(12)_{10}$ in BCD representation is

0001 0010

X not BCD code (>9)

decimal addition is modulo 10 and carry is generated when sum > 9

4 bit binary only generate carry when sum > 15

generate carry for BCD when sum > 9 $\Rightarrow$ add 6 to result ?

0110

$$\begin{array}{r} 5 \\ +7 \\ \hline 12 \end{array} \quad \begin{array}{r} 0101 \\ +0111 \\ \hline 1100 \\ +0110 \\ \hline 10010 \end{array}$$

$(1\ 2)_{10}$

S = A+B using BCD adder

If $S \leq 9 \Rightarrow$ sum = S, carry = 0

If $S > 9 \Rightarrow$ sum = S+6, carry = 1