Number Systems

Baback Izadi

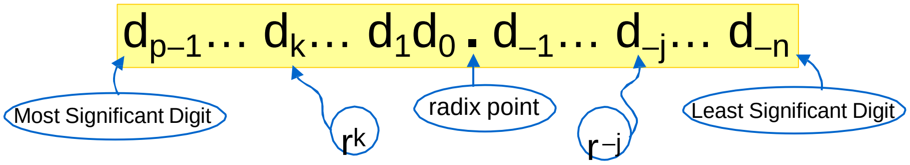
Division of Engineering Programs

bai@engr.newpaltz.edu

Number Systems

A positional number system has a

radix (or base of the number) any integer r ≥ 2



$$d_{p-1}\dots\ d_k\dots\ d_1 d_0\ .\ d_{-1}\dots\ d_{-j}\dots\ d_{-n}$$

Most Significant Digit

r^k

radix point

r^{-j}

Least Significant Digit

$$D = \sum_{i=-n}^{p-1} d_i \cdot r^i \qquad 0 \le d_i \le (r-1)$$

Example: 25.375 radix 10

$\mathbf{2}\cdot 10^1 + \mathbf{5}\cdot 10^0 + \mathbf{3}\cdot 10^{-1} + \mathbf{7}\cdot 10^{-2} + \mathbf{5}\cdot 10^{-3}$

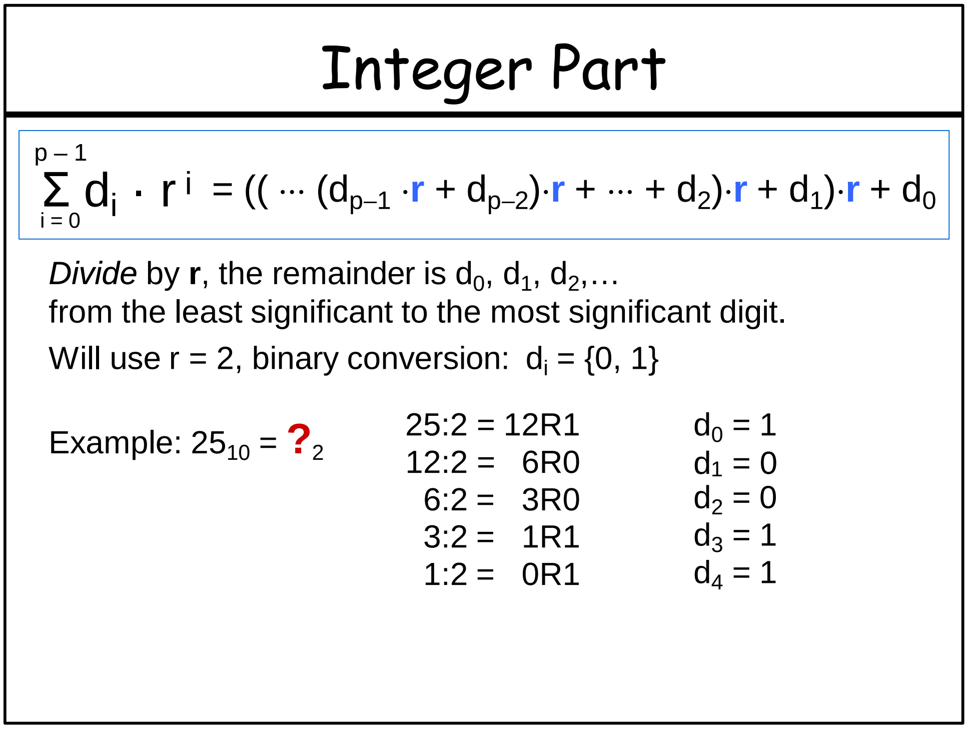
The integer and the fractional part are processed separately.

Powers of 2: 2^n

It will be convenient to remember these powers:

n	2^n
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024

n	2^n
–1	0.5
–2	0.25
–3	0.125



Integer Part

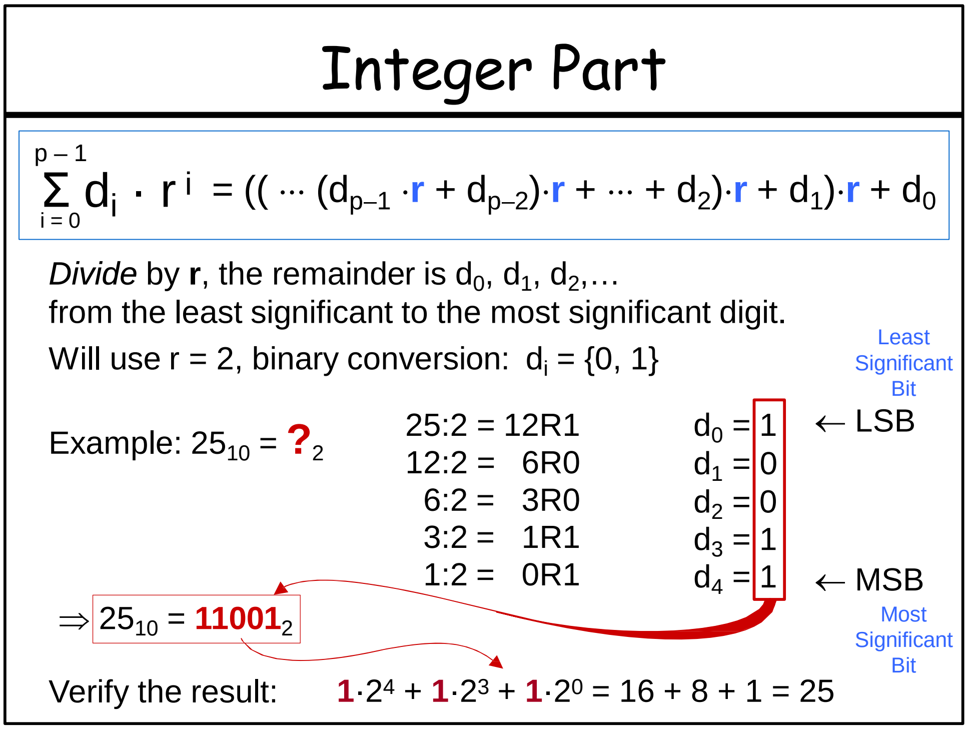
$$\sum_{i=0}^{p-1} d_i \cdot r^{i} = ((\cdots(d_{p-1}\cdot \mathbf{r} + d_{p-2})\cdot \mathbf{r} + \cdots + d_2)\cdot \mathbf{r} + d_1)\cdot \mathbf{r} + d_0$$

Divide by **r**, the remainder is d_0, d_1, d_2,…
from the least significant to the most significant digit.

Will use r = 2, binary conversion: $d_i = \{0, 1\}$

Example: $25_{10} = ?_2$

25:2 = 12R1	$d_0 = 1$
12:2 = 6R0	$d_1 = 0$
6:2 = 3R0	$d_2 = 0$
3:2 = 1R1	$d_3 = 1$
1:2 = 0R1	$d_4 = 1$



Integer Part

$$\sum_{i=0}^{p-1} d_i \cdot r^{i} = ((\cdots(d_{p-1}\cdot \mathbf{r} + d_{p-2})\cdot \mathbf{r} + \cdots + d_2)\cdot \mathbf{r} + d_1)\cdot \mathbf{r} + d_0$$

Divide by **r**, the remainder is d_0, d_1, d_2,…
from the least significant to the most significant digit.

Will use r = 2, binary conversion: $d_i = \{0, 1\}$

Example: $25_{10} = ?_2$

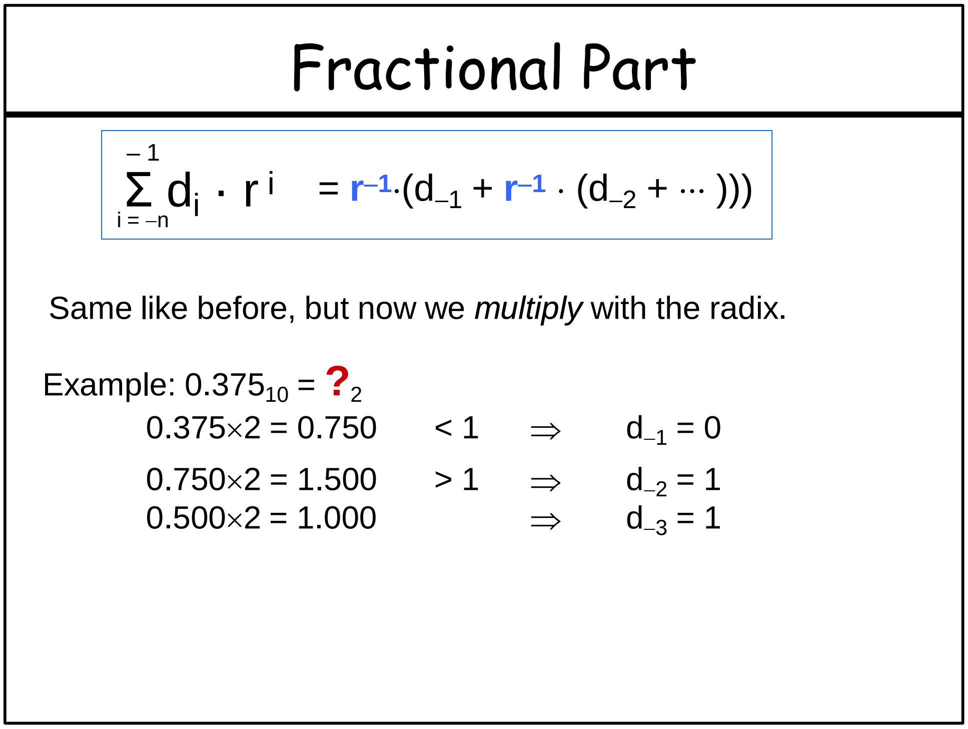
Least Significant Bit

25:2 = 12R1	$d_0 = 1$	← LSB
12:2 = 6R0	$d_1 = 0$	
6:2 = 3R0	$d_2 = 0$	
3:2 = 1R1	$d_3 = 1$	
1:2 = 0R1	$d_4 = 1$	← MSB

Most Significant Bit

⇒ $\boxed{25_{10} = \mathbf{11001}_2}$

Verify the result: $\mathbf{1}\cdot 2^4 + \mathbf{1}\cdot 2^3 + \mathbf{1}\cdot 2^0 = 16 + 8 + 1 = 25$



Fractional Part

$$\sum_{i=-n}^{-1} d_i \cdot r^{i} = r^{-1}\cdot(d_{-1} + r^{-1} \cdot (d_{-2} + \cdots)))$$

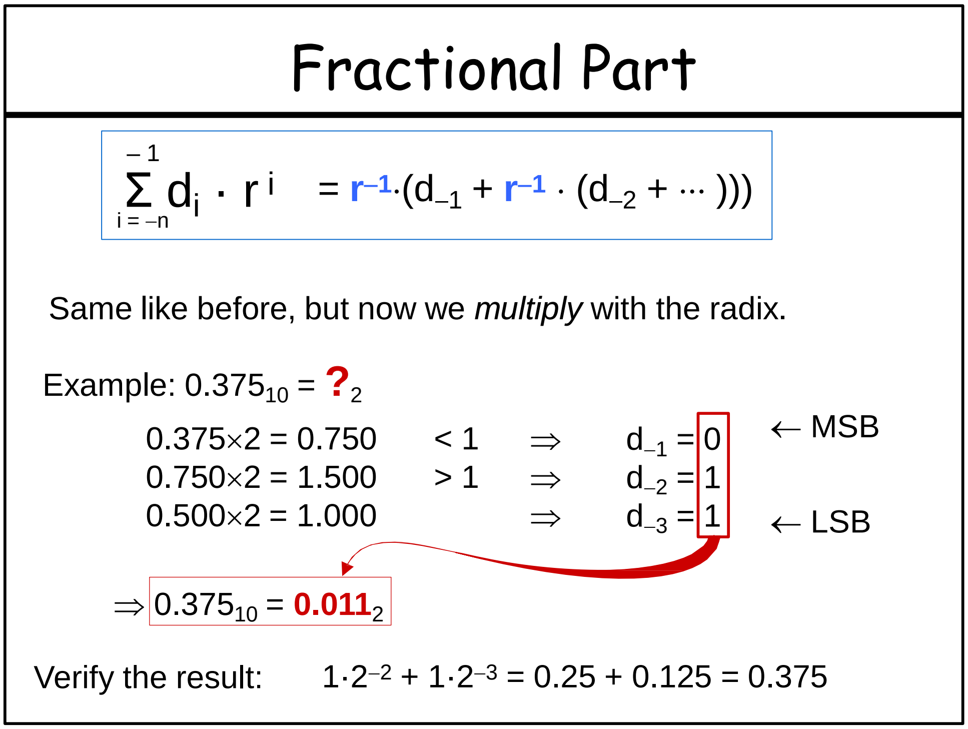
Same like before, but now we *multiply* with the radix.

Example: $0.375_{10} = ?_2$

$0.375\times2 = 0.750 \quad < 1 \quad \Rightarrow \quad d_{-1} = 0$

$0.750\times2 = 1.500 \quad > 1 \quad \Rightarrow \quad d_{-2} = 1$

$0.500\times2 = 1.000 \quad \Rightarrow \quad d_{-3} = 1$



Fractional Part

$$\sum_{i=-n}^{-1} d_i \cdot r^{i} = r^{-1}\cdot(d_{-1} + r^{-1} \cdot (d_{-2} + \cdots)))$$

Same like before, but now we *multiply* with the radix.

Example: $0.375_{10} = ?_2$

$0.375\times2 = 0.750 \quad < 1 \quad \Rightarrow \quad d_{-1} = 0$ ← MSB

$0.750\times2 = 1.500 \quad > 1 \quad \Rightarrow \quad d_{-2} = 1$

$0.500\times2 = 1.000 \quad \Rightarrow \quad d_{-3} = 1$ ← LSB

$\Rightarrow \boxed{0.375_{10} = \mathbf{0.011}_2}$

Verify the result: $1\cdot2^{-2} + 1\cdot2^{-3} = 0.25 + 0.125 = 0.375$

Important Radices

Radices important to computer engineers are: r = **2**, **8**, **16**

Binary	Decimal	Octal	3-Bit String	Hexadecimal	4-Bit String
0	0	0	000	0	0000
1	1	1	001	1	0001
10	2	2	010	2	0010
11	3	3	011	3	0011
100	4	4	100	4	0100
101	5	5	101	5	0101
110	6	6	110	6	0110
111	7	7	111	7	0111
1000	8	10	—	8	1000
1001	9	11	—	9	1001
1010	10	12	—	A	1010
1011	11	13	—	B	1011
1100	12	14	—	C	1100
1101	13	15	—	D	1101
1110	14	16	—	E	1110
1111	15	17	—	F	1111

Example: $11100001.011_2 = \underbrace{011}\,\underbrace{100}\,\underbrace{001}\,.\,\underbrace{011}_2 = 341.3_8$

$341.3_8 = 3{\cdot}8^2 + 4{\cdot}8^1 + 1{\cdot}8^0 + 3{\cdot}8^{-1} = 225.375_{10}$

Fourth digit was added to the fractional part

$11100001.011_2 = 1110\ 0001\ .\ 011\boxed{0}_2 = E1.6_{16}$

$E1.6_{16} = 14{\cdot}16^1 + 1{\cdot}16^0 + 6{\cdot}16^{-1} = 225.375_{10}$

Binary Addition

c_{in}	x	y	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

s = sum
c_{in} = carry in
c_{out} = carry out

X, Y, C_{in} → s, C_{out}

Decimal:

$1_{10} + 1_{10} = 2_{10}$

Binary:

$1_2 + 1_2 = 10_2$

⬇

$1_2 + 1_2 = \boxed{1}\boxed{0}_2$

sum

carry out

X + Y + C_{in} = s	C_{out}
0 + 1 + 0 = 1	0
1 + 0 + 1 = 0	1
1 + 1 + 1 = 1	1

Binary Addition

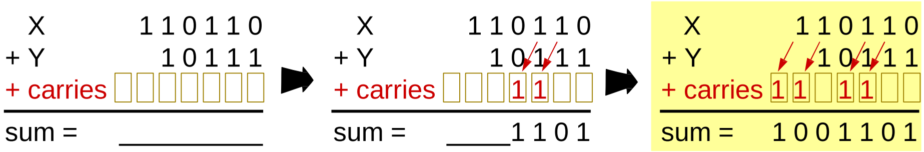
c_{in}	x	y	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

c_{in} = carry in
c_{out} = carry out

X, Y, C_{in} → s, C_{out}

augend
+ addend
= sum

Example addition:

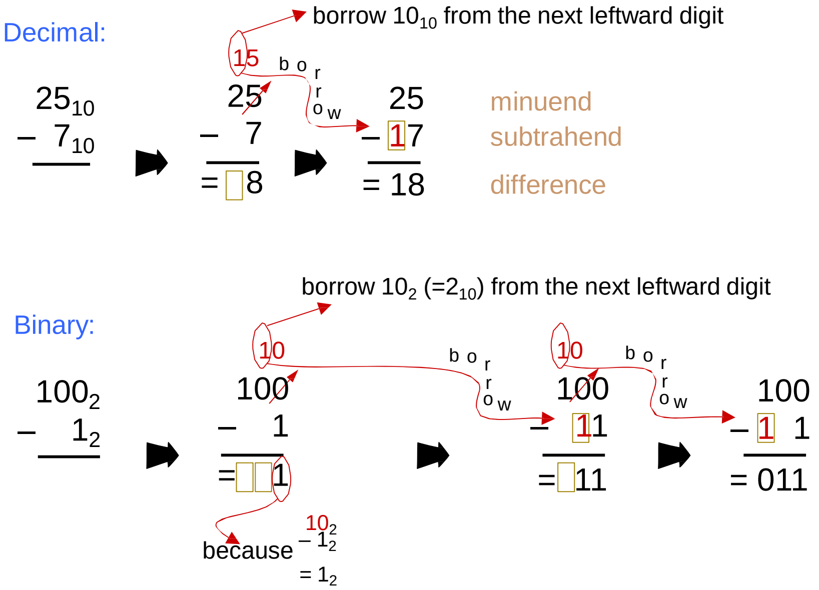


X 1 1 0 1 1 0
+ Y 1 0 1 1 1
+ carries □□□□□□□
sum = ________

X 1 1 0 1 1 0
+ Y 1 0 1 1 1
+ carries □□□ 1 1 □□
sum = ____1 1 0 1

X 1 1 0 1 1 0
+ Y 1 0 1 1 1
+ carries 1 1 □ 1 1 □□
sum = 1 0 0 1 1 0 1

Subtraction



Decimal:

borrow 10_{10} from the next leftward digit

25_{10}
$-\ 7_{10}$

15
borrow
25
− 7
= □8

25
− 17
= 18

minuend
subtrahend
difference

Binary:

borrow 10_2 ($=2_{10}$) from the next leftward digit

100_2
$-\ 1_2$

10
borrow
100
− 1
= □□1

because 10_2 $-\ 1_2$ $= 1_2$

10
borrow
100
− 11
= □11

100
− 1 1
= 011

Binary Subtraction

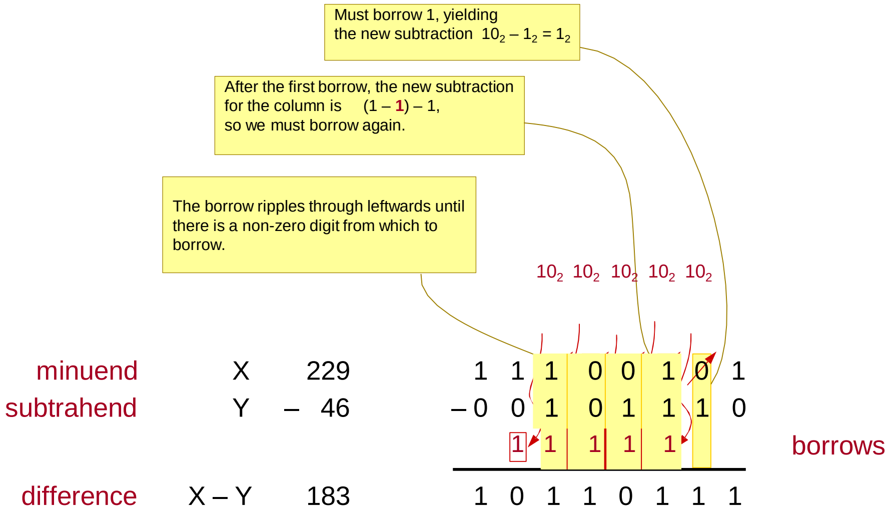
b_{in}	x	y	b_{out}	d
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

b_{in} = borrow in
b_{out} = borrow out

X, Y, B_{in} → s, B_{out}

$X - Y - B_{in} = d$	B_{ou}
0 – 1 – 0 = 1	^{t}1
1 – 0 – 1 = 0	1
1 – 1 – 1 = 1	0

Example Subtraction (1)



Must borrow 1, yielding the new subtraction $10_2 - 1_2 = 1_2$

After the first borrow, the new subtraction for the column is (1 – **1**) – 1, so we must borrow again.

The borrow ripples through leftwards until there is a non-zero digit from which to borrow.

						10_2	10_2	10_2	10_2	10_2			
minuend	X		229		1	1	1	0	0	1	0	1	
subtrahend	Y	–	46	–	0	0	1	0	1	1	1	0	
						1	1	1	1	1			borrows
difference	X – Y		183		1	0	1	1	0	1	1	1	

Example Subtraction (2)

Verify the result:

$$\begin{array}{r} 1\,1\,0\,1\,1\,0 \\ +\quad 1\,0\,1\,1\,1 \\ \hline 1\,0\,0\,1\,1\,0\,1 \end{array}$$

minuend	X	77
subtrahend	Y	– 23
difference	X – Y	54

$$\begin{array}{cccccccc} & & 10_2 & 10_2 & & 10_2 & 10_2 & \\ & 1 & 0 & 0 & 1 & \cancel{1} & \cancel{0} & 1 \\ - & & & 1 & 0 & 1 & 1 & 1 \\ & \boxed{1} & 1 & & \boxed{1} & 1 & & \\ \hline & & 1 & 1 & 0 & 1 & 1 & 0 \end{array}$$

borrows

Signed-Magnitude Representation

Use the MSB for the sign:

n bits:

$$\underbrace{d_{n-1}}_{\text{sign}}\ \underbrace{d_{n-2} \cdots d_0}_{\text{magnitude}}$$

$d_{n-1} = 0$ ← positive number

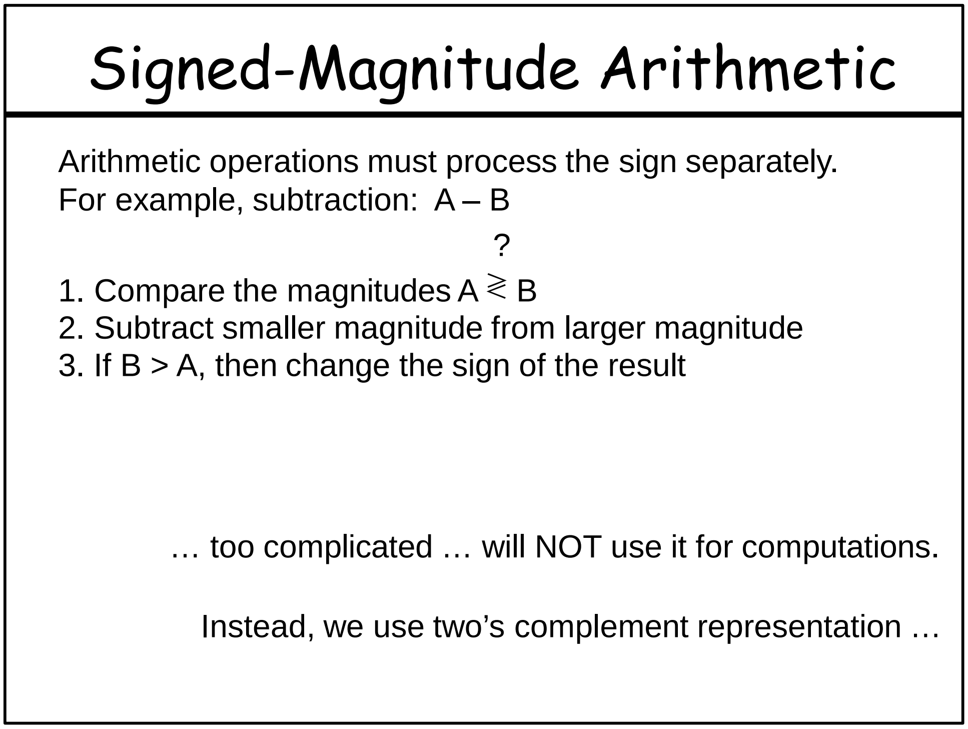
$d_{n-1} = 1$ ← negative number

n bits represent 2^n numbers

largest positive number: $0\overbrace{11\cdots1}^{n-1\text{ bits}}$ $\quad \sum_{i=0}^{n-2} 1\cdot 2^i = 2^{n-1} - 1$

smallest negative number: 111···1 $\quad -(2^{n-1} - 1)$

two representations for zero: 000···0 and 100···0



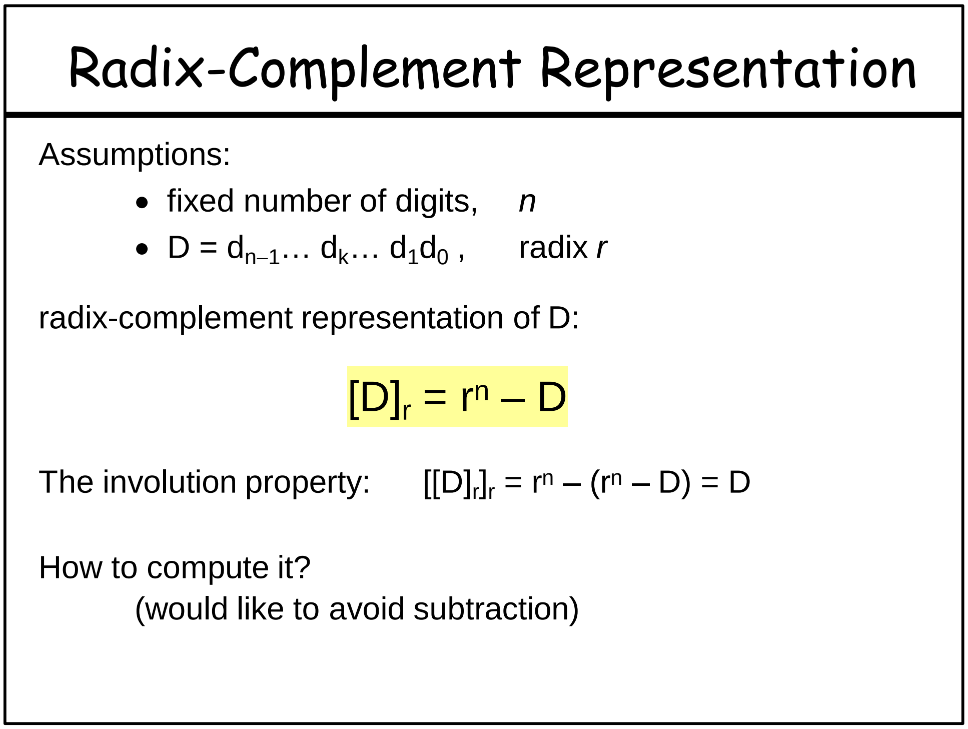
Signed-Magnitude Arithmetic

Arithmetic operations must process the sign separately.
For example, subtraction: A – B

1. Compare the magnitudes $A \overset{?}{\gtrless} B$
2. Subtract smaller magnitude from larger magnitude
3. If B > A, then change the sign of the result

… too complicated … will NOT use it for computations.

Instead, we use two's complement representation …



Radix-Complement Representation

Assumptions:

- fixed number of digits, *n*
- $D = d_{n-1} \ldots d_k \ldots d_1 d_0$, radix *r*

radix-complement representation of D:

$$[D]_r = r^n - D$$

The involution property: $[[D]_r]_r = r^n - (r^n - D) = D$

How to compute it?
(would like to avoid subtraction)

Radix-Complement Computation

$$[D]_r = r^n - D$$

- Rewrite $r^n = (r^n - 1) + 1$

  Then, $[D]_r = r^n - D = ((r^n - 1) - D) + 1$

- Observe that $(r^n - 1)$ has the form $\underbrace{mm \ldots mm}_{n}$
  where $m = r - 1$

  For example, for $r = 10$ and $n = 4$, $(r^n - 1) = 9999$
  for $r = 2$ and $n = 5$, $(r^n - 1) = 11111$

- Define the *complement* of a digit d to be $d'_r = r - 1 - d$

  For example, for $r = 10$, the complements of 3, 5, and 8 are
  $3'_{10} = 10 - 1 - 3 = 6$ $5'_{10} = 10 - 1 - 5 = 4$ $8'_{10} = 10 - 1 - 8 = 1$

- Then, the complement of D is obtained by complementing individual digits of D and adding 1

2's-Complement Representation

n-bit 2's-complement representation of D:

$$[D]_2 = 2^n - D_2$$

Compute two's complement as:

$$[D]_2 = (2^n - 1 - D_2) + 1$$

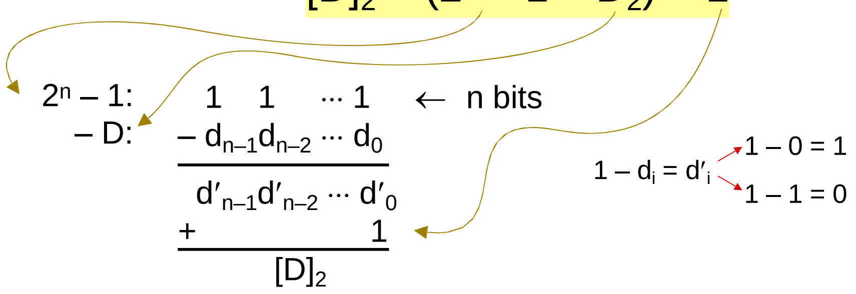
$$\begin{array}{rr} 2^n - 1: & 1\ \ 1\ \ \cdots 1 \\ -\,D: & -\,d_{n-1}d_{n-2}\cdots d_0 \\ \hline & d'_{n-1}d'_{n-2}\cdots d'_0 \\ + & 1 \\ \hline & [D]_2 \end{array} \quad \leftarrow \text{n bits}$$

$$1 - d_i = d'_i \begin{cases} 1 - 0 = 1 \\ 1 - 1 = 0 \end{cases}$$

2's-Complement Computation

1. Complement the digits
2. Add 1 to the Least Significant Bit
3. Discard carry out from Most Significant Bit

$$[D]_2 = (2^n - 1 - D_2) + 1$$



$2^n - 1$: 1 1 ··· 1 ← n bits

– D: $- d_{n-1}d_{n-2} \cdots d_0$

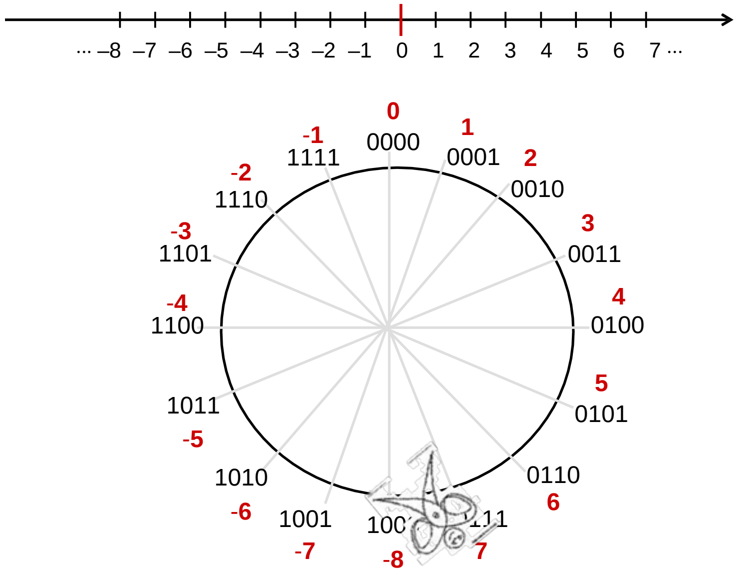
$d'_{n-1}d'_{n-2} \cdots d'_0$

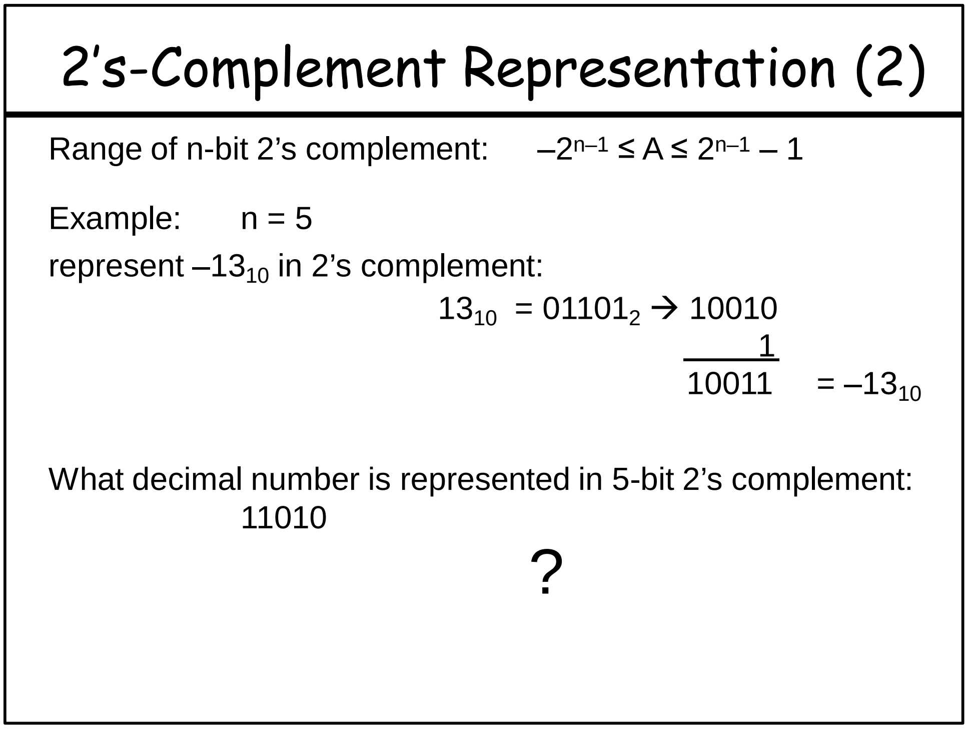
$+ \quad 1$

$[D]_2$

$1 - d_i = d'_i$ → $1 - 0 = 1$, $1 - 1 = 0$

Two's Complement Number System





2's-Complement Representation (2)

Range of n-bit 2's complement: $-2^{n-1} \le A \le 2^{n-1} - 1$

Example: n = 5

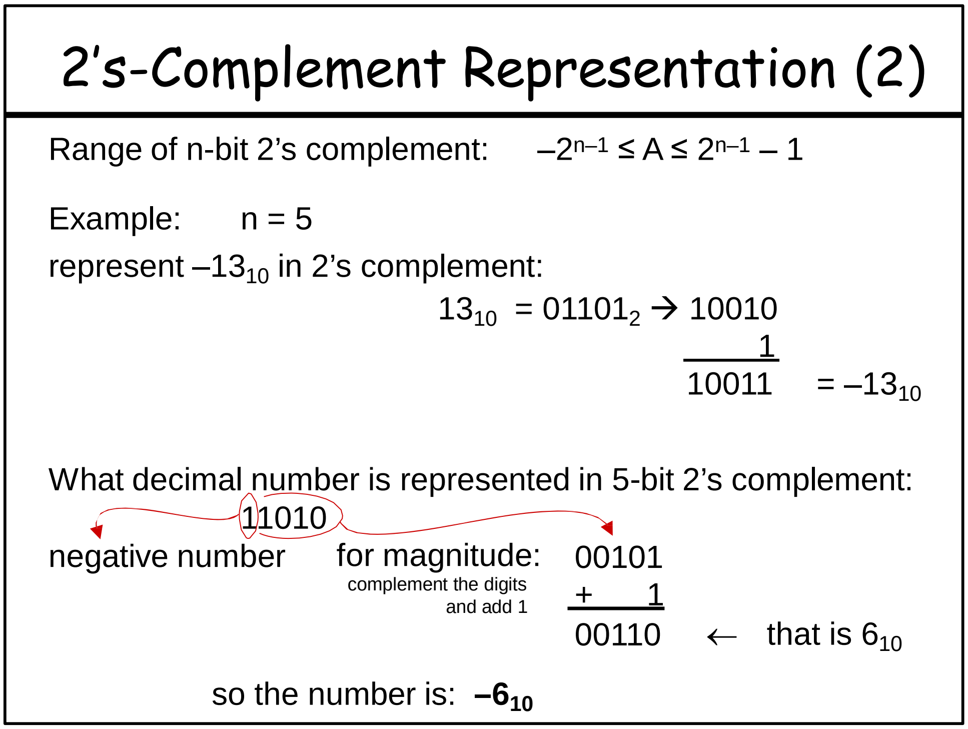
represent -13_{10} in 2's complement:

$$
\begin{array}{rr}
13_{10} = 01101_2 \rightarrow & 10010 \\
 & \underline{1} \\
 & 10011 \quad = -13_{10}
\end{array}
$$

What decimal number is represented in 5-bit 2's complement:

11010

?



2's-Complement Representation (2)

Range of n-bit 2's complement: $-2^{n-1} \le A \le 2^{n-1} - 1$

Example: n = 5

represent -13_{10} in 2's complement:

$$
\begin{array}{rr}
13_{10} = 01101_2 \rightarrow & 10010 \\
 & \underline{1} \\
 & 10011 \quad = -13_{10}
\end{array}
$$

What decimal number is represented in 5-bit 2's complement:

11010

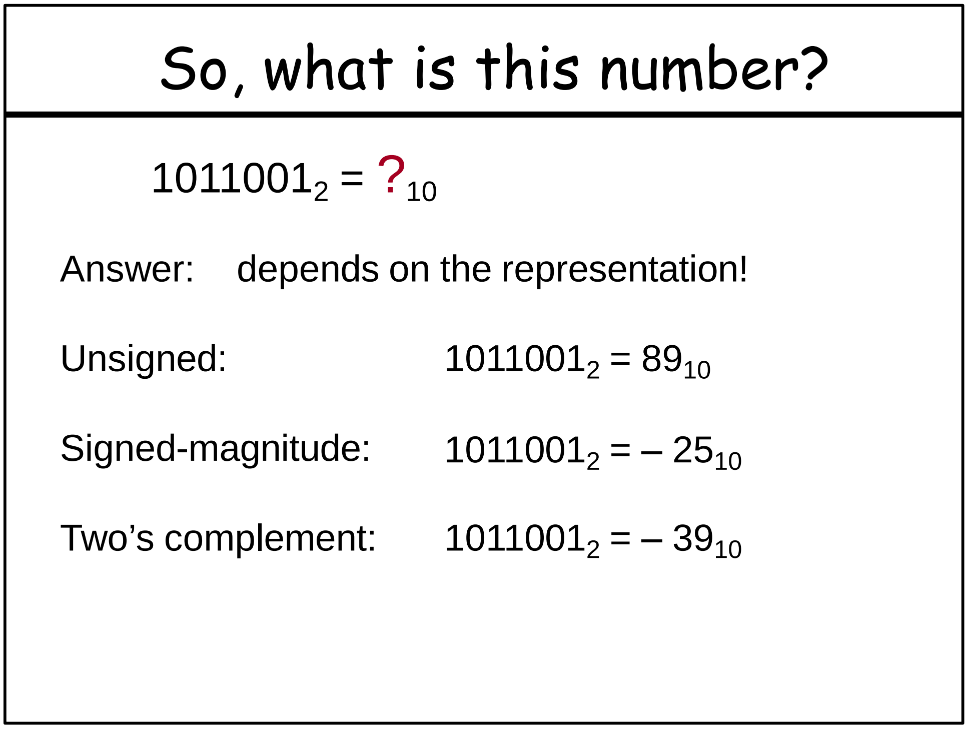
negative number

for magnitude:

complement the digits
and add 1

$$
\begin{array}{rl}
 & 00101 \\
 & \underline{+ \quad 1} \\
 & 00110 \quad \leftarrow \quad \text{that is } 6_{10}
\end{array}
$$

so the number is: $\mathbf{-6_{10}}$



So, what is this number?

$1011001_2 = ?_{10}$

Answer: depends on the representation!

Unsigned:	$1011001_2 = 89_{10}$
Signed-magnitude:	$1011001_2 = -25_{10}$
Two's complement:	$1011001_2 = -39_{10}$