RADIX r NUMBER SYSTEM

Let $(N)_r$ be a radix r number in a *positional weighting* number system, then

$$(N)_r = d_{n-1}r^{n-1} + \cdots + d_0 r^0 \bullet d_{-1}r^{-1} + \cdots + d_{-m}r^{-m}$$

where:

r = radix

d_i = digit at position i, $-m \leq i \leq n-1$

r^i = weight of position i, $-m \leq i \leq n-1$

n = number of integral digits in N

m = number of fractional digits in N

$0, \ldots, r-1$ = digits in radix r number system, that is

$$0 \leq d_i \leq r-1 \text{ where } -m \leq i \leq n-1$$

Computer Number Systems

Binary $r = 2$, digits are $0, 1$. Ex:

$$(101.01)_2 = 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \bullet 0\cdot 2^{-1} + 1\cdot 2^{-2}$$

Octal $r = 8$, digits are $0, 1, 2, 3, 4, 5, 6, 7$. Ex:

$$(62.357)_8 = 6\cdot 8^1 + 2\cdot 8^0 \bullet 3\cdot 8^{-1} + 5\cdot 8^{-2} + 7\cdot 8^{-3}$$

Decimal $r = 10$,
digits are $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$. Ex:

$$(468.93)_{10} = 4\cdot 10^2 + 6\cdot 10^1 + 8\cdot 10^0 \bullet 9\cdot 10^{-1} + 3\cdot 10^{-2}$$

Hexadecimal $r = 16$,
digits are $0, 1, 2, 3, 4, 5, 6, 7, 8, 9$, A,B,C,D,E,F. Ex:

$$(1C.A0F)_{16} = 1\cdot 16^1 + C\cdot 16^0 \bullet A\cdot 16^{-1} + 0\cdot 16^{-2} + F\cdot 16^{-3}$$

Numbers in Different Systems

Binary ($r = 2$)	**Octal** ($r = 8$)	**Decimal** ($r = 10$)	**Hexadecimal** ($r = 16$)
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F
10000	20	16	10
10001	21	17	11
10010	22	18	12
10011	23	19	13
10100	24	20	14

Conversion Algorithm: Polynomial Method

$$(N_1)_a \mapsto (N_2)_b$$

1. Represent $(N_1)_a$ into its positional weighting form

$$d_{n-1}r^{n-1} + \cdots + d_0 r^0 \bullet d_{-1}r^{-1} + \cdots + d_{-m}r^{-m}$$

2. Convert into their equivalent in radix b system

   - The radix a
   - Each digit d_i of $(N_1)_a$
   - Each weight r^i of $(N_1)_a$

3. Apply the positional weighting formula of $(N_1)_a$. The result is $(N_2)_b$

Attention calculations in step 3 are in radix b system

Practical for conversions of the form $(N_1)_a \mapsto (N_2)_{10}$

Example: Conversions to Decimal

$$(N_1)_a \mapsto (N_2)_{10}$$

Binary $\mapsto$ Decimal

$$\begin{aligned}(101.01)_2 &= 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 \bullet 0\cdot 2^{-1} + 1\cdot 2^{-2} \\ &= 1\cdot 4 + 0\cdot 2 + 1\cdot 1 \bullet 0\cdot \tfrac{1}{2} + 1\cdot \tfrac{1}{4} \\ &= 4 + 0 + 1 \bullet 0 + 0.25 = 5 + 0.25 \\ &= (5.25)_{10}\end{aligned}$$

Octal $\mapsto$ Decimal

$$\begin{aligned}(62.357)_8 &= 6\cdot 8^1 + 2\cdot 8^0 \bullet 3\cdot 8^{-1} + 5\cdot 8^{-2} + 7\cdot 8^{-3} \\ &= 6\cdot 8 + 2\cdot 1 \bullet 3\cdot \tfrac{1}{8} + 5\cdot \tfrac{1}{64} + 7\cdot \tfrac{1}{512} \\ &= 48 + 2 \bullet 0.375 + 0.078125 + 0.013671875 \\ &= 50 + 0.466796875 \\ &= (50.466796875)_{10}\end{aligned}$$

Hexadecimal $\mapsto$ Decimal

$$\begin{aligned}(\mathrm{F4C.8A})_{16} &= \mathrm{F}\cdot 16^2 + 4\cdot 16^1 + \mathrm{C}\cdot 16^0 \bullet 8\cdot 16^{-1} + \mathrm{A}\cdot 16^{-2} \\ &= 15\cdot 256 + 4\cdot 16 + 12\cdot 1 \bullet 8\cdot \tfrac{1}{16} + 10\cdot \tfrac{1}{256} \\ &= 3840 + 64 + 12 \bullet 0.5 + 0.0390625 \\ &= 3916 + 0.5390625 \\ &= (3916.5390625)_{10}\end{aligned}$$

Conversion Algorithm: Remainder Method

$(N_1)_a \mapsto (N_2)_b$

1. Convert the integral part of $(N_1)_a$ by successively dividing it by the equivalent of b in radix a system

2. Convert the fractional part of $(N_1)_a$ by successively multiplying it by the equivalent of b in radix a system

3. Convert into their equivalent in radix a system, each remainder obtained in step 1 and each integral part of products obtained in step 2

4. Concatenate all transformed remainders from step 3 by reading up the chain of remainders. The result is the integral part of $(N_2)_b$

5. Concatenate all transformed integral parts from step 3 by reading down the chain of integral parts. The result is the fractional part of $(N_2)_b$

Attention calculations in steps 1 and 2 are in radix a system

Practical for conversions of the form $(N_1)_{10} \mapsto (N_2)_b$

Example: Conversions from Decimal

$$(N_1)_{10} \mapsto (N_2)_b$$

Decimal $\mapsto$ Binary

$$(34.625)_{10} \mapsto (?)_2$$

Integral part			Fractional part	
$34 \div 2 =$	17	0	$0.625 \cdot 2 =$	1.25
$17 \div 2 =$	8	1	$0.25 \cdot 2 =$	0.5
$8 \div 2 =$	4	0	$0.5 \cdot 2 =$	1
$4 \div 2 =$	2	0		
$2 \div 2 =$	1	0		
$1 \div 2 =$	0	1		
		$\uparrow$		$\downarrow$

$$(34.625)_{10} = (100010.101)_2$$

Decimal $\mapsto$ Octal

$$(62.375)_{10} \mapsto (?)_8$$

Integral part			Fractional part	
$62 \div 8 =$	7	6	$0.375 \cdot 8 =$	3.00
$6 \div 8 =$	0	7		
		$\uparrow$		$\downarrow$

$$(62.375)_{10} = (76.3)_8$$

Decimal $\mapsto$ Hexadecimal

$$(37.9)_{10} \mapsto (?)_{16}$$

Integral part			Fractional part	
$37 \div 16 =$	2	5	$0.9 \cdot 16 =$	14.4
$2 \div 16 =$	0	2	$0.4 \cdot 16 =$	6.4
		$\uparrow$		$\downarrow$

$$(37.9)_{10} = (25.E6)_{16}$$

Binary $\leftrightarrow$ Octal/Hexadecimal Conversions

Binary $\mapsto$ Octal Form groups of 3 bits starting at binary point. Each group of 3 bits is an octal digit. Example:

$$\begin{array}{rcccccc} (100010.01)_2 \mapsto & 100 & 010 & \bullet & 010 \\ & 4 & 2 & & 2 \\ = & & (42.2)_8 & & \end{array}$$

Octal $\mapsto$ Binary Convert each octal digit to its equivalent in binary. Example:

$$\begin{array}{rcccccc} (35.6)_8 \mapsto & 011 & 101 & \bullet & 110 \\ = & & (11101.11)_2 & & \end{array}$$

Binary $\mapsto$ Hexadecimal Form groups of 4 bits starting at binary point. Each group of 4 bits is a hexadecimal digit. Example:

$$\begin{array}{rcccccc} (100010.01)_2 \mapsto & 0010 & 0010 & \bullet & 0100 \\ & 2 & 2 & & 4 \\ = & & (22.4)_{16} & & \end{array}$$

Hexadecimal $\mapsto$ Binary Convert each hexadecimal digit to its equivalent in binary. Example:

$$\begin{array}{rcccccc} (5D.A)_{16} \mapsto & 0101 & 1101 & \bullet & 1010 \\ = & & (1011101.101)_2 & & \end{array}$$

BINARY NUMBERS
Fixed Point Representation

N has an *implicit* binary point in a *fixed position*. If N is

Integer binary point is at the right of rightmost digit: $d_{n-1} \ldots d_0$

Fraction binary point is at the left of leftmost digit: $d_{-1} \ldots d_{-m}$

Mixed binary point is between two predetermined digits: $d_{n-1} \ldots d_0 d_{-1} \ldots d_{-m}$

Example 8 bits (5 integral bits, 3 fractional bits)
$(5.25)_{10} = (00101010)_{2(5.3)}$
$(9.625)_{10} = (01001101)_{2(5.3)}$

Notion of complement $\bar{d} = (r-1) - d =$ *complement* of digit d in radix r system. Example, if $r = 9$ then

$\bar{2} = 6,\ \bar{4} = 4,\ \bar{6} = 2$

$\bar{0} = r - 1,\ \bar{1} = r - 2,\ \overline{r-2} = 1,\ \overline{r-1} = 0$

$\bar{r} =$?

Complements of Binary Numbers

1's Complement Form (1CF) Replace each bit d in the number by $\bar{d}$. Example, assume 8-bit integer, then

$$\text{1CF of } (00101010)_2 = (11010101)_2$$
$$\text{1CF of } (10010101)_2 = (01101010)_2$$

2's Complement Form (2CF) Apply 1CF then add $(1)_2$ to the result. Example, assume 8-bit integer, then

$$\text{2CF of } (00101010)_2 = (11010110)_2$$
$$\text{2CF of } (10010101)_2 = (01101011)_2$$

2CF of 2CF of $(1001010)_2$ =?

What is the rule ?

Unsigned Binary integers (UBI) Unsigned n-bit integer N ($0 \leq N \leq 2^n - 1$) is represented in terms of positional weighting: $d_{n-1} \ldots d_0$

Signed Binary Integers (SBI)

For each representation of $\pm N$

- $d_{n-1} = \begin{cases} 0 & \text{if } N \geq 0 \\ 1 & \text{if } N < 0 \end{cases}$ = sign of $\pm N$
- $d_{n-2} \ldots d_0$ represents $|N|$
- ? $\leq |N| \leq$?

If $N \geq 0$, $d_{n-2} \ldots d_0 = |N|$

If $N < 0$, meaning of $d_{n-2} \ldots d_0$ depends on the sign form

Sign and Magnitude Form (SMF)

$d_{n-2} \ldots d_0$ is $|N|$

1's Complement Form (1CF)

$d_{n-2} \ldots d_0$ is 1CF of $|N|$

2's Complement Form (2CF)

$d_{n-2} \ldots d_0$ is 2CF of $|N|$

Comparison of UBI, SMF, 2CF and 1CF

(4-bit integers)

Binary	1CF	2CF	SMF	UBI
0111	+7	+7	+7	7
0110	+6	+6	+6	6
0101	+5	+5	+5	5
0100	+4	+4	+4	4
0011	+3	+3	+3	3
0010	+2	+2	+2	2
0001	+1	+1	+1	1
0000	+0	+0	+0	0
1111	−0	−1	−7	15
1110	−1	−2	−6	14
1101	−2	−3	−5	13
1100	−3	−4	−4	12
1011	−4	−5	−3	11
1010	−5	−6	−2	10
1001	−6	−7	−1	9
1000	−7	−8	−0	8

Leftmost bit is sign bit and represents no quantity

In 1CF and SMF there are 16 integers:

7 negative, 2 zeros (+0, −0), 7 positive

In 2CF there are 16 integers:

8 negative, 1 zero (+0), 7 positive

How Many n-bit Integers are Represented?

In 1CF and SMF, the number of

- Negative integers = ?
- Zeros = ?
- Positive integers = ?

In 2CF, the number of

- Negative integers = ?
- Zeros = ?
- Positive integers = ?

There are 2^n unique numbers in each representation

Signed Binary Addition/Substraction

sum = augend + addend

difference = minuend − substrahend

Addition in 1CF operands and result are in 1CF

1. Add augend and addend
2. Add carry-out (if any) to result

Addition in 2CF operands and result are in 2CF

1. Add augend and addend
2. Discard carry-out (if any)

Addition in SMF operands and result are in SMF

1. If augend and addend have same sign then add their magnitudes and result takes sign of operands
2. If augend and addend have distinct signs then substract smaller magnitude from larger magnitude and result takes sign of larger one.

Substraction is performed by addition, i.e. **difference = minuend + (− substrahend)**, where **(− substrahend) is 1CF/2CF of substrahend**

Overflow

Occurs when both operands have the same sign but the result of addition/substraction has different sign than both operands

Consider n-bit integers and let $s = x \pm y$

- x, y **positive** ($x_{n-1} = y_{n-1} = 0$) then overflow if $s_{n-1} = 1$
- x, y **negative** ($x_{n-1} = y_{n-1} = 1$) then overflow if $s_{n-1} = 0$

Result of addition/substraction is correct only if overflow does not occur, that is if

$$-2^{n-1} \leq x + y \leq 2^{n-1} - 1$$

in case of addition or

$$-2^{n-1} \leq x + (\text{1CF/2CF/SMF of } y) \leq 2^{n-1} - 1$$

in case of substraction

There cannot be overflow if x and y have distinct signs ($x_{n-1} \neq y_{n-1}$). Because $|s| \leq |x|$ or $|s| \leq |y|$