Positional Number Systems

- A number system consists of an order set of symbols (digits) with relations defined for +,-,*, /
- The radix (or base) of the number system is the total number of digits allowed in the the number system.
  - Example, for the decimal number system:
    - Radix, r = 10, Digits allowed = 0,1, 2, 3, 4, 5, 6, 7, 8, 9
- In positional number systems, a number is represented by a string of digits, where each digit position has an associated weight.
- The value of a number is the weighted sum of the digits.
- The general representation of an unsigned number D with whole and fraction portions number in a number system with radix r:

$$D_r = d_{p-1}\, d_{p-2} \ldots d_1\, d_0 . d_{-1}\, d_{-2} \ldots D_{-n}$$

- The number above has p digits to the left of the radix point and n fraction digits to the right.
- A digit in position i has as associated weight r^i
- The value of the number is the sum of the digits multiplied by the associated weight r^i :

$$D = \sum_{i=-n}^{p-1} d_i \times r^i$$

Positional Number Systems

Number: $D_r = d_{p-1}\, d_{p-2} \ldots d_1\, d_0 . d_{-1}\, d_{-2} \ldots D_{-n}$

Value: $$D = \sum_{i=-n}^{p-1} d_i \times r^i$$

- **For example in the decimal number system:**

$$5185.68_{10} = 5\text{x}10^3 + 1\text{x}10^2 + 8\text{x}10^1 + 5\text{x}10^0 + 6 \text{ x } 10^{-1} + 8 \text{ x } 10^{-2}$$

$$= 5\text{x}1000 + 1\text{x}100 + 8\text{x}10 + 5 \text{ x } 1 + 6\text{x}.1 + 8\text{x}.01$$

- **For the binary number system with radix = 2, digits 0, 1**

$$D_2 = d_{p-1} \times 2^{p-1} \ldots d_1 \times 2^1 + d_0 . 2^0 + d_{-1} \times 2^{-1} + d_{-2} \times 2^{-2} \ldots$$

- **For Example:**

$$10011_2 = 1 \times 16 + 0 \times 8 + 0 \times 4 + 1 \times 2 + 1 \times 1 = 19_{10}$$

MSB (most significant bit) — LSB (least significant bit)

$$101.001_2 = 1\text{x}4 + 0\text{x}2 + 1\text{x}1 + 0\text{x}.5 + 0\text{x}.25 + 1\text{x}.125 = 5.125_{10}$$

Binary Point

Number Systems Used in Computers

Name of Radix	Radix	Set of Digits	Example
Decimal	r=10	{0,1,2,3,4,5,6,7,8,9}	255_{10}
Binary	r=2	{0,1}	11111111_2
Octal	r= 8	{0,1,2,3,4,5,6,7}	377_8
Hexadecimal	r=16	{0,1,2,3,4,5,6,7,8,9,A, B, C, D, E, F}	FF_{16}

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Radix-r to Decimal Conversion

- **The decimal value of a number in any radix r is found by converting each digit to its radix 10 equivalent and expanding the value using radix arithmetic:**

$$D = \sum_{i=-n}^{p-1} d_i \times r^i$$

- **Examples:**

$1101.101_2 = 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-3}$
$= 8 + 4 + 1 + 0.5 + 0.125 = 13.625_{10}$

$572.6_8 = 5 \times 8^2 + 7 \times 8^1 + 2 \times 8^0 + 6 \times 8^{-1}$
$= 320 + 56 + 16 + 0.75 = 392.75_{10}$

$2A.8_{16} = 2 \times 16^1 + 10 \times 16^0 + 8 \times 16^{-1}$
$= 32 + 10 + 0.5 = 42.5_{10}$

$132.3_4 = 1 \times 4^2 + 3 \times 4^1 + 2 \times 4^0 + 3 \times 4^{-1}$
$= 16 + 12 + 2 + 0.75 = 30.75_{10}$

$341.24_5 = 3 \times 5^2 + 4 \times 5^1 + 1 \times 5^0 + 2 \times 5^{-1} + 4 \times 5^{-2}$
$= 75 + 20 + 1 + 0.4 + 0.16 = 96.56_{10}$

Decimal-to-Binary Conversion

- **Separate the decimal number into whole and fraction portions.**
- **To convert the whole number portion to binary, use successive division by 2 until the quotient is 0. The remainders form the answer, with the first remainder as the *least significant bit (LSB)* and the last as the *most significant bit (MSB)*.**
- **Example: Convert 179_{10} to binary:**

179 / 2 = 89 remainder 1 (LSB)
/ 2 = 44 remainder 1
/ 2 = 22 remainder 0
/ 2 = 11 remainder 0
/ 2 = 5 remainder 1
/ 2 = 2 remainder 1
/ 2 = 1 remainder 0
/ 2 = 0 remainder 1 (MSB)

$$179_{10} = 10110011_2$$

Decimal-to-Binary Conversion

- **To convert decimal fractions to binary, repeated multiplication by 2 is used, until the fractional product is 0 (or until the desired number of binary places). The whole digits of the multiplication results produce the answer, with the first as the MSB, and the last as the LSB.**
- **Example: Convert 0.3125_{10} to binary**

			Result Digit	
.3125 ´ 2	=	0.625	0	(MSB)
.625 ´ 2	=	1.25	1	
.25 ´ 2	=	0.50	0	
.5 ´ 2	=	1.0	1	(LSB)

$$0.3125_{10} = .0101_2$$

Decimal-to-Binary Conversion

Sum-of-Weights Method

- **Determine the set of binary weights whose sum is equal to the decimal number.**

Examples:

$9_{10} = 8 + 1 = 2^3 + 2^0 = 1001_2$

$18_{10} = 16 + 2 = 2^4 + 2^1 = 10010_2$

$58_{10} = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 = 111010_2$

$0.625_{10} = 0.5 + 0.125 = 2^{-1} + 2^{-3} = 0.101_2$

Decimal to Radix-r Conversion

- **Separate the decimal number into whole and fraction portions.**
- **To convert the whole number portion to binary, use successive division by r until the quotient is 0. The remainders form the answer, with the first remainder as the *least significant digit (LSD)* and the last as the *most significant digit (MSD).***
- **To convert decimal fractions to radix-r, repeated multiplication by r is used, until the fractional product is 0 (or until the desired number of binary places). The whole digits of the multiplication results produce the answer, with the first as the MSD, and the last as the LSD.**
- **Example: Convert 467_{10} to octal**

467 / 8 = 58 remainder 3 (LSD)

/ 8 = 7 remainder 2

/ 8 = 0 remainder 7 (MSD)

$467_{10} = 723_8$

Binary to Octal Conversion

- **Separate the whole binary number portion into groups of 3 beginning at the binary point and working to the left. Add leading zeroes as necessary.**
- **Separate the fraction binary number portion into groups of 3 beginning at the binary point and working to the right. Add trailing zeroes as necessary.**
- **Convert each group of 3 to the equivalent octal digit.**
- **Example:**

$$3564.875_{10} = 110\ 111\ 101\ 100.111_2$$
$$= (6 \times 8^3) + (7 \times 8^2) + (5 \times 8^1) + (4 \times 8^0) + (7 \times 8^{-1})$$
$$= 6754.7_8$$

Binary to Hexadecimal Conversion

- **Separate the whole binary number portion into groups of 4 beginning at the binary point and working to the left. Add leading zeroes as necessary.**
- **Separate the fraction binary number portion into groups of 4 beginning at the binary point and working to the right. Add trailing zeroes as necessary.**
- **Convert each group of 4 to the equivalent hexadecimal digit.**
- **Example:**

$$3564.875_{10} = 1101\ 1110\ 1100.1110_2$$
$$= (D \times 16^2) + (E \times 16^1) + (C \times 16^0) + (E \times 16^{-1})$$
$$= DEC.E_{16}$$

Conversion between Number Systems Summary

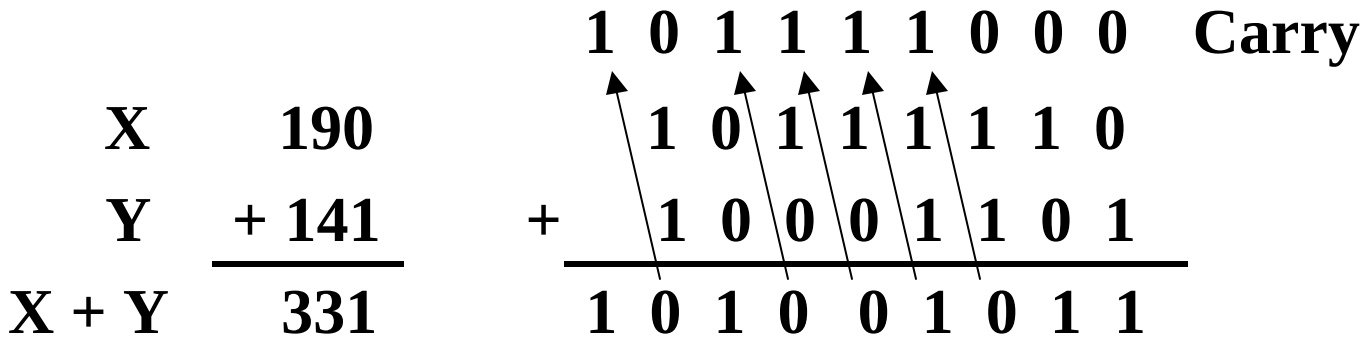
- **Radix-r to decimal:**
  - **Multiply digits with their corresponding weights and add**

$$D = \sum_{i=-n}^{p-1} d_i \times r^i$$

- **Decimal to binary (radix 2)**
  - **Whole numbers: repeated division by 2**
  - **Fractions: repeated multiplication by 2**
- **Decimal to radix-r**
  - **Whole numbers: repeated division by r**
  - **Fractions: repeated multiplication by r**
- **Binary to Octal**
  - **Substitute groups of three bits with corresponding octal digit.**
- **Binary to Hexadecimal**
  - **Substitute groups of four bits with corresponding hexadecimal digit.**

Binary Arithmetic Operations
Addition

- **Similar to decimal number addition, two binary numbers are added by adding each pair of bits together with carry propagation.**
- **Addition Example:**



Binary Arithmetic Operations
Subtraction

- **Two binary numbers are subtracted by subtracting each pair of bits together with borrowing, where needed.**
- **Subtraction Example:**

		0 0 1 1 1 1 1 0 0	Borrow
X	229	1 1 1 0 0 1 0 1	
Y -	46	- 0 0 1 0 1 1 1 0	
	183	1 0 1 1 0 1 1 1	

Negative Binary Number Representations

- **Signed-Magnitude Representation:**
  - **For an *n-bit* binary number:**

    Use the first bit (most significant bit, MSB) position to represent the sign where 0 is positive and 1 is negative.

    Ex. $1\,1\,1\,1\,1\,1\,1\,1_2 = -127_{10}$

    Sign **Magnitude**
  - **Remaining n-1 bits represent the magnitude which may range from:**

    $$-2^{(n-1)} + 1 \quad \text{to} \quad 2^{(n-1)} - 1$$
  - **This scheme has two representations for 0; i.e., both positive and negative 0: for 8 bits: 00000000, 10000000**
  - **Arithmetic under this scheme uses the sign bit to indicate the nature of the operation and the sign of the result, but the sign bit is not used as part of the arithmetic.**

Negative Binary Number Representations

- **Two's complement representation:**
- **MSB is the sign (MSB = 1 indicates a negative number)**
- **To negate a number complement all bits and add 1**
- **ex. 119_{10} = 01110111 complement bits**

  10001000

  +1 add 1

  10001001_2 = - 119_{10}

Properties of Two's Complement Numbers

- **X plus the complement of X equals 0.**
- **There is one unique 0.**
- **Positive numbers have 0 as their leading bit (MSB); while negatives have 1 as their MSB.**
- **The range for an *n-bit* binary number in 2's complement representation is:**

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

- **The complement of the complement of a number is the original number.**
- **Subtraction is done by *addition* to the 2's complement of the number.**

Value of Two's Complement Numbers

- **For an n-bit 2's complement number the weights of the bits is the same as for unsigned numbers except of the MSB or sign bit where the weight is -2^{n-1}, thus the value of the n-bit 2's complement number is given by:**

$$D_{\text{2's-complement}} = d_{n-1} \times -2^{n-1} + d_{n-2} \times 2^{n-2} \ldots\ldots d_1 \times 2^1 + d_0$$

For example:

the value of the 4-bit 2's complement number 1011 is given by:

$$\text{value} = d_3 \times -2^3 + d_2 \times 2^2 + d_1 \times 2^1 + d_0$$

$$= 1 \times -2^3 + 0 \times 2^2 + 1 \times 2^1 + 1$$

$$= -8 + 0 + 2 + 1$$

$$= -8 + 3 = -5$$

Extending Two's Complement Numbers: Sign Extension

- **An n-bit 2's complement number can converted to an m-bit number where m>n by appending m-n copies of the sign bit to the left of the number. This process is called sign extension.**

- **Example: To convert the 4-bit 2's complement number 1011 to an 8-bit representation, the sign bit (here = 1) must be extended by appending four 1's to left of the number:**

$$1011_{\text{4-bit 2's-complement}} = 11111011_{\text{8-bit 2's-complement}}$$

To verify that the value of the 8-bit number is still -5

$$\text{value of 8-bit number} = -2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2 + 1$$
$$= -128 + 64 + 32 + 16 + 8 + 2 + 1$$
$$= -128 + 123 = -5$$

Two' complement addition/subtraction

Examples:

4	0100		-2	1110	
+ -7	1001	+	-6	1010	
-3	1101		-8	1 1000	

Ignore carry out from MSB

- Overflow occurs if signs (MSBs) of both operands are the same and the sign of the result is different.
- Overflow can also be detected if the carry in the sign position is different from the carry out of the sign position.

Negative Binary Number Representations

- **One's-Complement representation**
- **MSB is the sign (MSB = 1 indicates a negative number)**
- **Negative numbers are found by complementing all bits**
- **ex. 119_{10} = 01110111**
  -119_{10} = 10001000
- **The range of values for an *n-bit* binary number in 1's complement representation is:**
- **from $-2^{(n-1)}+1$ to $2^{(n-1)}-1$**
- **One's-complement addition/subtraction:**

If there is a carry out of the sign position add 1

Ex.

	-2	1101
+	-5	1010
	-7	10111
		+ 1
		1000

Value of One's Complement Numbers

- **For an n-bit 2's complement number the weights of the bits is also the same as for unsigned numbers except of the MSB or sign bit where the weight is $-(2^{n-1}+1)$, thus the value of the n-bit 1's complement number is given by:**

$$D_{\text{1's-complement}} = d_{n-1} \times -(2^{n-1}+1) + d_{n-2} \times 2^{n-2} \ldots\ldots d_1 \times 2^1 + d_0$$

For example:

the value of the 4-bit 1's complement number 1011 is given by:

$$\text{value} = d_3 \times -(2^3+1) + d_2 \times 2^2 + d_1 \times 2^1 + d_0$$

$$= 1 \times -(2^3+1) + 0 \times 2^2 + 1 \times 2^1 + 1$$

$$= -7 + 0 + 2 + 1$$

$$= -7 + 3 = -4$$

Comparison of Signed-Magnitude & Complements

Example: 4-bit signed number (positive values)

Value	Sign-and-Magnitude	1s Comp.	2s Comp.
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000

Comparison of Signed-Magnitude & Complements

Example: 4-bit signed number (negative values)

Value	Sign-and-Magnitude	1s Comp.	2s Comp.
-0	1000	1111	-
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	-	-	1000

MSB = 1 indicates a negative number in either notation