CHAPTER V

NUMBER SYSTEMS AND ARITHMETIC

NUMBER SYSTEMS

RADIX-R REPRESENTATION

- Decimal number expansion

$$\mathbf{73625}_{10} = (\mathbf{7} \times \mathbf{10}^{\mathbf{4}}) + (\mathbf{3} \times \mathbf{10}^{\mathbf{3}}) + (\mathbf{6} \times \mathbf{10}^{\mathbf{2}}) + (\mathbf{2} \times \mathbf{10}^{\mathbf{1}}) + (\mathbf{5} \times \mathbf{10}^{\mathbf{0}})$$

- Binary number representation

$$\mathbf{10110}_{2} = (\mathbf{1} \times \mathbf{2}^{\mathbf{4}}) + (\mathbf{0} \times \mathbf{2}^{\mathbf{3}}) + (\mathbf{1} \times \mathbf{2}^{\mathbf{2}}) + (\mathbf{1} \times \mathbf{2}^{\mathbf{1}}) + (\mathbf{0} \times \mathbf{2}^{\mathbf{0}}) = \mathbf{22}_{10}$$

- Hexadecimal number representation

$$\mathbf{3E4B8}_{16} = (\mathbf{3} \times \mathbf{16}^{\mathbf{4}}) + (\mathbf{14} \times \mathbf{16}^{\mathbf{3}}) + (\mathbf{4} \times \mathbf{16}^{\mathbf{2}}) + (\mathbf{11} \times \mathbf{16}^{\mathbf{1}}) + (\mathbf{8} \times \mathbf{16}^{\mathbf{0}})$$
$$= \mathbf{255160}_{10}$$

NUMBER SYSTEMS

DECIMAL REPRESENTATION

Radix-10 Representation

$$73625.4385_{10}$$

	10^5	10^4	10^3	10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	
...	0	7	3	6	2	5	.	4	3	8	5	0	...

$$73625.4385_{10} = (7 \times 10^4) + (3 \times 10^3) + (6 \times 10^2) + (2 \times 10^1) + (5 \times 10^0)$$
$$+ (4 \times 10^{-1}) + (3 \times 10^{-2}) + (8 \times 10^{-3}) + (5 \times 10^{-4})$$

NUMBER SYSTEMS

BINARY REPRESENTATION

Radix-2 Representation

$\mathbf{10110.0011}_2$

	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	
···	0	1	0	1	1	0	.	0	0	1	1	0	···
	MSB											LSB	

$$\mathbf{10110.0011}_2 = (1 \times 2^4) + (0 \times 2^3) + (1 \times 2^2) + (1 \times 2^1) + (0 \times 2^0)$$
$$+ (0 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4})$$
$$= \mathbf{22.1875}_{10}$$

NUMBER SYSTEMS

OCTAL REPRESENTATION

Radix-8 Representation

$$\mathbf{26516.1731}_8$$

	8^5	8^4	8^3	8^2	8^1	8^0		8^{-1}	8^{-2}	8^{-3}	8^{-4}	8^{-5}	
···	0	2	6	5	1	6	.	1	7	3	1	0	···

$$\mathbf{26516.1731}_8 = (\mathbf{2} \times \mathbf{8^4}) + (\mathbf{6} \times \mathbf{8^3}) + (\mathbf{5} \times \mathbf{8^2}) + (\mathbf{1} \times \mathbf{8^1}) + (\mathbf{6} \times \mathbf{8^0})$$

$$+ (\mathbf{1} \times \mathbf{8^{-1}}) + (\mathbf{7} \times \mathbf{8^{-2}}) + (\mathbf{3} \times \mathbf{8^{-3}}) + (\mathbf{1} \times \mathbf{8^{-4}})$$

$$= \mathbf{11598.24}_{10}$$

NUMBER SYSTEMS

HEXADECIMAL REPRES.

Radix-16 Representation

$\mathbf{19AD6.F411}_{16}$

	16^5	16^4	16^3	16^2	16^1	16^0		16^{-1}	16^{-2}	16^{-3}	16^{-4}	16^{-5}	
...	0	1	9	A	D	6	.	F	4	1	1	0	...

$$\mathbf{19AD6.F411}_{16} = (\mathbf{1} \times \mathbf{16^4}) + (\mathbf{9} \times \mathbf{16^3}) + (\mathbf{A} \times \mathbf{16^2}) + (\mathbf{D} \times \mathbf{16^1}) + (\mathbf{6} \times \mathbf{16^0})$$

$$+ (\mathbf{F} \times \mathbf{16^{-1}}) + (\mathbf{4} \times \mathbf{16^{-2}}) + (\mathbf{1} \times \mathbf{16^{-3}}) + (\mathbf{1} \times \mathbf{16^{-4}})$$

$$\approx \mathbf{105174.95}_{10}$$

NUMBER SYSTEMS

BINARY <-> HEXADECIMAL

BINARY <-> HEXADECIMAL

$\mathbf{0000_2 = 0_{16}}$	$\mathbf{1000_2 = 8_{16}}$
$\mathbf{0001_2 = 1_{16}}$	$\mathbf{1001_2 = 9_{16}}$
$\mathbf{0010_2 = 2_{16}}$	$\mathbf{1010_2 = 10\ (A_{16})}$
$\mathbf{0011_2 = 3_{16}}$	$\mathbf{1011_2 = 11\ (B_{16})}$
$\mathbf{0100_2 = 4_{16}}$	$\mathbf{1100_2 = 12\ (C_{16})}$
$\mathbf{0101_2 = 5_{16}}$	$\mathbf{1101_2 = 13\ (D_{16})}$
$\mathbf{0110_2 = 6_{16}}$	$\mathbf{1110_2 = 14\ (E_{16})}$
$\mathbf{0111_2 = 7_{16}}$	$\mathbf{1111_2 = 15\ (F_{16})}$

BINARY -> HEXADECIMAL

Group binary by 4 bits from radix point

Examples:

$$\mathbf{\underbrace{0111}_{7}\ \underbrace{1011}_{B}{}_2 = 7B_{16}}$$

$$\mathbf{\underbrace{10}_{2}\ \underbrace{1010}_{A}\ \underbrace{0110}_{6}.\underbrace{1100}_{C}\ \underbrace{01}_{4}{}_2 = 2A6.C4_{16}}$$

BINARY <-> OCTAL

$\mathbf{000}_2 = \mathbf{0}_8$

$\mathbf{001}_2 = \mathbf{1}_8$

$\mathbf{010}_2 = \mathbf{2}_8$

$\mathbf{011}_2 = \mathbf{3}_8$

$\mathbf{100}_2 = \mathbf{4}_8$

$\mathbf{101}_2 = \mathbf{5}_8$

$\mathbf{110}_2 = \mathbf{6}_8$

$\mathbf{111}_2 = \mathbf{7}_8$

BINARY -> OCTAL

Group binary bits by 3 from LSB

Examples:

$$\underbrace{10}_{2}\ \underbrace{100}_{4}\ \underbrace{110}_{6}{}_2 = 246_8$$

$$\underbrace{10}_{2}\ \underbrace{101}_{5}\ \underbrace{111}_{7}\ \underbrace{011}_{3}.\underbrace{011}_{3}\ \underbrace{11}_{6}{}_2 = 2573.36_8$$

NUMBER SYSTEMS

BINARY -> DECIMAL

- Perform radix-2 expansion
  - Multiply each bit in the binary number by 2 to the power of its place. Then sum all of the values to get the decimal value.

Examples:

$$\mathbf{10111}_2 = (\mathbf{1}\times\mathbf{2}^{\mathbf{4}}) + (\mathbf{0}\times\mathbf{2}^{\mathbf{3}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{2}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{1}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{0}}) = \mathbf{23}_{10}$$

$$\mathbf{10110.0011}_2 = (\mathbf{1}\times\mathbf{2}^{\mathbf{4}}) + (\mathbf{0}\times\mathbf{2}^{\mathbf{3}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{2}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{1}}) + (\mathbf{0}\times\mathbf{2}^{\mathbf{0}})$$

$$+ (\mathbf{0}\times\mathbf{2}^{\mathbf{-1}}) + (\mathbf{0}\times\mathbf{2}^{\mathbf{-2}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{-3}}) + (\mathbf{1}\times\mathbf{2}^{\mathbf{-4}})$$

$$= \mathbf{22.1875}_{10}$$

NUMBER SYSTEMS

DECIMAL -> BINARY

- **Integer part:**
  - Modulo division of decimal integer by 2 to get each bit, starting with LSB.
- **Fraction part:**
  - Multiplication decimal fraction by 2 and collect resulting integers, starting with MSB.

Example: Convert $\mathbf{41.828125_{10}}$

$$
\begin{array}{rl}
\mathbf{41} \bmod \mathbf{2} = \mathbf{1} & \text{LSB} \\
\mathbf{20} \bmod \mathbf{2} = \mathbf{0} & \downarrow \\
\mathbf{10} \bmod \mathbf{2} = \mathbf{0} & \\
\mathbf{5} \bmod \mathbf{2} = \mathbf{1} & \\
\mathbf{2} \bmod \mathbf{2} = \mathbf{0} & \\
\mathbf{1} \bmod \mathbf{2} = \mathbf{1} & \text{MSB}
\end{array}
$$

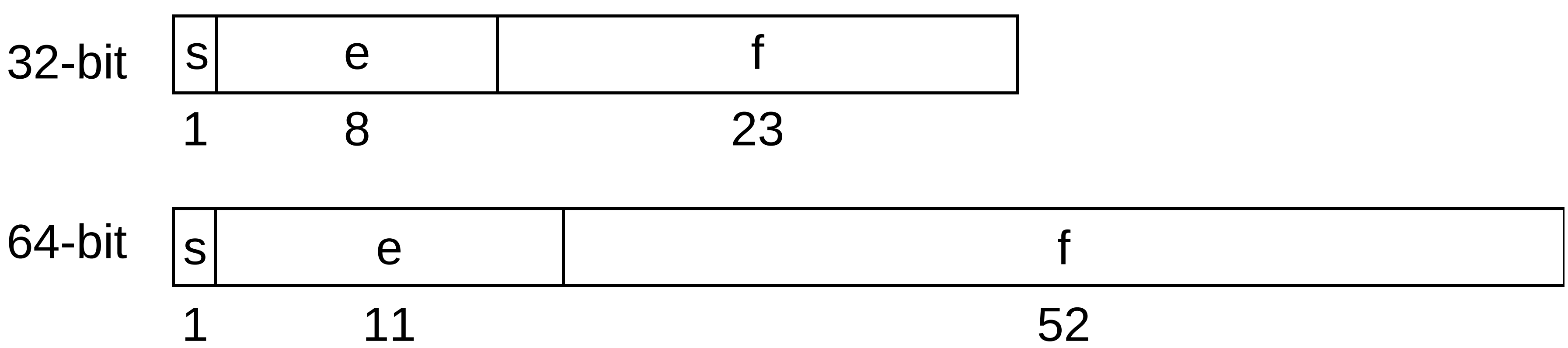
$$
\begin{array}{rl}
\mathbf{0.828125} \times \mathbf{2} = \mathbf{1.65625} & \text{MSB} \\
\mathbf{0.65625} \times \mathbf{2} = \mathbf{1.3125} & \downarrow \\
\mathbf{0.3125} \times \mathbf{2} = \mathbf{0.625} & \\
\mathbf{0.625} \times \mathbf{2} = \mathbf{1.25} & \\
\mathbf{0.25} \times \mathbf{2} = \mathbf{0.5} & \\
\mathbf{0.5} \times \mathbf{2} = \mathbf{1.0} & \text{LSB}
\end{array}
$$

Therefore $\mathbf{41.828125_{10} = 101001.110101_2}$

NUMBER SYSTEMS

FLOATING POINT NUMBERS

- Floating point numbers can be represented with a sign bit, a fraction (often referred to as the mantissa), and an exponent.
  - Example 1: $-\mathbf{267.426} = -\mathbf{0.267426} \times \mathbf{10^3}$, where the sign is negative, the fraction is $\mathbf{0.267426}$ and the exponent is $\mathbf{3}$.
  - Example 2: $\mathbf{0101011.1001} = \mathbf{0.1010111} \times \mathbf{2^6}$, where the sign is positive, the fraction is $\mathbf{0.1010111}$, and the exponent is $\mathbf{0110}$.
- Sample IEEE Floating-Point Formats



BINARY NUMBERS

UNSIGNED INTEGER

- The range for an *n*-bit radix-*r* unsigned integer is

$$[0, r_{10}^{n} - 1]$$

- Example: For a 16-bit binary unsigned integer, the range is

$$[0, 2^{16} - 1] = [0, 65535]$$

which has a binary representation of

0000 0000 0000 0000 = 0

0000 0000 0000 0001 = 1

0000 0000 0000 0010 = 2

. . .

1111 1111 1111 1110 = 65534

1111 1111 1111 1111 = 65535

BINARY NUMBERS

SIGNED INTEGERS (1)

- The range for an *n*-bit radix-*r* signed integer is

$$[-r_{10}^{n-1}, r_{10}^{n-1} - 1]$$

- The most-significant bit is used as a sign bit, where **0** indicates a positive integer and **1** indicates a negative integer.

Example: For a 16-bit binary signed integer, the range is

$$[-2^{16-1}, 2^{16-1} - 1] = [-32768, 32767]$$

BINARY NUMBERS

SIGNED INTEGERS (2)

- There are a number of different representations for signed integers, each which has its own advantage
  - Signed-magnitude representation:
    - **1010 0001 0110 1111**
  - Signed-1's complement representation:
    - **1101 1110 1001 0000**
  - Signed-2's complement representation:
    - **1101 1110 1001 0001**
- The above examples are all the same number, $-\mathbf{8559_{10}}$.

BINARY NUMBERS

SIGNED-MAGNITUDE

- The **signed-magnitude** binary integer representation is just like the **unsigned representation** with the addition of a **sign bit**.
  - For instance, using 8-bits, the number $-\mathbf{6}_{10}$ can be represented as the 7-bit magnitude of $\mathbf{6}_{10}$ using

    000 0110

    and then the sign bit appended to the MSB to form

    1000 0110

BINARY NUMBERS

RADIX COMPLEMENTS

- The **radix complement**, or **r's complement**, of an integer representation for an *n*-digit integer is defined as

$$r^n{}_{10} - \text{number}_{10}$$

- The **diminished radix complement**, or **(r - 1)'s complement**, of an integer representation for an *n*-digit integer is defined as

$$(r^n{}_{10} - 1_{10}) - \text{number}_{10}$$

- Example: Find the r's and (r - 1)'s complement for $\mathbf{3764}_{10}$

r's complement

$$\mathbf{10^5 - 3764 = 96236}$$

(r - 1)'s complement

$$\mathbf{(10^5 - 1) - 3764 = 96235}$$

BINARY NUMBERS

1'S COMPLEMENT

- The **1's complement** (diminished radix complement) binary integer representation for an n-bit integer is defined as

$$(2^n_{10} - 1_{10}) - \text{number}_{10}$$

- In essence, this takes the positive version of the number and flips all of the bits.
  - For instance, using 8-bits, the number $-\mathbf{6}_{10}$ can be represented as the 8-bit positive number $\mathbf{6}_{10}$ using

    0000 0110

    and then each of the bits flipped to form the 1's complement

    1111 1001

BINARY NUMBERS

2'S COMPLEMENT

- The **2's complement** (radix complement) binary integer representation for an n-bit integer is defined as

$$2^n_{10} - \text{number}_{10}$$

- In essence, this takes the 1's complement and adds one.
  - For instance, using 8-bits, the number $-\mathbf{6}_{10}$ can be represented as the 8-bit positive number $\mathbf{6}_{10}$ using

    0000 0110

    and then each of the bits flipped to form the 1's complement

    1111 1001

    and then add **1** to form the 2's complement

    1111 1010

BINARY NUMBERS

SIGNED EXAMPLES

- Below are some examples for the signed binary numbers using 6 bits.

Decimal	Signed-magnitude	1's complement	2's complement
0	00 0000	00 0000	00 0000
1	00 0001	00 0001	00 0001
-1	10 0001	11 1110	11 1111
5	00 0101	00 0101	00 0101
-5	10 0101	11 1010	11 1011
12	00 1100	00 1100	00 1100
-12	10 1100	11 0011	11 0100
15	00 1111	00 1111	00 1111
-15	10 1111	11 0000	11 0001
16	01 0000	01 0000	01 0000
-16	11 0000	10 1111	11 0000

- Notice that all representations are the **same for positive numbers**!!!!

BINARY ARITHMETIC

UNSIGNED ADDITION

- Unsigned binary addition follows the standard rules of addition.
  - Examples

```
  1111 0100  Carries
  0011 1011
+ 0111 1010
-----------
  1011 0101
```

```
  0000 0010  Carries
  1011 1001
+ 0100 0101
-----------
  1111 1110
```

```
  1111 0000  Carries
  1111 1001
+ 0100 1000
-----------
1 0100 0000
```

```
  1110 0000 0000.0000  Carries
  0101 1000 1001.1001
+ 0011 0011 0100.01
---------------------
  1000 1011 1101.1101
```

BINARY ARITHMETIC

UNSIGNED SUBTRACTION

- Unsigned binary subtraction follows the standard rules.
  - Examples

```
  0000 0000  Borrows
  1111 1001
- 0100 1000
-----------
  1011 0001
```

```
  1000 1000  Borrows
  1011 1001
- 0100 0101
-----------
  0111 0100
```

```
  1000 0000  Borrows
  0011 1011
- 0111 1010
-----------
  1100 0001
```

```
  0100 1110 1000.1000  Borrows
  0101 1000 1001.1001
- 0011 0011 0100.01
---------------------
  0010 0101 0101.0101
```

BINARY ARITHMETIC

SIGNED ADDITION

- **Signed-magnitude**
  - Add magnitudes if signs are the same, give result the sign
  - Subtract magnitudes if signs are different. Absence or presence of an end borrow determines the resulting sign compared to the augend. If negative, then a 2's complement correction must be taken.
- **2's complement**
  - Add the numbers using normal addition rules. Carry out bit is discarded.
- **1's complement**
  - Easiest to convert to 2's complement, perform the addition, and then convert back to 1's complement. This is done as follows:
    - Add 1 to each integer, add the integers, subtract 1 from the result

BINARY ARITHMETIC

SIGNED SUBTRACTION

- Typically want to do addition or subtraction of **A** and **B** as follows.

$$\mathbf{SUM} = \mathbf{A} + \mathbf{B}$$

$$\mathbf{DIFFERENCE} = \mathbf{A} - \mathbf{B}$$

- If we use **2's complement**, we can make life easy on us since addition and subtraction are done in the same manner: **with addition only!!!**
- A subtraction can be re-reprepresented as follows.

$$\mathbf{SUM} = \mathbf{A} + (-\mathbf{B})$$

- Or in general any two numbers can be added as follows.

$$\mathbf{SUM} = (\pm\mathbf{A}) + (\pm\mathbf{B})$$

BINARY ARITHMETIC

SIGNED MATH EXAMPLE

- Subtraction of signed numbers can best be done with 2's complement.
  - Performed by taking the 2's complement of the subtrahend and then performing addition (including the sign bit).
    - Example:

```
                                         discard carry out
                                       0 0111 1100  Carries
   59          0011 1011                 0011 1011    standard addition
 - 122    =  - 0111 1010    =          + 1000 0110
 ------      -----------               -----------
                                         1100 0001  = -(0011 1111) = -63

    2's complement                          2's complement
```