CHAPTER V

NUMBER SYSTEMS AND ARITHMETIC

NUMBER SYSTEMS

RADIX-R REPRESENTATION

- Decimal number expansion

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

- Binary number representation

$$\mathbf{10110}_2 = (\mathbf{1} \times \mathbf{2^4}) + (\mathbf{0} \times \mathbf{2^3}) + (\mathbf{1} \times \mathbf{2^2}) + (\mathbf{1} \times \mathbf{2^1}) + (\mathbf{0} \times \mathbf{2^0}) = \mathbf{22}_{10}$$

- Hexadecimal number representation

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

NUMBER SYSTEMS

BINARY <-> HEXADECIMAL

BINARY <-> HEXADECIMAL

$\mathbf{0000}_2 = \mathbf{0}_{16}$	$\mathbf{1000}_2 = \mathbf{8}_{16}$
$\mathbf{0001}_2 = \mathbf{1}_{16}$	$\mathbf{1001}_2 = \mathbf{9}_{16}$
$\mathbf{0010}_2 = \mathbf{2}_{16}$	$\mathbf{1010}_2 = \mathbf{10}\ (\mathbf{A}_{16})$
$\mathbf{0011}_2 = \mathbf{3}_{16}$	$\mathbf{1011}_2 = \mathbf{11}\ (\mathbf{B}_{16})$
$\mathbf{0100}_2 = \mathbf{4}_{16}$	$\mathbf{1100}_2 = \mathbf{12}\ (\mathbf{C}_{16})$
$\mathbf{0101}_2 = \mathbf{5}_{16}$	$\mathbf{1101}_2 = \mathbf{13}\ (\mathbf{D}_{16})$
$\mathbf{0110}_2 = \mathbf{6}_{16}$	$\mathbf{1110}_2 = \mathbf{14}\ (\mathbf{E}_{16})$
$\mathbf{0111}_2 = \mathbf{7}_{16}$	$\mathbf{1111}_2 = \mathbf{15}\ (\mathbf{F}_{16})$

BINARY -> HEXADECIMAL

Group binary by 4 bits from radix point

Examples:

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

$$\underbrace{\mathbf{10}}_{\mathbf{2}}\ \underbrace{\mathbf{1010}}_{\mathbf{A}}\ \underbrace{\mathbf{0110}}_{\mathbf{6}}.\underbrace{\mathbf{1100}}_{\mathbf{C}}\ \underbrace{\mathbf{01}}_{\mathbf{4}}{}_2 = \mathbf{2A6.C4}_{16}$$

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 $\mathbf{41.828125}_{10}$

$$
\begin{aligned}
\mathbf{41} \text{ mod } \mathbf{2} &= \mathbf{1} \quad \text{LSB} \\
\mathbf{20} \text{ mod } \mathbf{2} &= \mathbf{0} \\
\mathbf{10} \text{ mod } \mathbf{2} &= \mathbf{0} \\
\mathbf{5} \text{ mod } \mathbf{2} &= \mathbf{1} \\
\mathbf{2} \text{ mod } \mathbf{2} &= \mathbf{0} \\
\mathbf{1} \text{ mod } \mathbf{2} &= \mathbf{1} \quad \text{MSB}
\end{aligned}
$$

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

Therefore $\mathbf{41.828125}_{10} = \mathbf{101001.110101}_{2}$

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

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 0000.0001  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
-  122   =   - 0111 1010   =  + 1000 0110
------       -----------     -----------
                               1100 0001   = -(0011 1111) = -63
```

standard addition

2’s complement

2’s complement