le 1

Chapter 2 Number Systems and Codes

- Introduction
- Positional Number Systems
- Binary Addition and Subtraction
- Representation of Negative Numbers

le 2

Introduction

Digital design deals with binary digits. In practice, very few numbers, events, conditions, and operations are in binary. This chapter shows how to represent non-binary numeric quantities and how to perform numeric operations in binary.

Slide 3

Positional Number Systems

In the positional number systems, each number is represented by a string of digits. The value of the number is equal to the weighted sum of all digits, with the weights determined by the digit positions and the base (radix) of the numbering system.

Slide 4

Positional Number Systems (cont'd)

In the decimal system, we have 10 digits (0-9) and the base is 10. So

$$7856.32 = 7\times10^{3}+8\times10^{2}+5\times10^{1}+6\times10^{0}+3\times10^{-1}+2\times10^{-2}$$

le 5

Positional Number Systems (cont'd)

In general, for a decimal number with N digits to the left of the decimal point and P digits to the right of the decimal point, its value D is

$$D = \sum_{i=-P}^{N-1} d_i \times b^i$$

where d_i is the ith digit and b is the base of the numbering system.

le 6

Binary Number

Binary numbers consist of a string of digits of values 0 or 1 such as 0101. The leftmost digit is called the *most significant* bit (MSB) while the rightmost one the *least significant* bit (LSB).

Slide 7

Binary Number

A N bit binary number can represent up to 2^N distinctive binary codes. The 2^N distinctive codes can be formed by successively adding 1 to the previous code until all N bits are 1s. For example, for 2 bit binary number, it has 4 codes: 00 01 10 11.

Slide 8

Binary and Decimal Number Conversion

- Unsigned binary to Decimal

  $10011_2 = 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 19_{10}$

- Decimal to unsigned binary

  The binary equivalent of a decimal number is obtained by successively dividing the decimal number by 2 until the quotient is 1. The binary number is formed by the reminder (modulus) of each successive division and the final quotient from right to left.

  For example,

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

le 9

HEXADECIMAL Number

HEXADECIMAL number has a base of 16, consisting of 16 digits ranging from 0 to 9 and A to F. For example, F1A3. Hex numbers are often used to describe computer memory location.

e 10

HEX to DECIMAL

- HEX to Decimal

$$F1A3_{16} = 15 \times 16^3 + 1 \times 16^2 + 10 \times 16^1 + 3 \times 16^0 = 61859_{10}$$

- Decimal to HEX

  The HEX equivalent of a decimal number may be obtained by successively dividing the decimal number by 16 until the quotient is less than 16. The HEX number is formed by the reminder of each successive division and the final quotient from right to left.

  For example,

$$3417_{10} = D59_{16}$$

Slide 11

HEX and Binary Conversion

- HEX to Binary: replace each HEX digit with four corresponding binary digits as shown in Table 2-1.

Table 2-1 Binary, decimal, octal, and hexadecimal numbers.

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

Slide 12

$$F1A3_{16} = 1111000110100011$$

- Binary to HEX: starting from right, replace each four binary digits with the corresponding HEX digit as shown in Table 2-1. Add leading 0's if there are fewer than 4 bits.

  For example,

$$01100011001110_2 = 18CE_{16}$$

Refer to Table 2-2 for a summary of conversion methods.

e 13

Binary to Decimal

For a binary number of many digits, to convert it to decimal, it is convenient to convert it to HEX first, and then from HEX to decimal. Similarly, for a large decimal number, to convert it to binary, it will be more efficient to convert it to HEX and then to binary.

Slide 15

OCTAL Numbers (cont'd)

OCTAL to binary conversion can be accomplished the same way as HEX to binary except that each OCTAL digit corresponds to 3 binary bit. While converting binary to OCTAL, starting from the LSB and working towards left, separating binary bits into groups of 3 bits and replacing each group with the corresponding OCTAL digit.

Octal to Hex (or vice verse) conversion needs to convert to binary first, then to Hex,

e 14

OCTAL Numbers

The OCTAL numbering system has a base of 8, using digits 0-7. Each OCTAL digit can be uniquely represented by 3 binary bits as shown in Table 2-1.

Slide 16

Binary Addition and Subtraction

- Addition
- Subtraction

Slide 17

Binary Addition Table

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

Slide 18

Binary Addition: Example

Like decimal addition, binary addition proceeds from right to left and align the LSB and padding zeros to the left.

10111110 + 10001101= 101001011

see more examples on page 32

Slide 19

Binary Subtraction Table

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

Slide 20

Binary Subtraction: Example

11100101 - 00101110 = 10110111

see more examples on page 33

Slide 21

HEX Addition and Subtraction

Addition and subtraction in HEX can be done similarly. We can convert each HEX digit to decimal digit and perform operation in decimal, and then convert the result back to HEX. see the example in page 34.

19B9 + C7E6 = E19F

E19F-C7E6 = 19B9

e 22

Representation of Negative Numbers

- Signed Magnitude Representation
- Two's Complement Representation
- Two's Complement Addition and Subtraction

Slide 24

Two's Complement Representation

To represent a signed number in binary, we use two's complement binary system.

e 23

Signed Magnitude Representation

The MSB is the sign bit, with 0 for positive and 1 for negative number. The major problem with this representation is that we have two zeros: +0 and -0. It also has difficulty in performing addition and subtraction.

Slide 25

Two's Complement Representation (cont'd)

According to the two's complement representation, the MSB is the sign bit. A number is positive if the MSB is zero or negative if MSB is 1. The decimal equivalent for a two's complement binary number is computed the same way as for an unsigned number, except that the weight of the MSB is -2^{n-1}. So in two's complement, binary and decimal conversion is

$$D = \sum_{i=0}^{n-1} d_i \times 2^i$$

e 26

, where d_i is -d_i for the MSB.

Slide 28

Two's Complement Number Range

Like for unsigned binary, a N bit two's complement number can represent 2^N decimal values ranging from -2^{n-1} to $2^{n-1}-1$, one extra negative number (-2^{n-1}) that does not have its positive counterpart.

e 27

Two's Complement Representation (cont'd)

Two's complement means *binary* complement. The two's complement of a binary number is another binary number which when added to the number yields zero if the MSB of the sum is discarded

The two's complement of a binary number is obtained by complementing each individual bit of the number and then add 1 to the complement number. For example, see page 37.

Slide 29

Decimal to Two's Complement

- Determine the number of bits needed to represent the decimal number in two's complement.
- for positive decimal number, proceed the same way as for unsigned binary number.
- for negative decimal number, obtain two's complement for the corresponding positive number first, followed by performing two's complementing on the binary number, yielding the two's complement of the negative number.

For example, for number 10, we need 5 bits, its two's complement is 01010. For -10, its two's complement is obtained by two's complementing 01010, leading to 10110.

bit.

8-bit to 4 bit

00000011 $->$ 0011

11111010 $->$ 1010

Two's Complement Representation (cont'd)

Convert an n-bit two's complement to a m-bit two's complement, if $m > n$, pad $m - n$ copies of sign bit to the left of the number.

For example, to convert a 4-bit two's complement number to a 8-bit complement number, pad 4 zeros.

0011 $->$ 00000011

1010 $->$ 11111010

If $m < n$, discard $n - m$ leftmost bits and the result is valid only if the discarded bits are the same as the sign

Two's Complement Addition

Two's complement addition can proceed the same as the unsigned binary number addition, ignoring any carries beyond the MSB so long as the sum remains within the range (no overflow).

see example on page 39

e 34

Two's Complement Addition

Overflow occurs when the sum of two two's complements exceeds the range of the number system. An addition overflows if the signs of the add-ends are the same and the sign of the sum is different from the add-ends' sign.

Another way to detect overflow is looking at the carry-in and carry-out of the sign bit. Overflow occurs when the carry-in and carry-out of the sign bit is different.

See examples on page 41. Note overflow may also

e 35

occur if we subtract two numbers of different signs.

Slide 36

Two's Complement Subtraction

Two's complement subtraction can be performed using the same procedure as for the unsigned binary numbers.

Alternatively, we can convert two's complement subtraction into two's complement addition by converting subtrahend to its two's complement. See example on page 42.

Slide 37

Other Issues

- Binary Code for Decimal (BCD)
- Gray code
- Character Codes

e 38

Binary Code for Decimal (BCD)

BCD encodes the digits 0 to 9 by the 4-bit unsigned binary representation ,0000 through 1001. The code words 1010 to 1111 are not used. see table 2.9. Packed-BCD allows to to place two BCD digits in one byte, therefore representing values from 0-99.

e 39

Gray Code

Gray codes are binary numbers that have only one bit change between successive code words. They may be used to represent successive states or positions. Note binary codes of successive numbers may have more than one bit change between successive codes. See 3-bit gray code in table 2-10.

Slide 40

Gray Codes Construction

Gray codes of any number of bits can be reconstructed recursively using the three rules:

- A 1-bit gray code has two codes 0 and 1
- The first 2^{n-1} code words of an n-bit gray code equal the code words of (n-1)-bit gray code, written in order with a leading zero appended.
- The last 2^{n-1} code words of an n-bit gray code equal the code words of (n-1)-bit gray code, but written in reverse order with a leading 1 appended.

Slide 41

Character Codes

Character codes are binary numbers used to represent characters. The most commonly used character code is ASCII codes, which represent characters with 7 bit, representing a total of 128 characters.

see table 2-11.

Chapter 2 Summary

- different positional number systems: decimal, binary, hex, and octal.
- conversion between different numbering systems.
- Binary addition and subtraction
- Representation of signed decimal number using two's complement binary system
  - two's complement binary number range
  - decimal and two's complement conversion
  - overflow

  - two's complement addition and subtraction
  - two's complement representation using different bits
- Gray code