Arithmetic
Operations on Digital and Binary Numbers (1)

Lecture 02
Book Chapter(s): 1

COE211-Digital Logic Design

الفصل الدراسي الأول 1442-Fall2020

جامعة طيبة فرع ينبع - كلية علوم وهندسة الحاسبات – شطر الطالبات

د. فاطمة الحربي

How to Convert from Decimal to Binary?

- Two methods:
  1. Method 1: Sum-of-weights
  2. Method 2:
     a) Repeated Division-by-2 (for whole numbers)
     b) Repeated Multiplication-by-2 (for fractions)

Method1: Sum-Of-Weights

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

- $(9)_{10} = 8 + 1 = 2^3 + 2^0 = (1001)_2$
- $(18)_{10} = 16 + 2 = 2^4 + 2^1 = (10010)_2$
- $(58)_{10} = 32 + 16 + 8 + 2 = 2^5 + 2^4 + 2^3 + 2^1 = (111010)_2$
- $(0.625)_{10} = 0.5 + 0.125 = 2^{-1} + 2^{-3} = (0.101)_2$

Method2-a: Repeated Division-by-2 (whole numbers)

- To convert a whole number 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: $(43)_{10} = (101011)_2$

2	43
2	21 rem 1 □ LSB
2	10 rem 1
2	5 rem 0
2	2 rem 1
2	1 rem 0
	0 rem 1 □ MSB

Method2-a: Repeated Multiplication-by-2 (Decimal Fractions)

- To convert decimal fractions to binary, repeated multiplication by 2 is used, until the fractional product is 0 (or until the desired number of decimal places). The carried digits, or *carries*, produce the answer, with the first carry as the MSB, and the last as the LSB.
- Example: $(0.3125)_{10} = (.0101)_2$

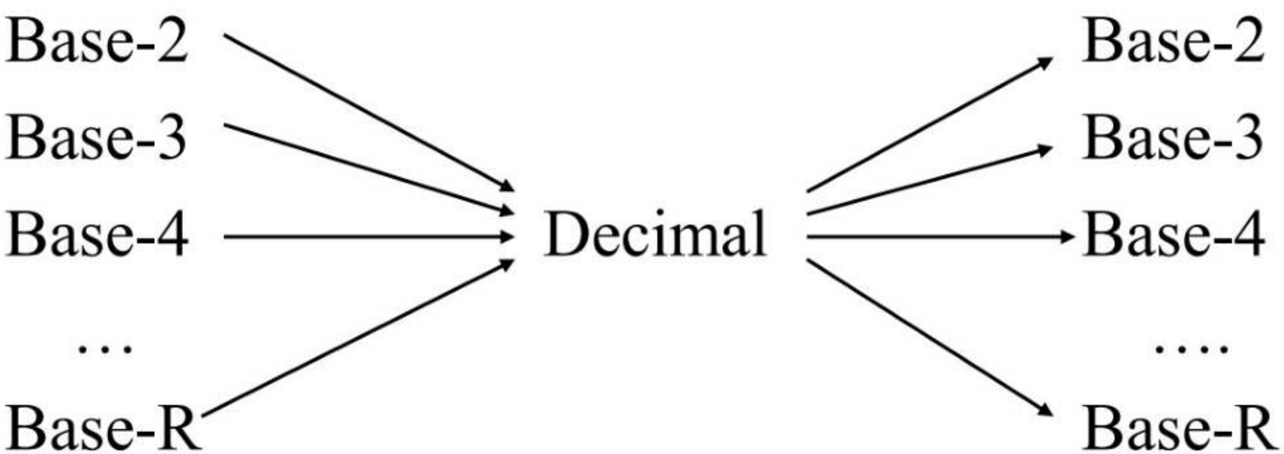
	Carry
0.3125 x 2 = 0.625	0 □ MSB
0.625 x 2 = 1.25	1
0.25 x 2 = 0.50	0
0.5 x 2 = 1	1 □ LSB

Conversion between Decimal and Other Bases

- Base-R to decimal: multiply digits with their corresponding weights.
- Decimal to binary (base 2):
  - Whole numbers: repeated division-by-2
  - Fractions: repeated multiplication-by-2
- Decimal to base-R:
  - Whole numbers: repeated division-by-R
  - Fractions: repeated multiplication-by-R

Conversion between Bases

- In general, conversion between bases can be done via decimal:



- Shortcuts for conversion between bases 2, 4, 8, 16.

Binary-Octal/Hexadecimal Conversion

- Binary Octal: Partition in groups of 3
  $(10\ 111\ 011\ 001\ .\ 101\ 110)_2 = (2731.56)_8$

- Octal Binary: reverse
  $(2731.56)_8 = (10\ 111\ 011\ 001\ .\ 101\ 110)_2$

- Binary Hexadecimal: Partition in groups of 4
  $(101\ 1101\ 1001\ .\ 1011\ 1000)_2 = (5D9.B8)_{16}$

- Hexadecimal Binary: reverse
  $(5D9.B8)_{16} = (101\ 1101\ 1001\ .\ 1011\ 1000)_2$

Binary Arithmetic Operations – Addition (1)

- Like decimal numbers, two numbers can be added by adding each pair of digits together with carry propagation.

$$\begin{array}{r} (11011)_2 \\ +\ (10011)_2 \\ \hline (101110)_2 \\ \hline \end{array} \qquad \begin{array}{r} (647)_{10} \\ +\ (537)_{10} \\ \hline (1184)_{10} \\ \hline \end{array}$$

Binary Arithmetic Operations – Addition (1)

- Digit addition table:

BINARY	DECIMAL
0 + 0 + 0 = 0 0	0 + 0 + 0 = 0 0
0 + 1 + 0 = 0 1	0 + 1 + 0 = 0 1
1 + 0 + 0 = 0 1	0 + 2 + 0 = 0 2
1 + 1 + 0 = 1 0	...
0 + 0 + 1 = 0 1	1 + 8 + 0 = 0 9
0 + 1 + 1 = 1 0	1 + 9 + 0 = 1 0
1 + 0 + 1 = 1 0	...
1 + 1 + 1 = 1 1	9 + 9 + 1 = 1 9

Carry in

Carry out

$$\begin{array}{r} (11011)_2 \\ +\ (10011)_2 \\ \hline (101110)_2 \end{array}$$

1	0	0	1	1	0
0	1	1	0	1	1
0	1	0	0	1	1
1	0	1	1	1	0
0	1	0	0	1	1

Binary Arithmetic Operations – Subtraction (1)

- Two numbers can be subtracted by subtracting each pair of digits together with borrowing, where needed.

$$
\begin{array}{r} (11001)_2 \\ -\ (10011)_2 \\ \hline (00110)_2 \\ \hline \end{array}
\qquad\qquad
\begin{array}{r} (627)_{10} \\ -\ (537)_{10} \\ \hline (090)_{10} \\ \hline \end{array}
$$

Binary Arithmetic Operations – Subtraction (2)

- Digit subtraction table:

BINARY	DECIMAL
0 - 0 - 0 = 0 0	0 - 0 - 0 = 0 0
0 - 1 - 0 = 1 1	0 - 1 - 0 = 1 9
1 - 0 - 0 = 0 1	0 - 2 - 0 = 1 8
1 - 1 - 0 = 0 0	...
0 - 0 - 1 = 1 1	0 - 9 - 1 = 1 0
0 - 1 - 1 = 1 0	1 - 0 - 1 = 0 0
1 - 0 - 1 = 0 0	...
1 - 1 - 1 = 1 1	9 - 9 - 1 = 1 9

↑ Borrow

$$
\begin{array}{r}
(11001)_2 \\
-\ (10011)_2 \\
\hline
(00110)_2 \\
\hline
\end{array}
$$

0	0	1	1	0	0
0	1	1	0	0	1
0	1	0	0	1	1
0	0	0	1	1	0
0	0	0	1	1	0