CS 223 – Digital Logic and Design

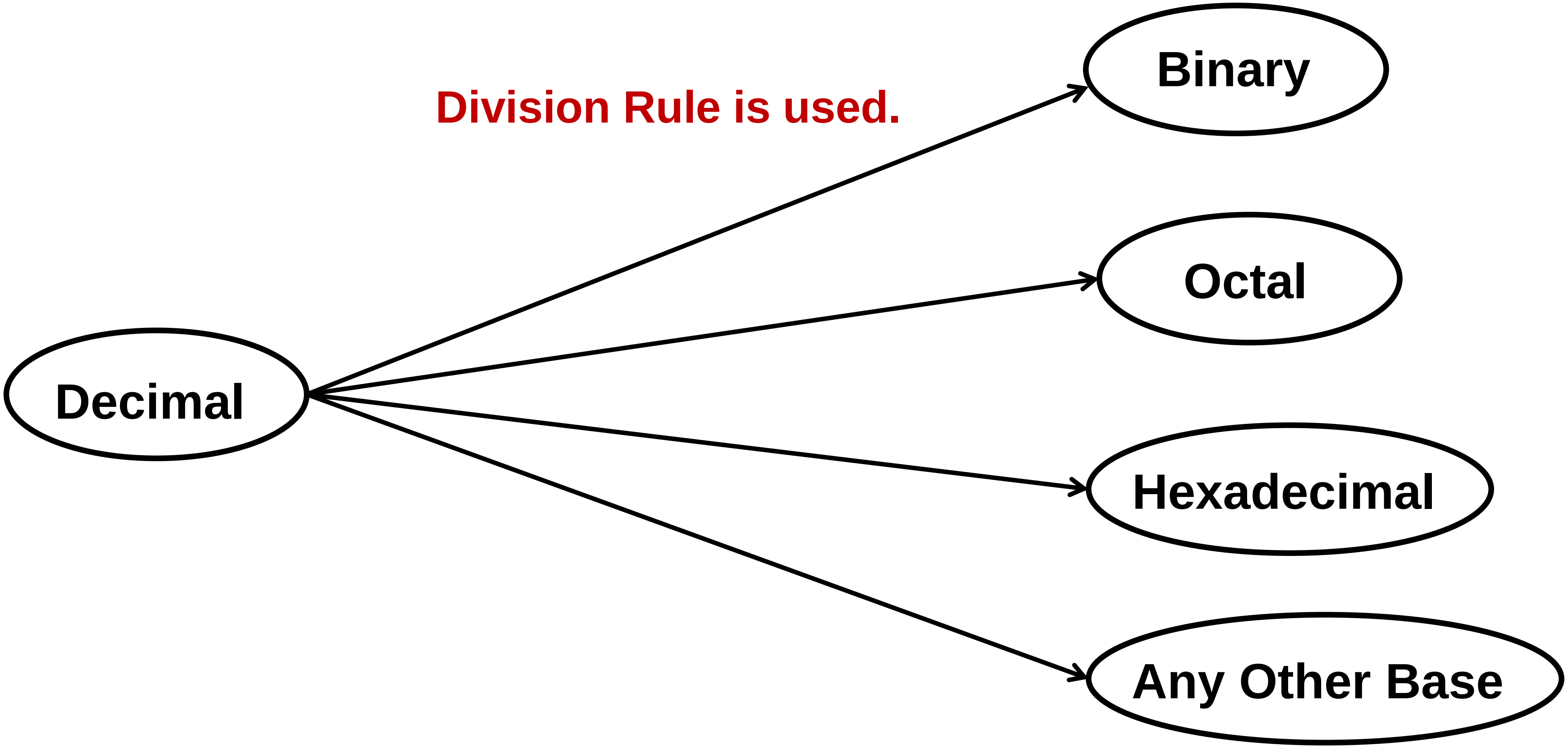
Lecture 4 – Number-Base Conversions

Sameer Akram

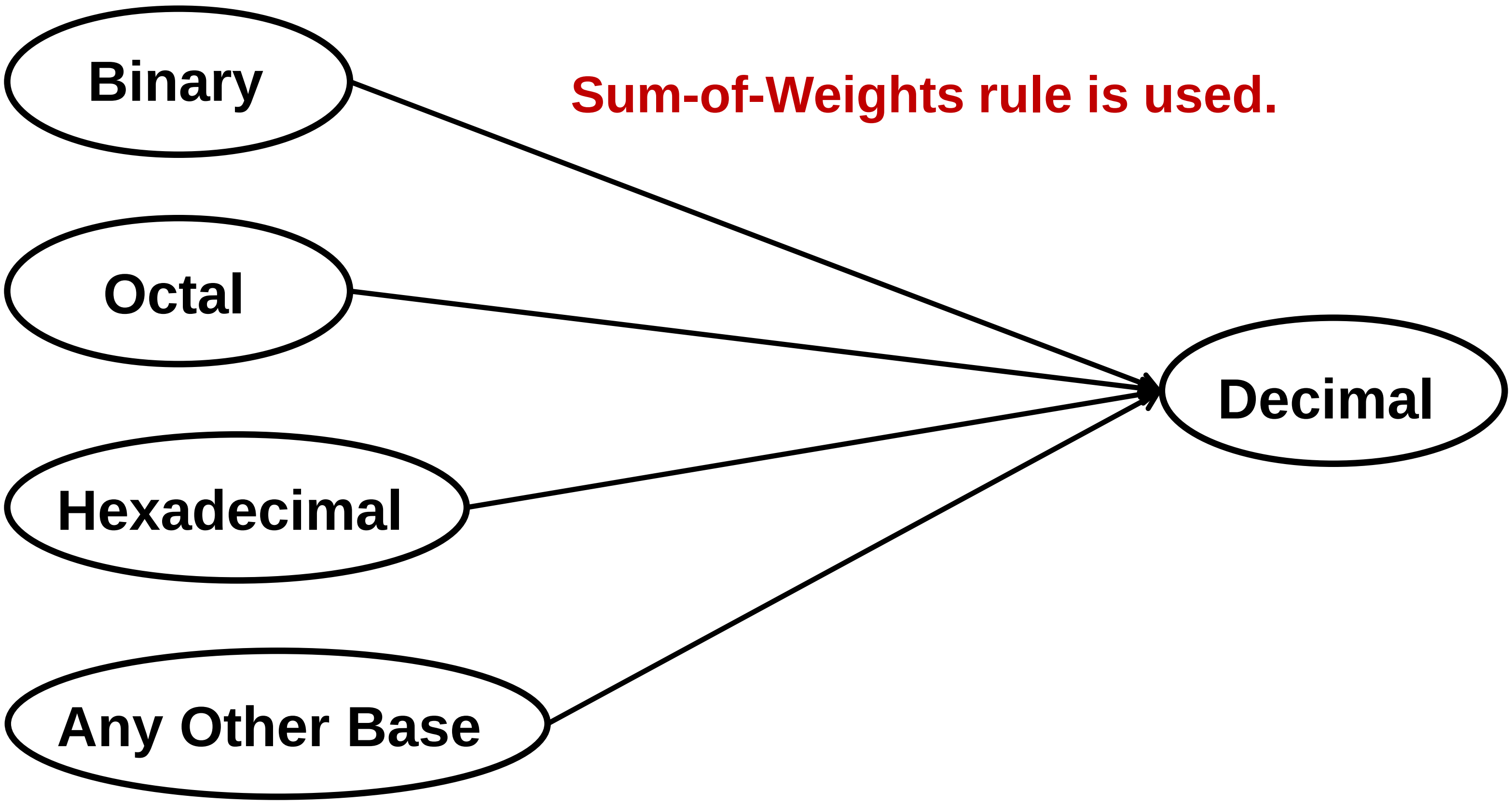
Lecture 4

Number-Base Conversions

Number-Base Conversions



Number-Base Conversions (Contd.)



Number-Base Conversions (Contd.)

EXAMPLE 1.1

Convert decimal 41 to binary. First, 41 is divided by 2 to give an integer quotient of 20 and a remainder of $\frac{1}{2}$. Then the quotient is again divided by 2 to give a new quotient and remainder. The process is continued until the integer quotient becomes 0. The *coefficients* of the desired binary number are obtained from the *remainders* as follows:

	Integer Quotient		Remainder	Coefficient
$41/2 =$	20	+	$\frac{1}{2}$	$a_0 = 1$
$20/2 =$	10	+	0	$a_1 = 0$
$10/2 =$	5	+	0	$a_2 = 0$
$5/2 =$	2	+	$\frac{1}{2}$	$a_3 = 1$
$2/2 =$	1	+	0	$a_4 = 0$
$1/2 =$	0	+	$\frac{1}{2}$	$a_5 = 1$

Therefore, the answer is $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$.

Number-Base Conversions (Contd.)

The arithmetic process can be manipulated more conveniently as follows:

Integer	Remainder
41	
20	1
10	0
5	0
2	1
1	0
0	1 101001 = answer

Conversion from decimal integers to any base-*r* system is similar to this example, except that division is done by *r* instead of 2.

Number-Base Conversions (Contd.)

EXAMPLE 1.2

Convert decimal 153 to octal. The required base r is 8. First, 153 is divided by 8 to give an integer quotient of 19 and a remainder of 1. Then 19 is divided by 8 to give an integer quotient of 2 and a remainder of 3. Finally, 2 is divided by 8 to give a quotient of 0 and a remainder of 2. This process can be conveniently manipulated as follows:

$$\begin{array}{r|l} 153 & \\ 19 & 1 \\ 2 & 3 \\ 0 & 2 = (231)_8 \end{array}$$

The conversion of a decimal *fraction* to binary is accomplished by a method similar to that used for integers. However, multiplication is used instead of division, and integers instead of remainders are accumulated. Again, the method is best explained by example.

Number-Base Conversions (Contd.)

EXAMPLE 1.3

Convert $(0.6875)_{10}$ to binary. First, 0.6875 is multiplied by 2 to give an integer and a fraction. Then the new fraction is multiplied by 2 to give a new integer and a new fraction. The process is continued until the fraction becomes 0 or until the number of digits has sufficient accuracy. The coefficients of the binary number are obtained from the integers as follows:

	Integer		Fraction	Coefficient
$0.6875 \times 2 =$	1	+	0.3750	$a_{-1} = 1$
$0.3750 \times 2 =$	0	+	0.7500	$a_{-2} = 0$
$0.7500 \times 2 =$	1	+	0.5000	$a_{-3} = 1$
$0.5000 \times 2 =$	1	+	0.0000	$a_{-4} = 1$

$(41.6875)_{10} \rightarrow (101001.1011)_2$

The conversion of decimal numbers with both integer and fraction parts is done by converting the integer and the fraction separately and then combining the two answers. Using the results of Examples 1.1 and 1.3, we obtain

$$(41.6875)_{10} = (101001.1011)_2$$

OCTAL and HEXADECIMAL Numbers

1.4 OCTAL AND HEXADECIMAL NUMBERS

The conversion from and to binary, octal, and hexadecimal plays an important role in digital computers, because shorter patterns of hex characters are easier to recognize than long patterns of 1's and 0's. Since $2^3 = 8$ and $2^4 = 16$, each octal digit corresponds to three binary digits and each hexadecimal digit corresponds to four binary digits. The first 16 numbers in the decimal, binary, octal, and hexadecimal number systems are listed in Table 1.2.

Table 1.2 – Numbers with Different Bases

Table 1.2
Numbers with Different Bases

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Shortcut Conversion from BINARY → OCTAL & HEXADECIMAL

The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right. The corresponding octal digit is then assigned to each group. The following example illustrates the procedure:

$$\begin{array}{ccccccccccl} (10 & 110 & 001 & 101 & 011 & \cdot & 111 & 100 & 000 & 110)_2 & = (26153.7406)_8 \\ 2 & 6 & 1 & 5 & 3 & & 7 & 4 & 0 & 6 & \end{array}$$

Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of *four* digits:

$$\begin{array}{cccccccl} (10 & 1100 & 0110 & 1011 & \cdot & 1111 & 0010)_2 & = (2C6B.F2)_{16} \\ 2 & C & 6 & B & & F & 2 & \end{array}$$

The corresponding hexadecimal (or octal) digit for each group of binary digits is easily remembered from the values listed in Table 1.2.

Shortcut Conversion from OCTAL & HEXADECIMAL → BINARY

Conversion from octal or hexadecimal to binary is done by reversing the preceding procedure. Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent. The procedure is illustrated in the following examples:

$$\begin{array}{llllllll} (673.124)_8 = (110 & 111 & 011 & \cdot & 001 & 010 & 100)_2 \\ \quad\quad\quad\quad\quad\;\; 6 & 7 & 3 & & 1 & 2 & 4 \end{array}$$

and

$$\begin{array}{lllll} (306.D)_{16} = (0011 & 0000 & 0110 & \cdot & 1101)_2 \\ \quad\quad\quad\quad\quad\;\; 3 & 0 & 6 & & D \end{array}$$

Most computer manuals use either octal or hexadecimal numbers to specify binary quantities .

The choice between them is arbitrary, although hexadecimal tends to win out, since it can represent a byte with two digits.

Complements

- Complements are used in digital computers to **simplify the subtraction operation** and for **logical manipulation**.
- Simplifying operations leads to simpler, less expensive circuits to implement the operations.
- There are two types of complements for each **base-r** *system:*
- **The radix complement** and **the diminished radix complement**.
- The first is referred to as the ***r's complement*** *and the second as the* ***(r - 1)'s complement****.*
- When the value of the base r is substituted in the name, the two types are referred to as the 2's complement and 1's complement for binary numbers and the 10's complement and 9's complement for decimal numbers.

Diminished Radix Complement OR (r – 1)'s Complement

- Given a number ***N*** *in base* ***r*** *having* ***n*** *digits,*
  the ***(r – 1)'s*** *complement of* ***N*** , is defined as $(r^n - 1) - N$.
- For decimal numbers, $r = 10$ and $r - 1 = 9$, *so the* ***9's*** *complement of* ***N*** *is* $(10^n - 1) - N$.
- In this case, 10^n represents a number that consists of a single ***1*** followed by ***n 0's***.
- $10^n - 1$ is a number represented by ***n 9's***.
- Example: N = 546700, r =10, n = 6
  9's complement of N = $(10^6 - 1) - 546700$
  = 999999 – 546700
  = 453299
- 9's complement of 546700 is 999999 – 546700 = 453299
- 9's complement of 012398 is 999999 – 012398 = 987601

(r – 1)'s Complement (Contd.)

- For binary numbers, **r = 2** and **r – 1 = 1**, so the **1's** complement of **N** is **$(2^n – 1) – N$**.
- Again, **2^n** is represented by a binary number that consists of a 1 followed by **n 0's**.
- **$2^n – 1$** is a binary number represented by **n *1's***.
- For example, if **n = 4**, we have **$2^4 = (10000)_2$**
- and **$2^4 – 1 = (1111)_2$**.
- Thus, the **1's complement** of a binary number is obtained by
- subtracting each digit from **1**.
- However, when subtracting binary digits from **1**, we can have either **1 - 0 = 1** or **1 - 1 = 0**, which causes the bit to change from **0** to **1** or from **1** to **0**, respectively.
- Therefore, **the 1's complement of a binary number is formed by changing 1's to 0's and 0's to 1's**.

(r – 1)'s Complement (Contd.)

- Examples:
- The **1's** complement of **1011000** is **0100111**.
- The **1's** complement of **0101101** is **1010010**.
- The **(r – 1)'s** *complement of octal or hexadecimal numbers is obtained by subtracting* each digit from **7** or **F** (decimal **15**), respectively.

Reading and Exercise Assignment

Radix Complement OR r’s Complement

- That’s end of the presentation ! ☺