ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Applied Discrete Mathematics

Spring 2024 - **Dr. Abdelgwad Elashry**
Abdelgwad.Elashry@sut.edu.eg

ELSEWEDY UNIVERSITY OF TECHNOLOGY
- POLYTECHNIC OF EGYPT -

Applied Discrete Mathematics
Lecture 4

Table of Contents

Computer Number Systems

❖ NUMBER SYSTEM

Number systems are the technique **to represent numbers in the computer system architecture**, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

Computer Number Systems

BINARY NUMBER SYSTEM

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The **base of binary number system is 2**, because it has only two digits.

$$(11110000)_2$$

OCTAL NUMBER SYSTEM

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The **base of octal number system is 8**, because it has only 8 digits.

$$(360)_8$$

Computer Number Systems

DECIMAL NUMBER SYSTEM

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The **base of decimal number system is 10**, because it has only 10 digits.

$$(890)_{10}$$

HEXADECIMAL NUMBER SYSTEM

A Hexadecimal number system **has sixteen (16) alphanumeric values from 0 to 9 and A to F.** Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. **Here A is 10, B is 11, C is 12, D is 13, E is 14 and F is 15.**

$$(F9B0)_{16}$$

Computer Number Systems

Number system	Base(Radix)	Used digits	Example
Binary	2	0,1	$(11110000)_2$
Octal	8	0,1,2,3,4,5,6,7	$(360)_8$
Decimal	10	0,1,2,3,4,5,6,7,8,9	$(240)_{10}$
Hexadecimal	16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	$(F0)_{16}$

Computer Number Systems

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

Computer Number Systems

Example 1: The Decimal Number 7654.23 Analyzed into the following:

$$7\,6\,5\,4\,.\,2\,3$$

7×1000 | 6×100 | 5×10 | 4×1 | $2 \times 1/10$ | $3 \times 1/100$

7×10^{3} | 6×10^{2} | 5×10^{1} | 4×10^{0} | 2×10^{-1} | 3×10^{-2}

$$a_5a_4a_3a_2a_1a_0.\ a_{-1}a_{-2}a_{-3}$$

Computer Number Systems

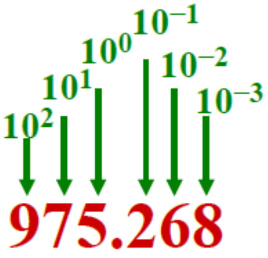
Thus, the preceding **decimal number** (7392) can be expressed as:

$$10^5 a_5 + 10^4 a_4 + 10^3 a_3 + 10^2 a_2 + 10^1 a_1 + 10^0 a_0 + 10^{-1} a_{-1} + 10^{-2} a_{-2} + 10^{-3} a_{-3}$$

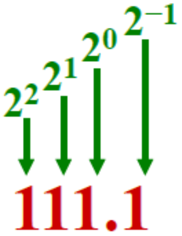
with $a_3 = 7, a_2 = 3, a_1 = 9$, and $a_0 = 2$.

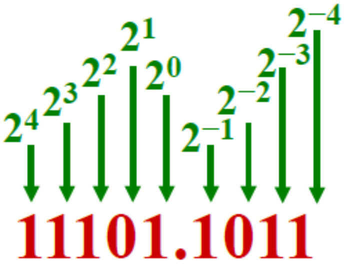
Computer Number Systems

- In a decimal number, a non-0 digit in a column is treated as the multiplier of the power of 10 represented by that column (0's clearly add no value).



- We read binary numbers the same way; 0's count nothing and a 1 in any column means that the power of 2 represented by that column is part of the magnitude of the number. That is:





Conversion

Binary –to– Decimal Process

The Process : *Weighted Multiplication*

a) Multiply each bit of the *Binary Number* by it corresponding bit-weighting factor (i.e. Bit-0→2^0=1; Bit-1→2^1=2; Bit-2→2^2=4; etc).

b) Sum up all the products in step (a) to get the *Decimal Number*.

Example:

Convert the decimal number 0110_2 into its decimal equivalent.

0	1	1	0	
2^3	2^2	2^1	2^0	Bit-Weighting Factors
8	4	2	1	

$0 + 4 + 2 + 0 = 6_{10}$

$\boxed{\therefore\ 0110_2 = 6_{10}}$

Computer Number Systems

Example:

Convert the binary number 10010_2 into its decimal equivalent.

Solution:

1		0		0		1		0			
2^4		2^3		2^2		2^1		2^0			
16		8		4		2		1			
16	+	0	+	0	+	2	+	0	=	18_{10}	

$$\boxed{\therefore 10010_2 = 18_{10}}$$

Binary → Dec : More Examples

a) $0110_2 = ?$

b) $11010_2 = ?$

c) $0110101_2 = ?$

d) $11010011_2 = ?$

Binary → Dec : More Examples

a) $0110_2 = ?$ 6_{10}

b) $11010_2 = ?$ 26_{10}

c) $0110101_2 = ?$ 53_{10}

d) $11010011_2 = ?$ 211_{10}

Computer Number Systems

Decimal (*Integer*) to Binary Conversion

- Divide the number by the 'Base' (=2)
- Take the remainder (either 0 or 1) as a coefficient
- Take the quotient and repeat the division

Example: $(13)_{10}$

	Quotient	Remainder	Coefficient
13 / 2 =	6	1	$a_0 = 1$
6 / 2 =	3	0	$a_1 = 0$
3 / 2 =	1	1	$a_2 = 1$
1 / 2 =	0	1	$a_3 = 1$

Answer: $(13)_{10} = (a_3\, a_2\, a_1\, a_0)_2 = (1101)_2$

MSB LSB

Computer Number Systems

Decimal (*Integer*) to Binary Conversion

Example:

Convert the decimal number 26_{10} into its binary equivalent.

Solution:

$$2\,\overline{)\,26}^{\;13} \quad r = 0 \leftarrow \text{LSB}$$

$$2\,\overline{)\,13}^{\;6} \quad r = 1$$

$$2\,\overline{)\,6}^{\;3} \quad r = 0$$

$$2\,\overline{)\,3}^{\;1} \quad r = 1$$

$$2\,\overline{)\,1}^{\;0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\; 26_{10} = 11010_2}$$

Decimal (*Integer*) to Binary Conversion

Example:

Convert the decimal number 41_{10} into its binary equivalent.

Solution:

$$2\,\overline{)\,41}^{\,20} \quad r = 1 \leftarrow \text{LSB}$$

$$2\,\overline{)\,20}^{\,10} \quad r = 0$$

$$2\,\overline{)\,10}^{\,5} \quad r = 0$$

$$2\,\overline{)\,5}^{\,2} \quad r = 1$$

$$2\,\overline{)\,2}^{\,1} \quad r = 0$$

$$2\,\overline{)\,1}^{\,0} \quad r = 1 \leftarrow \text{MSB}$$

$$\boxed{\therefore\ 41_{10} = 101001_2}$$

Decimal (*Integer*) to Binary Conversion

Dec → Binary : More Examples

a) $13_{10} = ?$

b) $22_{10} = ?$

c) $43_{10} = ?$

d) $158_{10} = ?$

Decimal (*Integer*) to Binary Conversion

Dec → Binary : More Examples

a) 13_{10} = ? $1\,1\,0\,1_2$

b) 22_{10} = ? $1\,0\,1\,1\,0_2$

c) 43_{10} = ? $1\,0\,1\,0\,1\,1_2$

d) 158_{10} = ? $1\,0\,0\,1\,1\,1\,1\,0_2$

Computer Number Systems

Decimal (*Fraction*) to Binary Conversion

- Multiply the number by the 'Base' (=2)
- Take the integer (either 0 or 1) as a coefficient
- Take the resultant fraction and repeat the division

Example: $(0.625)_{10}$

		Integer		Fraction	Coefficient
0.625	* 2 =	1	.	25	$a_{-1} = 1$
0.25	* 2 =	0	.	5	$a_{-2} = 0$
0.5	* 2 =	1	.	0	$a_{-3} = 1$

Answer: $(0.625)_{10} = (0.a_{-1}\ a_{-2}\ a_{-3})_2 = (0.101)_2$

MSB LSB

$a_5a_4a_3a_2a_1a_0.\ a_{-1}a_{-2}a_{-3}$

Computer Number Systems

Decimal (*Fraction*) to Binary Conversion

Ex. Convert $(13.125)_{10}$ into Binary Number:

Computer Number Systems

Decimal (*Fraction*) to Binary Conversion

Ex. Convert $(13.125)_{10}$ into Binary Number:

	Result	Reminder
$13 \div 2 =$	6	**1**
$6 \div 2 =$	3	**0**
$3 \div 2 =$	1	**1**
$1 \div 2 =$	0	**1**

$(1\ 1\ 0\ 1)_2$

$0.125 \times 2 =$	**0**.25
$0.25 \times 2 =$	**0**.5
$0.5 \times 2 =$	**1**.0

$(0.\ 0\ 0\ 1\ )_2$

$(1101.001)_2$ = The Result

$a_5 a_4 a_3 a_2 a_1 a_0.\ a_{-1} a_{-2} a_{-3}$

Computer Number Systems

Decimal to Octal Conversion

Example: $(175)_{10}$

	Quotient	Remainder	Coefficient
175 / 8 =	21	7	$a_0 = 7$
21 / 8 =	2	5	$a_1 = 5$
2 / 8 =	0	2	$a_2 = 2$

Answer: $(175)_{10} = (a_2\, a_1\, a_0)_8 = (257)_8$

Example: $(0.3125)_{10}$

	Integer		Fraction	Coefficient
0.3125 * 8 =	2	.	5	$a_{-1} = 2$
0.5 * 8 =	4	.	0	$a_{-2} = 4$

Answer: $(0.3125)_{10} = (0.a_{-1}\, a_{-2}\, a_{-3})_8 = (0.24)_8$

Computer Number Systems

Decimal to Octal Conversion

Ex. Convert $(145.45)_{10}$ into Octal Number:

	Result	Reminder
$145 \div 8 =$	18	**1**
$18 \div 8 =$	2	**2**
$2 \div 8 =$	0	**2**

$(2\ \ 2\ \ 1)_8$

$0.45 \times 8 =$	**3**.6
$0.6 \times 8 =$	**4**.8
$0.8 \times 8 =$	**6**.4
$0.4 \times 8 =$	**3**.2
$0.2 \times 8 =$	**1**.6

$(0.\ 3\ \ 4\ \ 6\ \ 3\ \ 1\ )_8$

$(221.34631)_8$ = the Result

$a_5a_4a_3a_2a_1a_0.\ a_{-1}a_{-2}a_{-3}$

Computer Number Systems

Decimal to Hexadecimal Conversion

Ex. Convert $(394.36)_{10}$ into Hexadecimal Number:

	Result	Reminder
$394 \div 16 =$	24	**10(A)**
$24 \div 16 =$	1	**8**
$1 \div 16 =$	0	**1**

$(1\ \ 8\ \ A)_{16}$

$0.36 \times 16 = \mathbf{5}.76$

$0.76 \times 16 = \mathbf{12}.16$

$0.16 \times 16 = \mathbf{2}.56$

$0.56 \times 16 = \mathbf{8}.96$

$\vdots$

$(0.\ 5\ \ C\ \ 2\ \ 8)_{16}$

$(18A.5C28)_{16} \approx$ the Result

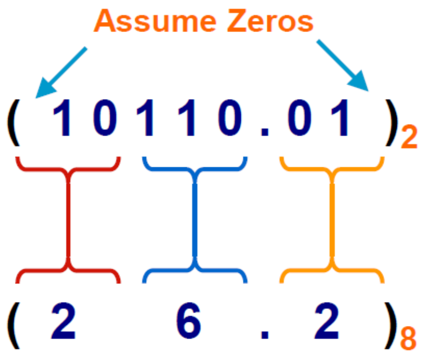
$a_5a_4a_3a_2a_1a_0.\ a_{-1}a_{-2}a_{-3}$

Computer Number Systems

Binary to Octal Conversion

- $8 = 2^3$
- Each group of 3 bits represents an octal digit

Example:



Octal	Binary
0	0 0 0
1	0 0 1
2	0 1 0
3	0 1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1

Works both ways (*Binary* to *Octal* & *Octal* to *Binary*)

Computer Number Systems

Binary to Hexadecimal Conversion

- $16 = 2^4$
- Each group of 4 bits represents a hexadecimal digit

Example:

Assume Zeros

$(\underbrace{1}\,\underbrace{0\,1\,1\,0}\,.\,\underbrace{0\,1})_2$

$(1\quad 6\quad .\quad 4)_{16}$

Hex	Binary
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
A	1 0 1 0
B	1 0 1 1
C	1 1 0 0
D	1 1 0 1
E	1 1 1 0
F	1 1 1 1

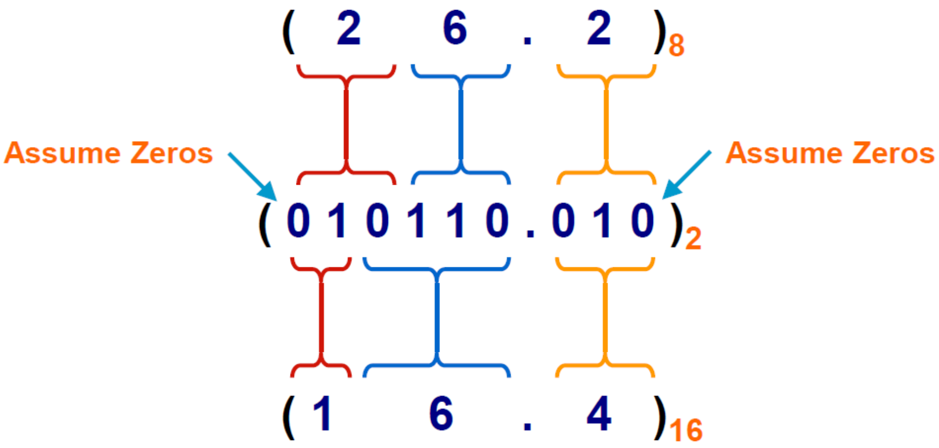
Works both ways (*Binary* to *Hex* & *Hex* to *Binary*)

Computer Number Systems

Octal to Hexadecimal Conversion

- Convert to Binary as an intermediate step

Example:



Works both ways (*Octal* to *Hex* & *Hex* to *Octal*)

Computer Number Systems

Revision: (Conversion to Decimal)

For example, the decimal equivalent of the binary number 11010.11 is 26.75, as shown from the multiplication of the coefficients **by powers of 2**:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 26.75$$

Examples of other numbering systems:

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46{,}687)_{10}$$

Computer Number Systems

Revision: (Conversion to Decimal)

Ex1: Convert the Binary Number $(1101.01)_2$ to Decimal Number?

$$(1101.01)_2 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

$$= 1 \times 1 + 0 \times 2 + 1 \times 4 + 1 \times 8 + 0 \times 1/2 + 1 \times 1/4$$

$$= 1 + 0 + 4 + 8 + 0 + 0.25$$

$$= (13.25)_{10}$$

Computer Number Systems

Revision: (Conversion to Decimal)

For example, the decimal equivalent of the binary number 11010.11 is 26.75, as shown from the multiplication of the coefficients **by powers of 2**:

$$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2} =$$

Examples of other numbering systems:

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} =$$

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 =$$

Computer Number Systems

Decimal, Binary, Octal and Hexadecimal

Decimal	Binary	Octal	Hex
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

Mathematical Operations Number System

Computer Number Systems

Mathematical Operations of Binary Number System

Decimal Addition

$$
\begin{array}{rrrl}
 & 1 & 1 & \leftarrow \text{Carry} \\
 & & 5 & 5 \\
+ & & 5 & 5 \\
\hline
 & 1 & 1 & 0
\end{array}
$$

$$= \textit{Ten} \geq \textit{Base}$$

➔ **Subtract a Base**

$$
\begin{array}{rr}
 & 11 \leftarrow \text{carries} \\
 & 3734 \\
+ & 5168 \\
\hline
 & 8902
\end{array}
$$

Mathematical Operations of Binary Number System

Binary Addition

Adding in binary is identical to the operations in our own decimal system. You simply have to remember the "Addition table." Thus:

0	0	1	1
+0	+1	+0	+1
0	1	1	10

```
   1 1 1 1 1 1
     1 1 1 1 0 1      = 61
+      1 0 1 1 1      = 23
-----------------
   1 0 1 0 1 0 0      = 84
```

$\geq (2)_{10}$

c_{i-1}	0	0	0	0	1	1	1	1
$+a_i$	0	0	1	1	0	0	1	1
$+b_i$	+0	+1	+0	+1	+0	+1	+0	+1
T	0	1	1	10	1	10	10	11

```
    11  ← carries
   1011
+  0011
-------
   1110
```

Computer Number Systems

Mathematical Operations of Binary Number System

Binarv Addition

Adding Examples

1 +1	1 +11	100 +11	101 +110

110 +111	1101 +1100	101110 +110001	1100101 +1110001

Mathematical Operations of Binary Number System

Binary Addition

Adding Answers

1 +1 10	1 +11 100	100 +11 111	101 +110 1011

110 +111 1101	1101 +1100 11001	101110 +110001 1011111	1100101 +1110001 11010110

Computer Number Systems

Mathematical Operations of Binary Number System

Binary Subtraction

- Borrow a “Base” when needed

		1			2			$= (10)_2$
	0	$\cancel{2}$	2	0	0	2		
	$\cancel{1}$	0	0	$\cancel{1}$	$\cancel{1}$	0	1	= 77
−			1	0	1	1	1	= 23
	0	**1**	**1**	**0**	**1**	**1**	**0**	**= 54**

Mathematical Operations of Binary Number System

Binary Multiplication

				1	0	1	1	1
x					1	0	1	0
				0	0	0	0	0
			1	0	1	1	1	
		0	0	0	0	0		
	1	0	1	1	1			
	1	1	1	0	0	1	1	0

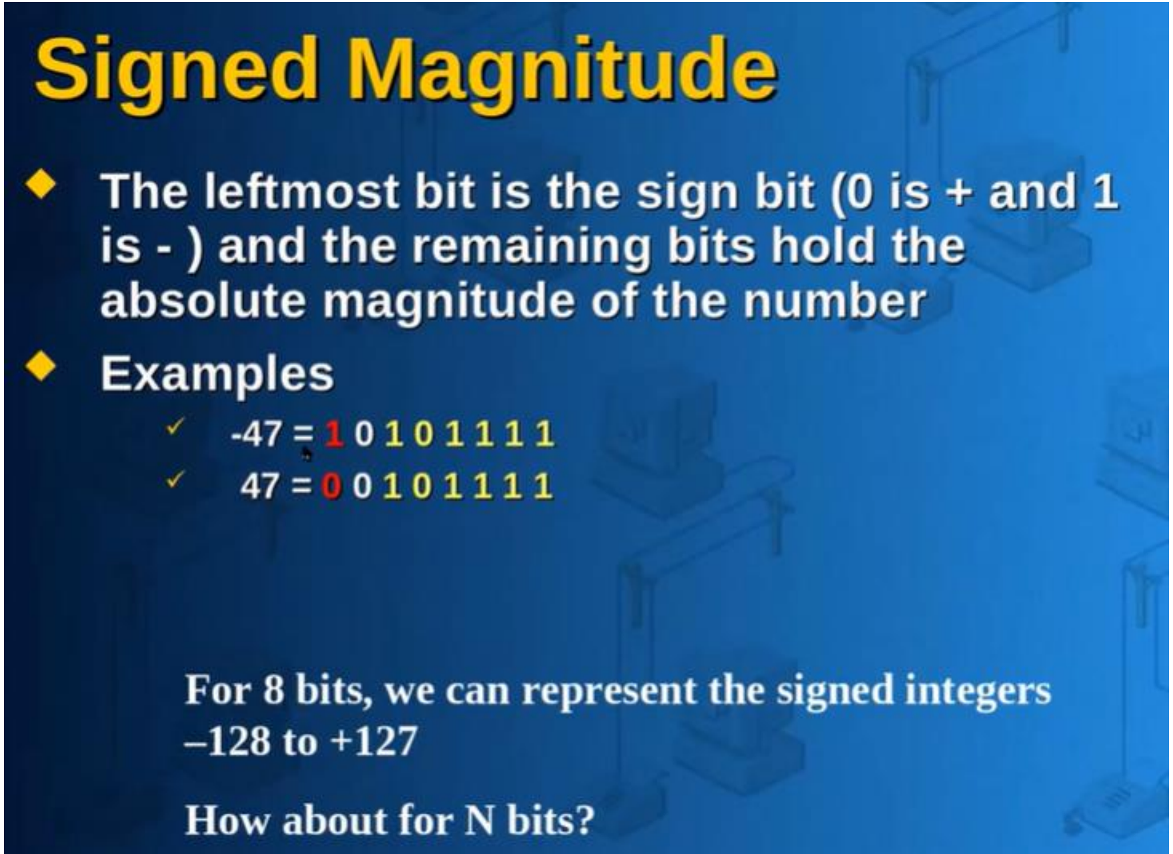
Computer Number Systems

Negative Binary Numbers

- Three different systems have been used
  - Signed magnitude
  - One's complement
  - Two's complement

NOTE: For negative numbers the sign bit is always 1, and for positive numbers it is 0 in these three systems

Computer Number Systems



Computer Number Systems

One's complement

- Replace each 1 by 0 and each 0 by 1
- Example (-6)
  - First represent 6 in binary format (00000110)
  - Then replace (11111001)

Computer Number Systems

Two's complement

- Find one's complement
- Add 1
- Example (-6)
  - First represent 6 in binary format (00000110)
  - One's complement (11111001)
  - Two's complement (11111010)

Computer Number Systems

Complements

- They are used to simplify the subtraction operation
- Two types (for each *base-r* system)
  - Diminishing radix complement (r-1 complement)
  - Radix complement (r complement)

Computer Number Systems

1's and 2's Complements

- 1's complement of 10111001
  - 11111111 – 10111001 = 01000110
  - Simply replace 1's and 0's
- 1's complement of 10100010
  - 01011101
- 2's complement of 10111001
  - 01000110 + 1 = 01000111
  - Add 1 to 1's complement
- 2's complement of 10100010
  - 01011101 + 1 = 01011110

Computer Number Systems

EXAMPLE 1.7

Given the two binary numbers $X = 1010100$ and $Y = 1000011$, perform the subtraction **(a)** $X - Y$ and **(b)** $Y - X$ by using 2's complements.

$$
\begin{array}{lrr}
\text{(a)} & X = & 1010100 \\
& \text{2's complement of } Y = & +\ \underline{0111101} \\
& \text{Sum} = & 10010001 \\
& \text{Discard end carry } 2^7 = & -\ \underline{10000000} \\
& \textit{Answer: } X - Y = & 0010001
\end{array}
$$

Take care (there is a **carry**)

Computer Number Systems

EXAMPLE 1.7

Given the two binary numbers $X = 1010100$ and $Y = 1000011$, perform the subtraction **(a)** $X - Y$ and **(b)** $Y - X$ by using 2's complements.

(b)

$$\begin{aligned} Y &= \quad 1000011 \\ \text{2's complement of } X &= + \ \underline{0101100} \\ \text{Sum} &= \quad 1101111 \end{aligned}$$

Take care (there is **no carry**)

There is no end carry. Therefore, the answer is $Y - X = -(\text{2's complement of } 1101111) = -0010001$.

Computer Number Systems

EXAMPLE 1.8

Repeat Example 1.7, but this time using 1's complement.

(a) $X - Y = 1010100 - 1000011$

$$
\begin{aligned}
X &= 1010100 \\
\text{1's complement of } Y &= +\ \underline{0111100} \\
\text{Sum} &= 10010000 \\
\text{End-around carry} &= +\ \underline{1} \\
\textit{Answer}: X - Y &= 0010001
\end{aligned}
$$

Computer Number Systems

EXAMPLE 1.8

Repeat Example 1.7, but this time using 1's complement.

(a) $X - Y = 1010100 - 1000011$

(b) $Y - X = 1000011 - 1010100$

$$
\begin{array}{rr}
Y = & 1000011 \\
\text{1's complement of } X = & +\ \underline{0101011} \\
\text{Sum} = & 1101110
\end{array}
$$

Take care (there is **no carry**)

There is no end carry. Therefore, the answer is $Y - X = -(\text{1's complement of } 1101110) = -0010001$.

Computer Number Systems

Signed Binary Numbers

Signed Binary Numbers

Decimal	Signed-2's Complement	Signed-1's Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
−0	—	1111	1000
−1	1111	1110	1001
−2	1110	1101	1010
−3	1101	1100	1011
−4	1100	1011	1100
−5	1011	1010	1101
−6	1010	1001	1110
−7	1001	1000	1111
−8	1000	—	—

The Foundations: Logic and Proofs

Thank you !