Lecture 2 – Digital Systems and Binary Numbers – Part I

Rami Abielmona
University of Ottawa
January 16, 2015
ITI 1100 B
Digital Systems I

Presentation Outline

- Numbering Systems
- Positional vs. Polynomial Notation
- Decimal vs. Binary vs. Octal vs. Hex
- Binary Addition and Subtraction
- Binary Multiplication and Division
- Key terms

Digital Systems

→ Early computers were designed to perform numeric computations

→ They used ***discrete*** elements of information named digits (finite sets)

→DIGITAL SYSTEMS: manipulate ***discrete*** elements of information

such as the 10 decimal digits or the 26 letters of the alphabet

we live in the "Digital Age"!

Binary System and Logic Circuits

- What kind of data do computers work with?
  - Deep down inside, it's all 1s and 0s
- What can you do with 1s and 0s?
  - Boolean algebra operations
  - These operations map directly to hardware circuits (logic circuits)

Different Numbering Systems

- Decimal (Arabic): (0,1,2,3,4,5,6,7,8,9):
  Example: **$(452968)_{10}$**
- Octal: (0,1,2,3,4,5,6,7):
  Example **$(4073)_{8}$**
- Hexadecimal(0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F)
  Example: $(2BF3)_{16}$
- Binary: (0,1):
  Example: 1001110001011

Base in Numbering systems

- The decimal numbering system uses **base 10**. The values of the positions are calculated by taking 10 to some power.

1	6	2	.	3	7	5	Digits
100	10	1		1/10	1/100	1/1000	Weights

1	6	2	.	3	7	5	Digits
10^2	10^1	10^0		10^{-1}	10^{-2}	10^{-3}	Weights

- Base 10 for decimal numbers?

  It uses 10 digits: The digits 0 through 9.

Base in Numbering systems [2]

- The binary numbering system is called binary because it uses **base 2**. The values of the positions are calculated by taking 2 to some power.
- Base 2 for binary numbers :

  It uses 2 digits. The digits 0 and 1.

Representation of Numbers

→ ***There are two possible ways of writing a number in a given system:***

1- Positional Notation

2- Polynomial Representation

Positional Notation

$$N = (a_{n-1}a_{n-2} \ldots a_1a_0 \,.\, a_{-1}a_{-2} \ldots a_{-m})_r$$

Where

. = radix point

r = radix or base

n = number of integer digits to the left of the radix point

m = number of fractional digits to the right of the radix point

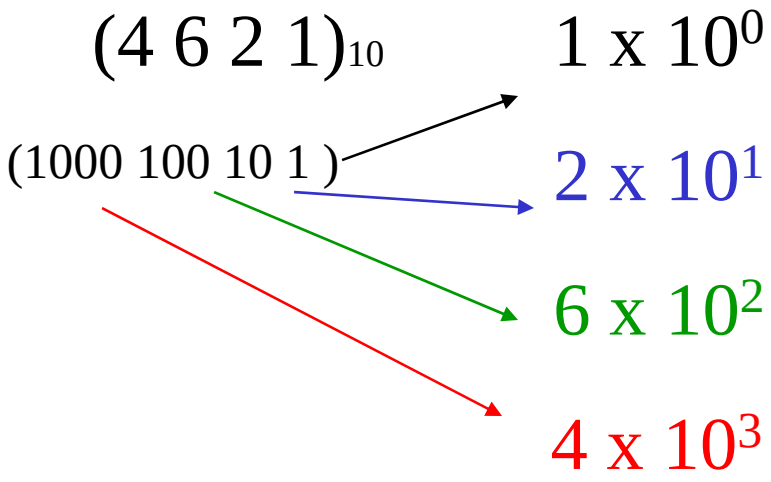
a_{n-1} = most significant digit (MSD)

a_{-m} = least significant digit (LSD)

Positional Notation

The Decimal Numbering System

- The decimal numbering system is a positional number system.
- Example:



Positional Notation

Binary Numbering System

- The Binary Numbering System is also a positional numbering system.
- Instead of using ten digits, 0 - 9, the binary system uses only two digits, the 0 and the 1.

→Example of a binary number & the values of the positions.

			1	0	1	0	1	0	1
			2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	1	0	1	.	0	1	Binary digits, or bits
2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	Weights (in base 10)

Polynomial Notation

$$N = a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \ldots + a_0 \times r^0 + a_{-1} \times r^{-1} \ldots + a_{-m} \times r^{-m}$$

$$= \sum_{i=-m}^{n-1} a_i r^i$$

Example:

Positional (N) ***Polynomial (N)***

$$N = (651.45)_{10} = 6 \times 10^2 + 5 \times 10^1 + 1 \times 10^0$$

$$+ 4 \times 10^{-1} + 5 \times 10^{-2}$$

Important number systems

→ There are three important number systems

- Binary Number System
- Octal Number System
- Hexadecimal Number System

Binary numbers

Digits = {0, 1}

Positional Polynomial

$(11010.11)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$

- **1 K (kilo) = 2^{10} = 1,024**
- **1M (mega) = 2^{20} = 1,048,576**
- **1G (giga) = 2^{30} = 1,073,741,824**

Converting Decimal to Binary

$$N = (a_{n-1}a_{n-2} \ldots a_1a_0 . a_{-1}a_{-2} \ldots a_{-m})_r$$

Integer ← → *Fractional*

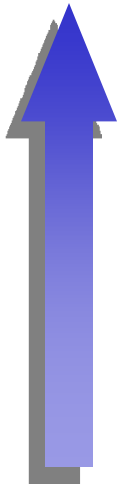
Radix point

→ **Integer** part and **Fractional** part are **converted differently**

Converting the Integer Part

- keep dividing by 2 until the quotient is 0. Collect the remainders *in reverse order*.
- Example: $(162)_{10}$.

```
162 / 2  = 81   rem 0
 81 / 2  = 40   rem 1
 40 / 2  = 20   rem 0
 20 / 2  = 10   rem 0
 10 / 2  = 5    rem 0
  5 / 2  = 2    rem 1
  2 / 2  = 1    rem 0
  1 / 2  = 0    rem 1
```



- Then $(162)_{10} = (10100010)_2$

Converting the Fraction Part

→ keep multiplying the *fractional part* by 2 until it becomes 0. Collect the integer parts (in forward order).

– However this may not terminate!

–Example: $(0.375)_{10}$

0.375 x 2 = 0.750
0.750 x 2 = 1.500
0.500 x 2 = 1.000

→ So, $(.375)_{10} = (.011)_2$

And $(162.375)_{10} = (10100010.011)_2$

Why does this work?

- This method can be applied to convert from decimal to *any* base
- Try converting 162.375 from decimal to decimal.

```
162 / 10  = 16 rem 2
 16 / 10  = 1  rem 6
  1 / 10  = 0  rem 1
```

- Each division "strips off" the rightmost digit (the remainder). The quotient represents the remaining digits in the number.

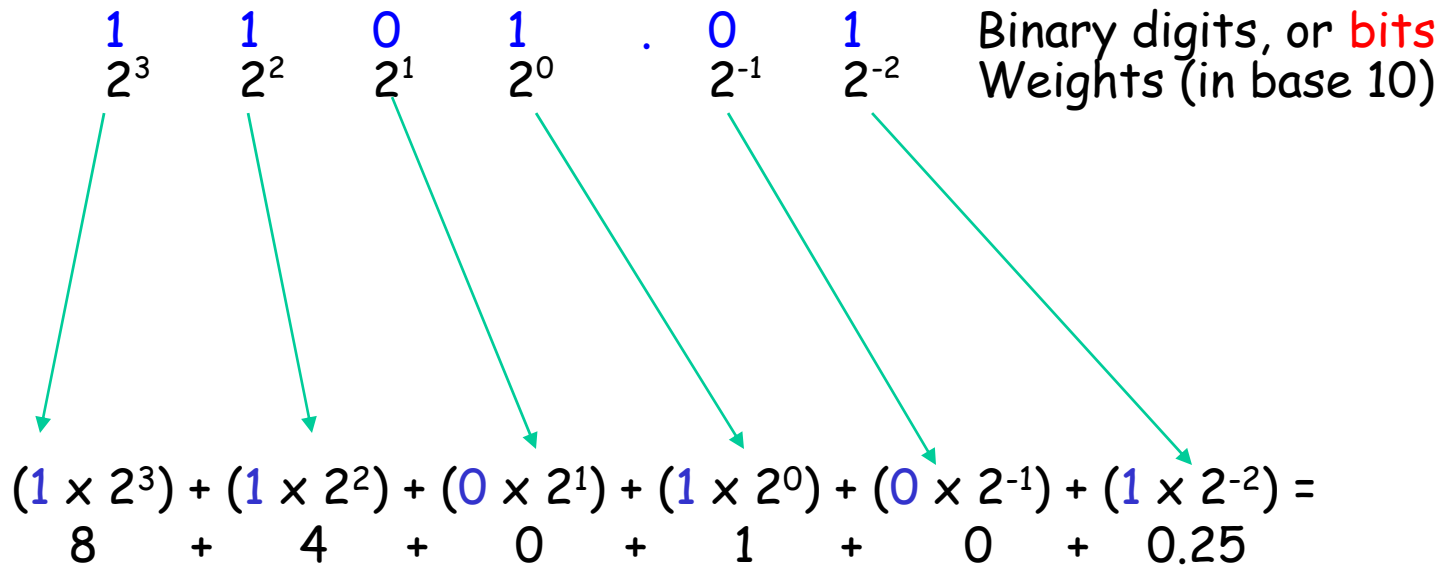
Why does this work? [2]

$0.375 \times 10 = 3.750$
$0.750 \times 10 = 7.500$
$0.500 \times 10 = 5.000$

- Each multiplication "strips off" the leftmost digit (the integer part). The fraction represents the remaining digits.

Converting binary to decimal

- To convert binary, or base 2, numbers to decimal we first obtain the polynomial representation of the number, then sum the products.
  - Example: $(1101.01)_2$



→ The decimal value is: $(13.25)_{10}$

Octal number system

–Digits = {0, 1, 2, 3, 4, 5, 6, 7}

Positional = Polynomial

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

- **Octal (base 8) digits range from 0 to 7.**

Since $8 = 2^3$, one octal digit is equivalent to 3 binary digits.

Converting Decimal to Octal

→ Integer part: keep dividing by 8 until the quotient is 0. Collect the remainders *in reverse order*.

→ Fractional Part: keep multiplying the *fractional part* by 8 until it becomes 0. Collect the integer parts (in forward order).

Same method as for the decimal to binary

Converting Octal to Decimal

- To convert Octal, or base 8, numbers to decimal we first obtain the polynomial representation of the number, then sum the products.

Example

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

Converting Binary to Octal

•To convert from binary to octal, make groups of 3 bits, starting from the binary point. Add 0s to the ends of the number if needed. Then convert each group of bits to its corresponding octal digit.

Example

$(10110100.001011)_2$ =(010 110 100 . 001 011 $)_2$

= (2 6 4 . 1 3 $)_8$

Converting Octal to Binary

- To convert from octal to binary, replace each Octal digit with its equivalent 3-bit binary sequence.
- Example

$$(261.35)_8 = (2\ \ 6\ \ 1\ .\ 3\ \ 5)_8$$
$$= (010\ \ 110\ \ 001\ .\ 011\ \ 101)_2$$

Hexadecimal numbers

-Digits = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Positional Polynomial

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

• *Hexadecimal* (base 16) digits are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. Since $16 = 2^4$, one hexa digit is equivalent to 4 binary digits.

– It's often easier to work with a number like B5 instead of 10110101.

Converting Decimal to Hexadecimal

→ **Integer part**: keep dividing by 16 until the quotient is 0. Collect the remainders *in reverse order*.

→ **Fractional Part**: keep multiplying the *fractional part* by 16 until it becomes 0. Collect the integer parts (in forward order).

Same method as for the decimal to binary conversion

Converting Hexadecimal to Decimal

• To convert Hexadecimal, or base 16, numbers to decimal, first obtain the polynomial representation of the number, then sum the products.

Example

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

Converting Hexadecimal to Binary

- To convert from hexadecimal to binary, replace each hex digit with its equivalent 4-bit binary sequence.
- Example

261.35_{16} = (2 6 1 . 3 5)$_{16}$

= (0010 0110 0001 . 0011 0101)$_2$

Converting Binary to Hexadecimal

- To convert from binary to hex, make groups of 4 bits, starting from the binary point. Add 0s to the ends of the number if needed. Then convert each group of bits to its corresponding hex digit.
- Example

$(10110100.001011)_2 = (1011\ 0100.\ 0010\ 1100)_2$

$= (B\ 4\ .\ 2\ C)_{16}$

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

Binary Addition

	0	0	1	1
	+0	+ 1	+0	+1
	--	--	--	--
sum	0	1	1	10

(sum of 0 and carryover of 1)

Examples:

1 0 0 1	0 0 0 1	1 1 0 0
+ 0 1 1 0	+ 1 0 0 1	+ 0 1 0 1
1 1 1 1	1 0 1 0	1 0 0 0 1

carryover

Binary Addition-Examples

Carry 1 1 1 1 0 1

1 0 1 1 0 1

\+ 0 1 1 1 0 1

Sum 1 0 0 1 0 1 0

Check your work

$(45)_{10}$

\+ $(29)_{10}$

= 74

$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$= 32 + 0 + 8 + 4 + 1 = 45$

Binary Addition- Examples

Addition of three Binary Digits

x	y	CarryIn	Sum	CarryOut
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Addition of Large Binary Numbers

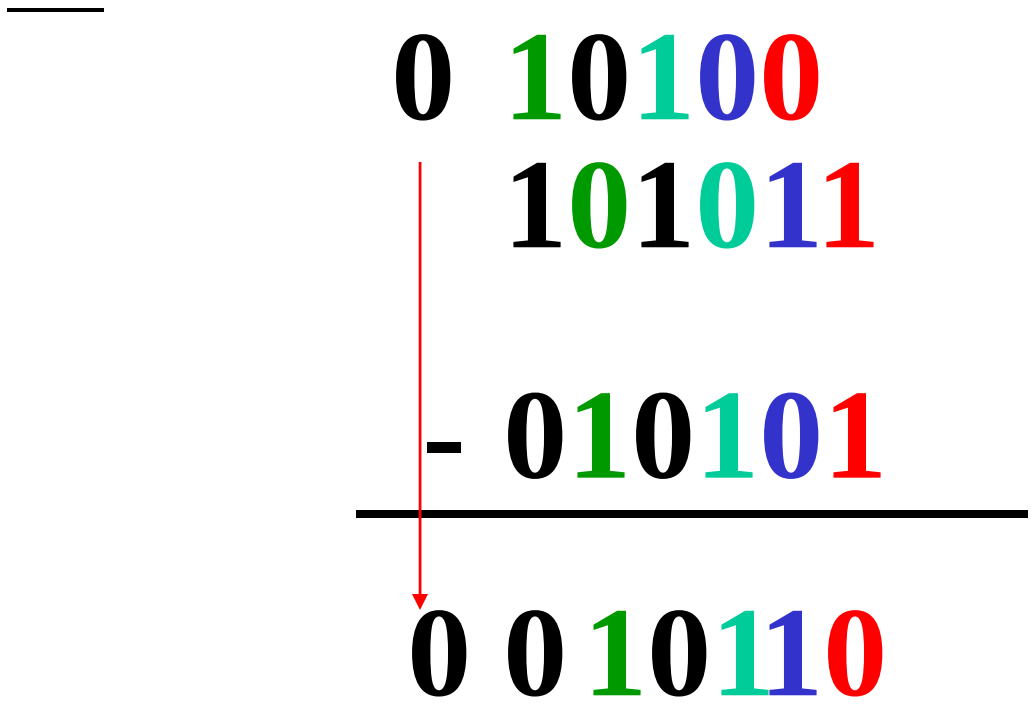
- Example showing larger numbers:

```
   1 0 1 0   0 0 1 1   1 0 1 1   0 0 0 1
 + 0 1 1 1   0 1 0 0   0 0 0 1   1 0 0 1
 ---------------------------------------
1  0 0 0 1   0 1 1 1   1 1 0 0   1 0 1 0
```

Binary Subtraction

	0	0	1	1
	- 0	- 1	- 0	- 1
	--	--	--	--
difference	0	11	1	0

(Diff. of 1 and carryover of 1)



Binary Multiplication

0	0	1	1
x 0	x 1	x 0	x 1
--	--	--	--
Product 0	0	0	1

A 0 010000.010

x B 0 001000.010

0000000000
0010000010
0000000000
0000000000
0000000000
0000000000
0010000010

1000110.000100

Binary Division

	0		**1**	
	÷ 1		**÷ 1**	
=	**0**		**1**	

Key Terms and Review Points

- Binary Addition
- Binary Division
- Binary Multiplication
- Binary Subtraction
- Binary Numbering System
- Conversion between Numbering Systems
- Decimal Numbering System
- Fractional Part
- Integer Part
- Hexadecimal Numbering System
- Numbering System
- Octal Numbering System
- Polynomial Notation
- Positional Notation

- Main Reference: Dr. Ahmed Karmouch ITI 1100 Slides