CMPE 212

Principles of Digital Design

Lecture 2

Number Systems

January 27, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture’s Overview

Previous Lecture:

- Von Neumann and System Bus models.
- Different levels of abstractions
  (programmer’s view, operating system view, hardware designer, etc.)
- Some historical perspective
- Digital versus analog systems
  (concepts covered in this course goes beyond computers)

This Lecture:

- Different representations of numbers
- Arithmetic in different radix
- Converting numbers between bases (whole and fractions)

Weighted Position Code

- Number systems: Ordered set of symbols or digits, with defined relations for $+ - \times \div$
- The base, or radix of a number system defines the number of symbols, i.e., the range of possible values that a digit may have: 0 – 9 for decimal; 0,1 for binary; 0 - 7 for octal; 0-4 base 5
- The general form for determining the decimal value of a number is given by:

$$Value = \sum_{i=-m}^{n-1} b_i . k^i$$

Example:

$541.25_{10} = 5 \times 10^2 + 4 \times 10^1 + 1 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}$

$= (500)_{10} + (40)_{10} + (1)_{10} + (2/10)_{10} + (5/100)_{10}$

$= (541.25)_{10}$

* Slide is courtesy of M. Murdocca and V. Heuring

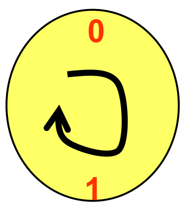
Addition in Different Bases

+	0	1	2	3	4
0	0	1	2	3	4
1	1	2	3	4	10
2	2	3	4	10	11
3	3	4	10	11	12
4	4	10	11	12	13

Base 5

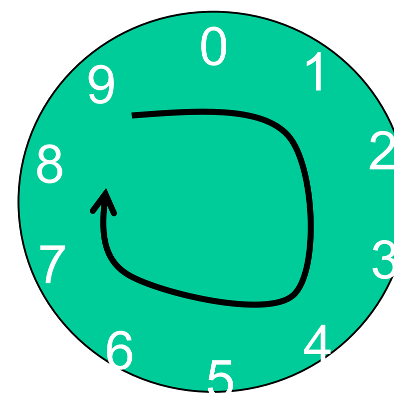
+	0	1
0	0	1
1	1	10

Binary



+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13
5	5	6	7	8	9	10	11	12	13	14
6	6	7	8	9	10	11	12	13	14	15
7	7	8	9	10	11	12	13	14	15	16
8	8	9	10	11	12	13	14	15	16	17
9	9	10	11	12	13	14	15	16	17	18

Decimal (Base 10)



- Addition table enlists the results of adding two one-digit numbers
- Analogous to decimal, the addition table for other bases
- Carry out reflects a complete cycle

Binary Addition

Carry in	0	0	1	0	1	1	1	0
Operands	0	0	0	1	0	0	1	1
	+ 0	+ 1	+ 0	+ 1	+ 0	+ 1	+ 0	+ 1
	0 0	0 1	0 1	1 0	0 1	1 0	1 0	1 0

Carry out Sum

+	0	1
0	0	1
1	1	10

Example:

Carry		1	1	1	1	0	0	0	0	
Addend: *A*		0	1	1	1	1	1	0	0	$(124)_{10}$
Augend: *B*	+	0	1	0	1	1	0	1	0	$(90)_{10}$
Sum		1	1	0	1	0	1	1	0	$(214)_{10}$

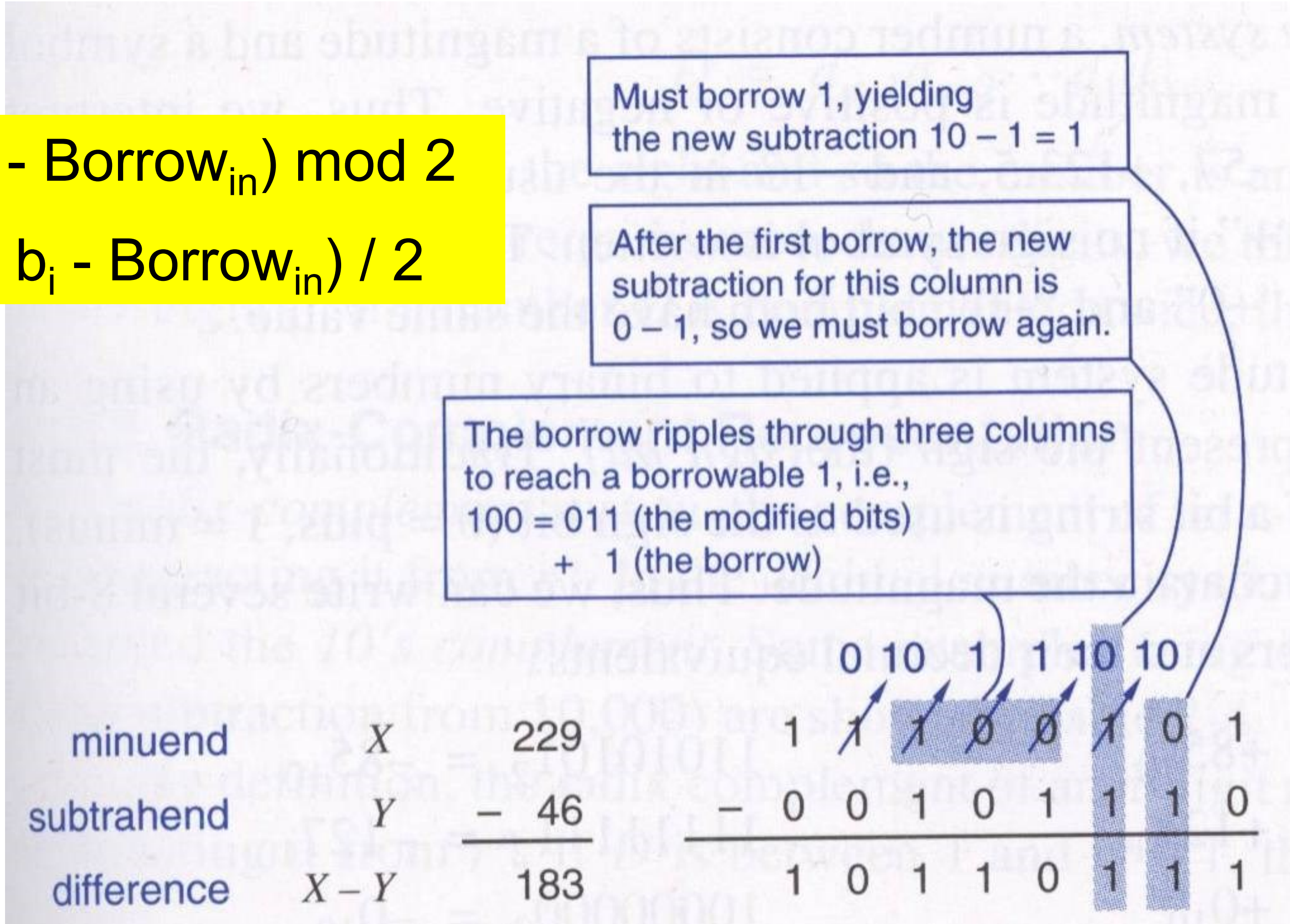
$Sum_i = (a_i + b_i + Carry_{in}) \bmod 2$, $\quad Carry_{out} = (a_i + b_i + Carry_{in}) / 2$

* Slide is (partially) courtesy of M. Murdocca and V. Heuring

Binary Subtraction

$$\text{Result}_i = (a_i - b_i - \text{Borrow}_{in}) \bmod 2$$

$$\text{Borrow}_{out} = (a_i - b_i - \text{Borrow}_{in}) / 2$$



-	0	1
0	0	1
1	11	0

borrow

- Representation of negative numbers covert subtraction to be addition
- Similarity of the equations enables the use of the same circuit to add and subtract

Signed Fixed Point Numbers

- For an 8-bit number, there are $2^8 = 256$ possible bit patterns. These bit patterns can represent negative numbers if we choose to assign bit patterns to numbers in this way. We can assign half of the bit patterns to negative numbers and half of the bit patterns to positive numbers.
- Four signed representations we will cover are:
  - Signed Magnitude
  - Radix complement (e.g., Two's Complement for binary)
  - Diminished radix complement (e.g., One's complement for binary)
  - Excess-M (Biased)

* Slide is (partially) courtesy of M. Murdocca and V. Heuring

Signed Magnitude

- Also known as "sign and magnitude," the leftmost bit is the sign (0 = positive, 1 = negative) and the remaining bits are the magnitude.
- Example:

  $+25_{10} = 00011001_2$, $-25_{10} = 10011001_2$

- Two representations for zero:

  $+0 = 00000000_2$, $-0 = 10000000_2$

- Using an 8-bit representation:
  - Largest number is $+127_{10}$, smallest number is -127_{10}
- Using an N-bit representation:
  - Largest number is $+(2^{N-1}-1)_{10}$, smallest number is $-(2^{N-1}-1)_{10}$

* Slide is courtesy of M. Murdocca and V. Heuring

Unsigned Multiplication

- Multiplication of two 4-bit unsigned binary integers produces an 8-bit result

				1	1	0	1		$(13)_{10}$	Multiplicand M
			×	1	0	1	1		$(11)_{10}$	Multiplier Q
				1	1	0	1	Partial products		
			1	1	0	1				
		0	0	0	0					
	1	1	0	1						
1	0	0	0	1	1	1	1		$(143)_{10}$	Product P

- Multiplication of two 4-bit signed binary integers produces only a 7-bit result (each operand reduces to a sign bit and a 3-bit magnitude, producing a sign-bit and a 6-bit result)

* Slide is courtesy of M. Murdocca and V. Heuring

Example of Base 2 Division

- $(7 / 3 = 2)_{10}$ with a remainder R of 1.
- Equivalently, $(0111/ 11 = 10)_2$ with a remainder R of 1.

$$
\begin{array}{r|l}
 & 0\,0\,1\,0\ \ R\,1 \\
\cline{2-2}
1\,1 & 0\,1\,1\,1 \\
 & \ \ \ 1\,1 \\
\cline{2-2}
 & \ \ \ \ \ 0\,1
\end{array}
$$

Base Conversion

Integer Part

- If source radix S < target radix T
  - ➢ Evaluate the polynomial representation of the number in the target number system, e.g., $(32)_4 = 3 \times 4^1 + 2 \times 4^0 = (14)_{10}$
- If source radix S > Target radix T

$(N)_S = d_{n-1}S^{n-1} + \ldots + d_1S^1 + d_0S^0$
$\quad = (M)_T = x_{m-1}T^{m-1} + \ldots + x_1T^1 + x_0T^0$

Apply Reminder Method in order to determine the digits $x_{m-1}, \ldots, x_1, x_0$

$(N)_S = (M)_T = (x_{m-1}T^{m-2} + \ldots + x_1)T^1 + x_0T^0$

$\Rightarrow x_0 = (N)_S$ modulo T, $\quad Q1 = (N)_S / T = x_{m-1}T^{m-2} + \ldots + x_1$

$\Rightarrow x_1 = Q1$ modulo T, $\quad Q2 = Q1 / T = x_{m-1}T^{m-2} + \ldots + x_2$

… and so on

Fraction Part

$F = d_{-1}S^{-1} + d_{-2}S^{-2} + \ldots \quad = x_{-1}T^{-1} + x_{-2}T^{-2} + \ldots$

Repetitively multiply by T (Multiplication Method) the fraction part

$F \times T = x_{-1} + x_{-2}T^{-1} + \ldots \Rightarrow x_{-1} = \lfloor F \times T \rfloor$, which is the integer part

Base Conversion: Remainder Method

- **Example:** Convert 23.375_{10} to base 2.

➢ Start by converting the integer portion:

		Integer	Remainder	
23/2	=	11	1	← Least significant bit
11/2	=	5	1	
5/2	=	2	1	
2/2	=	1	0	
1/2	=	0	1	← Most significant bit

$$(23)_{10} = (10111)_2$$

* Slide is courtesy of M. Murdocca and V. Heuring

Base Conversion: Multiplication Method

- Now, convert the fraction:

Most significant bit

$$.375 \times 2 = 0.75$$

$$.75 \times 2 = 1.5$$

$$.5 \times 2 = 1.0$$

Least significant bit

$$(.375)_{10} = (.011)_2$$

- **Putting it all together, $23.375_{10} = 10111.011_2$.**

* Slide is courtesy of M. Murdocca and V. Heuring

Non-terminating Base 2 Fraction

We can't always convert a terminating base 10 fraction into an equivalent terminating base 2 fraction:

$$.2 \times 2 = 0.4$$

$$.4 \times 2 = 0.8$$

$$.8 \times 2 = 1.6$$

$$.6 \times 2 = 1.2$$

Cycle back

$$.2 \times 2 = 0.4$$

$$\vdots$$

* Slide is courtesy of M. Murdocca and V. Heuring

Base 2, 8, 10, 16 Number Systems

Example:

Show a column for ternary (base 3). As an extension of that, convert 14_{10} to base 3, using 3 as the divisor for the remainder method (instead of 2).

Result is 112_3

Binary (base 2)	Octal (base 8)	Decimal (base 10)	Hexadecimal (base 16)
0	0	0	0
1	1	1	1
10	2	2	2
11	3	3	3
100	4	4	4
101	5	5	5
110	6	6	6
111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

* Slide is courtesy of M. Murdocca and V. Heuring

More on Base Conversions

- Converting among power-of-2 bases is particularly simple:

$1011_2 = (10_2)(11_2) = 23_4$

$23_4 = (2_4)(3_4) = (10_2)(11_2) = 1011_2$

$101010_2 = (101_2)(010_2) = 52_8$

$01101101_2 = (0110_2)(1101_2) = 6D_{16}$

- How many bits should be used for each base 4, 8, etc., digit?

For base 2, in which $2 = 2^1$, the exponent is 1 and so one bit is used for each base 2 digit. For base 4, in which $4 = 2^2$, the exponent is 2, so two bits are used for each base 4 digit. Likewise, for base 8 and base 16, $8 = 2^3$ and $16 = 2^4$, and so 3 bits and 4 bits are used for base 8 and base 16 digits, respectively.

* Slide is courtesy of M. Murdocca and V. Heuring

Conclusion

❑ *Summary*

- ➔ Different representations of numbers
  (Radix, Weighted Position Code, Binary, octal and hexadecimal)
- ➔ Arithmetic in different radix
- ➔ Converting numbers between bases
  (multiplication and remainder conversion method, bit grouping)
- ➔ Converting integer and fractions

❑ *Next Lecture*

- ➔ Representation of signed fixed numbers
  - ➢ Signed Magnitude
  - ➢ Diminished radix Complement
  - ➢ Excess (Biased)
  - ➢ Radix Complement

Reading assignment: sections 1.1 - 1.3, in the textbook