ECE-332:437
DIGITAL SYSTEMS DESIGN (DSD)

Fall 2016 – Lecture 2

Nagi Naganathan
September 15, 2016

Announcements/Comments – September 15, 2016

- Is Sakai accessible?
  - Last week's (9/8/16) slides posted (Intro, Lecture 1)
- HW 1 Posted today
  - Due on 9/22/16
- Lab Sessions formed
  - Any issues/concerns
- Swapnil will do the class lecture on 9/29 (Work Travel)
- Guest Lecture (SystemVerilog, Verification) – 10/6

- Questions ??

Topics Covered on – September 8, 2016

- Intro Slides
- Lecture #1
  - Basic of digital logic
  - Digital systems surround us
  - Digital systems use 0s and 1s
  - Encoding analog signals to digital can provide many benefits
    - e.g., audio—higher-quality storage/transmission, compression, etc.
  - Encoding integers as 0s and 1s: Binary numbers
  - Boolean Gates (AND, OR, NOT)

Topics to cover today – September 15, 2016

- Lecture 2 (Parts 1, 2)
  - Basics of Number Systems
  - Representation of negative numbers
  - 2’s complement
  - Binary addition, subtraction
- Lecture 3
  - Logic Gates
  - Boolean Algebra

Part 1
Basics of Binary System

Roman Numerals

- Roman numerals:

I	1
II	2
III	3
IV	4
V	5
X	10
XVII	17
L	50
C	100
D	500
M	1000
…	
MMVI	2006

Number Systems – Examples

		General	Decimal	Binary
Radix (Base)		r	10	2
Digits		0 => r - 1	0 => 9	0 => 1
	0	r^0	1	1
	1	r^1	10	2
	2	r^2	100	4
	3	r^3	1000	8
Powers of	4	r^4	10,000	16
Radix	5	r^5	100,000	32
	-1	r^{-1}	0.1	0.5
	-2	r^{-2}	0.01	0.25
	-3	r^{-3}	0.001	0.125
	-4	r^{-4}	0.0001	0.0625
	-5	r^{-5}	0.00001	0.03125

Common Number System – Examples

Decimal system: $r = 10$, ten digits: 0…9

Binary system: $r = 2$, two digits: 0 or 1

$$\ldots x_3 x_2 x_1 x_0 \ldots = \qquad \ldots x_3 2^3 + x_2 2^2 + x_1 2^1 + x_0 2^0 \ldots$$

$$1001_2$$
$$= 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 8 + 0 + 0 + 1$$
$$= 9$$

Common Number System

Decimal system: $r = 10$, ten digits: 0…9

Binary system: $r = 2$, two digits: 0 or 1 (bit)

Octal system: $r = 8$, eight digits: 0…7

Hexadecimal system: $r = 16$, 16 digits: 0…9A…F

$A_{16} = 10_{10}$,

$B_{16} = 11_{10}$,

$C_{16} = 12_{10}$,

$D_{16} = 13_{10}$,

$E_{16} = 14_{10}$,

$F_{16} = 15_{10}$,

Arabic System

- Arabic is a type of positional system
  - The value a digit represents depends on its position
    - In Roman system, X is always 10. So XX = 20.
- Ten digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- A digit's value depends on its position

$$1712 = 1000 + 700 + 10 + 2$$

$$\ldots x_3 x_2 x_1 x_0 . x_{-1} \ldots =$$
$$\ldots x_3 10^3 + x_2 10^2 + x_1 10^1 + x_0 10^0 + x_{-1} 10^{-1} \ldots$$

Here, the base is 10. We have ten digits,

The value of a digit increases 10 times per position when it is moved to the left.

NUMBER SYSTEMS – Representation

- Positive radix, positional number systems
- A number with *radix* $\boldsymbol{r}$ is represented by a string of digits:

$$A_{n-1}A_{n-2}\ \ldots\ A_1A_0\ .\ A_{-1}\ A_{-2}\ \ldots\ A_{-m+1}\ A_{-m}$$

in which $\mathbf{0} \le A_i < \boldsymbol{r}$ and **.** is the *radix point*.

- The string of digits represents the power series:

$$(\text{Number})_r = \left(\sum_{i=0}^{i=n-1} A_i \cdot r^i\right) + \left(\sum_{j=-m}^{j=-1} A_j \cdot r^j\right)$$

(Integer Portion) + (Fraction Portion)

Powers of Two

- $2^0 =$
- $2^1 =$
- $2^2 =$
- $2^3 =$
- $2^4 =$
- $2^5 =$
- $2^6 =$
- $2^7 =$
- $2^8 =$
- $2^9 =$
- $2^{10} =$
- $2^{11} =$
- $2^{12} =$
- $2^{13} =$
- $2^{14} =$
- $2^{15} =$

Powers of Two

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$
- Handy to memorize up to 2^9

Conversion from Decimal to Binary

- Convert decimal 41 to binary
  - 41/2 = 20 +1/2   Remainder =1 (Least Significant Digit)
  - 20/2 = 10   = 0
  - 10/2 = 5   = 0
  - 5/2 = 2+1/2   =1
  - 2/2 =1   = 0
  - ½ = 0+1/2   1 (Most Significant Digit)

- Decimal 41 = (101001)

Number Conversion

- Decimal to binary conversion:
  - Convert 10101_2 to decimal

- Decimal to binary conversion:
  - Convert 47_{10} to binary

Number Conversion

- Decimal to binary conversion:
  - Convert 10011_2 to decimal
  - $16\times1 + 8\times0 + 4\times0 + 2\times1 + 1\times1 = 19_{10}$

- Decimal to binary conversion:
  - Convert 47_{10} to binary
  - $32\times1 + 16\times0 + 8\times1 + 4\times1 + 2\times1 + 1\times1 = 101111_2$

1000's column
100's column
10's column
1's column

$$9742_{10} = \underbrace{9 \times 10^3}_{\text{nine thousands}} + \underbrace{7 \times 10^2}_{\text{seven hundreds}} + \underbrace{4 \times 10^1}_{\text{four tens}} + \underbrace{2 \times 10^0}_{\text{two ones}}$$

Figure 1.4 Representation of a decimal number

1's column
2's column
4's column
8's column
16's column

$$10110_2 = \underbrace{1 \times 2^4}_{\text{one sixteen}} + \underbrace{0 \times 2^3}_{\text{no eight}} + \underbrace{1 \times 2^2}_{\text{one four}} + \underbrace{1 \times 2^1}_{\text{one two}} + \underbrace{0 \times 2^0}_{\text{no one}} = 22_{10}$$

Figure 1.5 Conversion of a binary number to decimal

$$\overset{\text{256's column}}{2}\overset{\text{16's column}}{E}\overset{\text{1's column}}{D}_{16} = \underset{\text{two two hundred fifty six's}}{2 \times 16^2} + \underset{\text{fourteen sixteens}}{E \times 16^1} + \underset{\text{thirteen ones}}{D \times 16^0} = 749_{10}$$

Figure 1.6 Conversion of a hexadecimal number to decimal

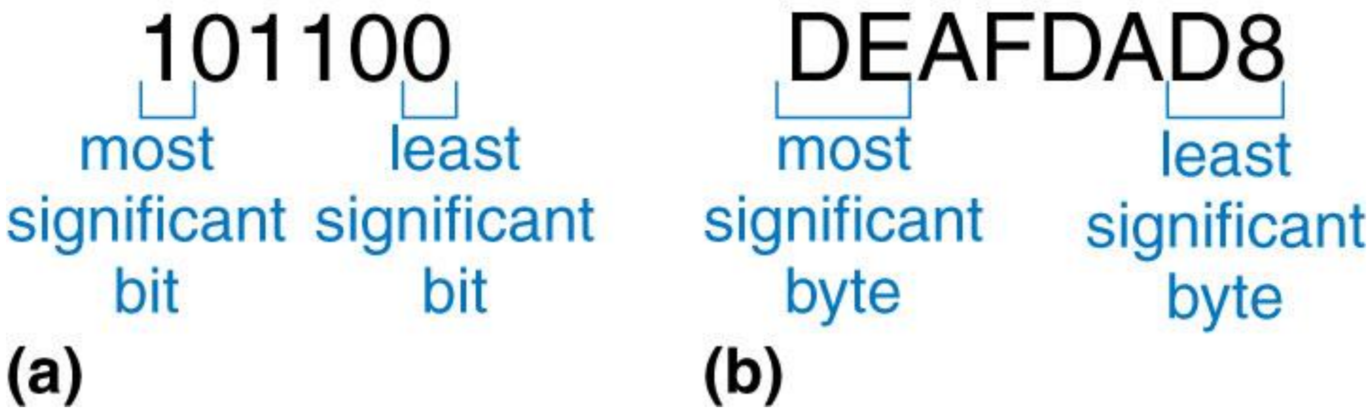

Figure 1.7 Least and most significant bits and bytes

Number Systems

- Decimal numbers

1000's column
100's column
10's column
1's column

$5374_{10} =$

- Binary numbers

8's column
4's column
2's column
1's column

$1101_2 =$

Number Systems

- Decimal numbers

1000's column
100's column
10's column
1's column

$$5374_{10} = \underbrace{5 \times 10^3}_{\text{five thousands}} + \underbrace{3 \times 10^2}_{\text{three hundreds}} + \underbrace{7 \times 10^1}_{\text{seven tens}} + \underbrace{4 \times 10^0}_{\text{four ones}}$$

- Binary numbers

8's column
4's column
2's column
1's column

$$1101_2 = \underbrace{1 \times 2^3}_{\text{one eight}} + \underbrace{1 \times 2^2}_{\text{one four}} + \underbrace{0 \times 2^1}_{\text{no two}} + \underbrace{1 \times 2^0}_{\text{one one}} = 13_{10}$$

Special Powers of 2

- 2^{10} (1024) is Kilo, denoted "K"
- 2^{20} (1,048,576) is Mega, denoted "M"
- 2^{30} (1,073, 741,824)is Giga, denoted "G"
- 2^{40} (1,099,511,627,776) is Tera, denoted “T"

Bits, Bytes, Nibbles…

- Bits

10010110

most significant bit

least significant bit

- Bytes & Nibbles

byte

10010110

nibble

- Bytes

CEBF9AD7

most significant byte

least significant byte

Hexadecimal Numbers

- Base 16
- Shorthand to write long binary numbers

Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

Hexadecimal Numbers

Hex Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Hexadecimal to Binary Conversion

- Hexadecimal to binary conversion:
  – Convert $4AF_{16}$ (also written 0x4AF) to binary

- Hexadecimal to decimal conversion:
  – Convert 0x4AF to decimal

Hexadecimal to Binary Conversion

- Hexadecimal to binary conversion:
  - Convert $4AF_{16}$ (also written 0x4AF) to binary
  - $0100\ 1010\ 1111_2$
- Hexadecimal to decimal conversion:
  - Convert $4AF_{16}$ to decimal
  - $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

Conversion to Decimal Numbers

You can convert numbers of any radix to decimal numbers

$(1001)_2$
$= (1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0)_{10}$
$= 8 + 0 + 0 + 1$
$= 9$

0xAB76
$= 10 \times 16^3 + 11 \times 16^2 + 7 \times 16 + 6$
$= 43894$

Binary to Hexadecimal

- Method 1:
  - Convert the number to a decimal number first

- Method 2:
  - Divide digits into groups of four
  - Convert each group to a hex digit

1110	0110	0001
E	6	1

Convert each digit to a 4-digit binary number

$(E61)_{16} = (1110\ 0110\ 0001)_2$

$(ABCD)_{16} = (1010101111001101)_2$

Part 2
Basics of Addition/Subtraction

Addition

- Decimal

$$\begin{array}{r} \color{blue}{11} \leftarrow \text{carries} \\ 3734 \\ +\ 5168 \\ \hline 8902 \end{array}$$

- Binary

$$\begin{array}{r} \color{blue}{11} \leftarrow \text{carries} \\ 1011 \\ +\ 0011 \\ \hline 1110 \end{array}$$

Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1001 \\ +\ 0101 \\ \hline \end{array}$$

- Add the following 4-bit binary numbers

$$\begin{array}{r} 1011 \\ +\ 0110 \\ \hline \end{array}$$

Binary Addition Examples

- Add the following 4-bit binary numbers

$$\begin{array}{r} \color{blue}{1} \\ 1001 \\ +\ \ 0101 \\ \hline \color{blue}{1110} \end{array}$$

- Add the following 4-bit binary numbers

$$\begin{array}{r} \color{blue}{111} \\ 1011 \\ +\ \ 0110 \\ \hline \color{blue}{10001} \end{array}$$

Overflow!

Overflow

- Digital systems operate on a fixed number of bits
- Addition overflows when the result is too big to fit in the available number of bits
- See previous example of 11 + 6

Signed Binary Numbers

- Sign/Magnitude Numbers
- Two's Complement Numbers

Sign/Magnitude Numbers

- 1 sign bit, *N*-1 magnitude bits
- Sign bit is the most significant (left-most) bit
  - Positive number: sign bit = 0
  - Negative number: sign bit = 1

$$A : \{a_{N-1}, a_{N-2}, \cdots a_2, a_1, a_0\}$$

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Example, 4-bit sign/mag representations of ± 6:

  +6 =

  - 6 =

- Range of an *N*-bit sign/magnitude number:

Sign/Magnitude Numbers

- 1 sign bit, *N*-1 magnitude bits
- Sign bit is the most significant (left-most) bit
  - Positive number: sign bit = 0
  - Negative number: sign bit = 1

$$A : \{a_{N-1}, a_{N-2}, \cdots a_2, a_1, a_0\}$$

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Example, 4-bit sign/mag representations of ± 6:

  +6 = **0110**

  - 6 = **1110**

- Range of an *N*-bit sign/magnitude number:

  $[-(2^{N-1}-1), 2^{N-1}-1]$

Sign/Magnitude Numbers

- Problems:
  - Addition doesn't work, for example -6 + 6:

$$
\begin{array}{r}
1110 \\
+\ 0110 \\
\hline
10100
\end{array}
$$

(wrong!)

  - Two representations of 0 (± 0):

1000

0000

Complement Numbers

- 1's complement –

  <u>One's complement</u>: One obtains 1's complement[4] of a binary number by subtracting the number from the binary number consisting of the same number of bits, with all 1's. This is the same as switching all of the bits (substituting every 0 by a 1 and vice-versa).

- Example 011 = 111 – 011 = 100
- 2's complement

Example: The 2's Complement of binary 101100 is :

$$\begin{array}{r} 010011 \\ +\quad 1 \\ \hline 010100 \end{array}$$

1's Complement

The 2's Complement can be obtained by leaving all least significant 0's and the first 1 unchanged and replacing 1's with 0's and 0's with 1's in all other significant digits.

Example:

The 2's Complement of 111001100 is: $\underline{000110}\,\underline{\mathbf{1}}\,\underline{\mathbf{00}}$

All other significant bits are changed

First 1 unchanged

Two least significant 0: unchanged

Two's Complement Numbers

- Don't have same problems as sign/magnitude numbers:
  - Addition works
  - Single representation for 0

"Taking the Two's Complement"

- Flip the sign of a two's complement number
- Method:
  1. Invert the bits
  2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$

"Taking the Two's Complement"

- Flip the sign of a two's complement number
- Method:
  1. Invert the bits
  2. Add 1
- Example: Flip the sign of $3_{10} = 0011_2$
  1. **1100**
  2. **+ 1**

1101 = -3_{10}

Two's Complement Numbers

- Same as unsigned binary, but the most significant bit (msb) has value of -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- Most positive 4-bit number:
- Most negative 4-bit number:
- The most significant bit still indicates the sign (1 = negative, 0 = positive)
- Range of an *N*-bit two's comp number:

Two's Complement Examples

- Take the two's complement of $6_{10} = 0110_2$

- What is the decimal value of 1001_2?

Two's Complement Numbers

- Same as unsigned binary, but the most significant bit (msb) has value of -2^{N-1}

$$A = a_{n-1}\left(-2^{n-1}\right) + \sum_{i=0}^{n-2} a_i 2^i$$

- Most positive 4-bit number: **0111**
- Most negative 4-bit number: **1000**
- The most significant bit still indicates the sign (1 = negative, 0 = positive)
- Range of an *N*-bit two's comp number:

$[-(2^{N-1}), 2^{N-1}-1]$

Two's Complement Examples

- Take the two's complement of $6_{10} = 0110_2$
  1. 1001
  2. $+ \quad 1$

  $1010_2 = -6_{10}$

- What is the decimal value of the two's complement number 1001_2?
  1. 0110
  2. $+ \quad 1$

  $0111_2 = 7_{10}$, so $1001_2 = -7_{10}$

Two's Complement Addition

- Add 6 + (-6) using two's complement numbers

$$\begin{array}{r} 0110 \\ +\ \ 1010 \\ \hline \end{array}$$

- Add -2 + 3 using two's complement numbers

$$\begin{array}{r} 1110 \\ +\ \ 0011 \\ \hline \end{array}$$

Two's Complement Addition

- Add 6 + (-6) using two's complement numbers

```
    111
     0110
  +  1010
  -------
    10000
```

- Add -2 + 3 using two's complement numbers

```
    111
     1110
  +  0011
  -------
    10001
```


ELSEVIER

Increasing Bit Width

- A value can be extended from N bits to M bits (where $M > N$) by using:
  - Sign-extension
  - Zero-extension

Sign-Extension

- Sign bit is copied into most significant bits.
- Number value remains the same.

- **Example 1:**
  - 4-bit representation of 3 = 0011
  - 8-bit sign-extended value: 00000011
- **Example 2:**
  - 4-bit representation of -5 = 1011
  - 8-bit sign-extended value: 11111011

Zero-Extension

- Zeros are copied into most significant bits.
- Value will change for negative numbers.

- **Example 1:**
  - 4-bit value = $0011_2 = 3_{10}$
  - 8-bit zero-extended value: $00000011 = 3_{10}$
- **Example 2:**
  - 4-bit value = $1011 = -5_{10}$
  - 8-bit zero-extended value: $00001011 = 11_{10}$