Digital Logic

CS6133 - Computer Architecture I

Vikram Padman

Polytechnic Institute of New York University

vikram.padman@nyu.edu

Agenda

1. **Numbering System**
   - Binary
   - Hexadecimal
   - Grouping Bits (Nibbles, Bytes, Word, Double Words)
   - Binary Addition
   - Signed Binary Numbers
2. **Logic Gates**
   - AND, OR, NOT, NAND and XOR
   - Propagation or Gate Delay
3. **Digital Discipline**
   - Supply Voltage & Logic Levels
   - Noise Margins
   - Reality - DC Transfer Characteristics
4. **Power Consumption**
5. **Activities**

Reading List
Week 2

1. "Digital Design and Computer Architecture", Chapter 1
2. "Computer Organization And Design", Chapter 1
3. "Computer Architecture - A Quantitative Approach", Chapter 1, section 1.4 and 1.5

Binary Numbers

- **Decimal Number**

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

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

- **Binary Number**

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

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

Binary Numbers

1-Bit Binary Numbers	2-Bit Binnary Numbers	3-Bit Binary Numbers	4-Bit Binary Numbers	Decimal Equivalents
0	00	000	0000	0
1	01	001	0001	1
	10	010	0010	2
	11	011	0011	3
		100	0100	4
		101	0101	5
		110	0110	6
		111	0111	7
			1000	8
			1001	9
			1010	10
			1011	11
			1100	12
			1101	13
			1110	14
			1111	15

Hexadecimal Numbers

- Used to represent long binary number or objects

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

Hexadecimal Numbers

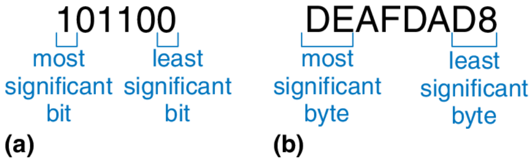
Hexadecimal Digit	Decimal Equivalent	Binary Equivalent
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Grouping Bits

Nibbles, Bytes, Word, Double Words

Bits are grouped together to form larger objects

- Nibble = 4 bits
- Byte = 8 bits
- Word = 16 bits
- Dword = 32 bits
- "B" = Byte & "b" = Bit



Binary Addition

$$
\begin{array}{rr}
 & 11 \\
 & 4277 \\
+ & 5499 \\
\hline
 & 9776
\end{array}
\qquad \leftarrow \text{carries} \rightarrow \qquad
\begin{array}{rr}
 & 11 \\
 & 1011 \\
+ & 0011 \\
\hline
 & 1110
\end{array}
$$

(a) **(b)**

$$
\begin{array}{rr}
 & 111 \\
 & 0111 \\
+ & 0101 \\
\hline
 & 1100
\end{array}
\qquad\qquad
\begin{array}{rr}
 & 11\ \ 1 \\
 & 1101 \\
+ & 0101 \\
\hline
 & 10010
\end{array}
$$

Signed Binary Numbers

How are *negative* numbers represented?

Signed Binary Numbers

How are *negative* numbers represented?

- There are multiple methods
  - **Sign/Magnitude Numbers**: Simple use the MSb as a sign bit.
    - If MSb = "1" the number is negative
    - If MSb = "0" the number is positive
    - Simple and intuitive, but has drawbacks: addition does not work and has multiple representation for "0"

Signed Binary Numbers

How are *negative* numbers represented?

- There are multiple methods
  - **Sign/Magnitude Numbers**: Simple use the MSb as a sign bit.
    - If MSb = "1" the number is negative
    - If MSb = "0" the number is positive
    - Simple and intuitive, but has drawbacks: addition does not work and has multiple representation for "0"
  - **2's Complement Number**: Overcome shortcomings of sign/magnitude
    - MSb = -2^{N-1} instead of 2^{N-1}
    - Largest positive number = 01111...111
    - Largest negative number = 10000...000
    - Converting positive to negative is simple: Invert and add one

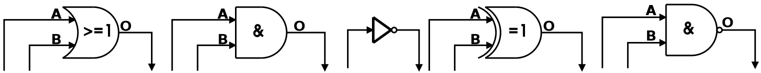
Range

System	Range
Unsigned	$[0, 2^N - 1]$
Sign/Magnitude	$[-2^{N-1} + 1, 2^{N-1} - 1]$
Two's Complement	$[-2^{N-1}, 2^{N-1} - 1]$

An Example

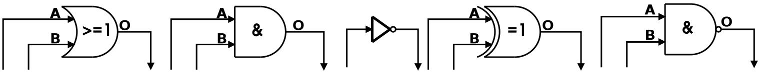
	–8	–7	–6	–5	–4	–3	–2	–1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Unsigned									0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Two's Complement	1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111								
Sign/Magnitude		1111	1110	1101	1100	1011	1010	1001	0000 1000	0001	0010	0011	0100	0101	0110	0111								

Logic Gates



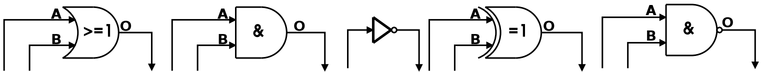
- Fundamental building blocks that are used in all digital electronic devices.

Logic Gates



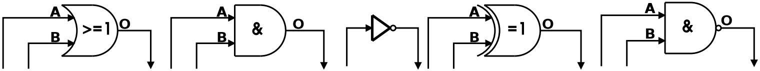
- Fundamental building blocks that are used in all digital electronic devices.
- Logic gates are used to build general purpose components (such as Adder / Subtractors, MUX / DEMUX ...etc)

Logic Gates



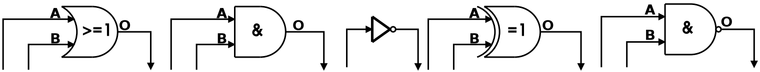
- Fundamental building blocks that are used in all digital electronic devices.
- Logic gates are used to build general purpose components (such as Adder / Subtractors, MUX / DEMUX ...etc)
  - General purpose components are then used to build larger, function specific, components such as arithmetic and logic unit, control unit and various type of memories.

Logic Gates



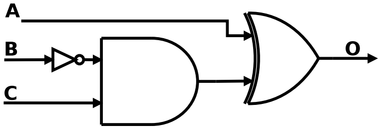
- Fundamental building blocks that are used in all digital electronic devices.
- Logic gates are used to build general purpose components (such as Adder / Subtractors, MUX / DEMUX ...etc)
  - General purpose components are then used to build larger, function specific, components such as arithmetic and logic unit, control unit and various type of memories.
  - Many function specific components are assembled together to make a microprocessor, I/O HUB, SRAM ... etc

Logic Gates

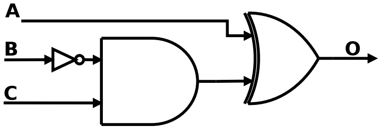


- Fundamental building blocks that are used in all digital electronic devices.
- Logic gates are used to build general purpose components (such as Adder / Subtractors, MUX / DEMUX ...etc)
  - General purpose components are then used to build larger, function specific, components such as arithmetic and logic unit, control unit and various type of memories.
  - Many function specific components are assembled together to make a microprocessor, I/O HUB, SRAM ... etc
- **Completeness Theorem:** NAND or NOR gate could be used to build any boolean function.

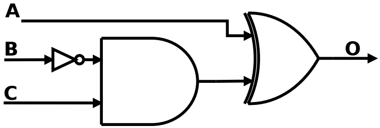
- Logic gates do not produce correct output instantaneously!



- Logic gates do not produce correct output instantaneously!
- Propagation delay is the time taken for a gate to perform its function.



- Logic gates do not produce correct output instantaneously!
- Propagation delay is the time taken for a gate to perform its function.
- Propagation delay is not a constant! it changes with silicon's manufacturing process, age and operating environment.



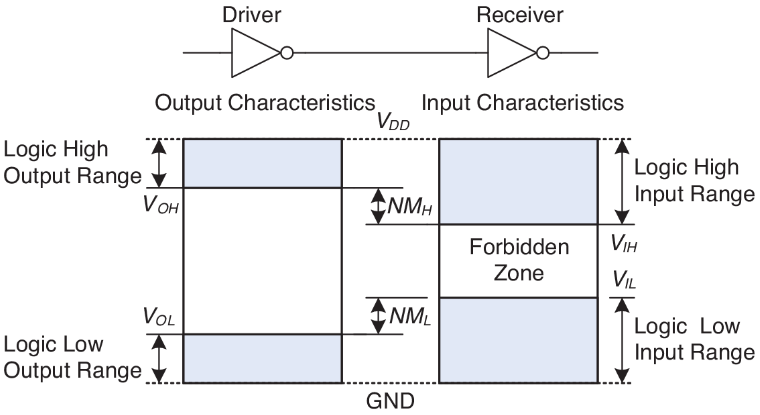
Supply Voltage & Logic Levels

- Digital systems use discrete-value variables
- In reality these variables are represented using voltage levels, a continuous quantity.
- In a 5V (V_{DD}) system, "0" $=$ 0 volts (V_l) and "1" $=$ 5 volts (V_h)
- As transistors become small (V_{DD}) has dropped to lower level. Modern CPU's use 1V or less.

Supply Voltage & Logic Levels

		Receiver			
		TTL	CMOS	LVTTL	LVCMOS
Driver	TTL	OK	NO: $V_{OH} < V_{IH}$	MAYBE[a]	MAYBE[a]
	CMOS	OK	OK	MAYBE[a]	MAYBE[a]
	LVTTL	OK	NO: $V_{OH} < V_{IH}$	OK	OK
	LVCMOS	OK	NO: $V_{OH} < V_{IH}$	OK	OK

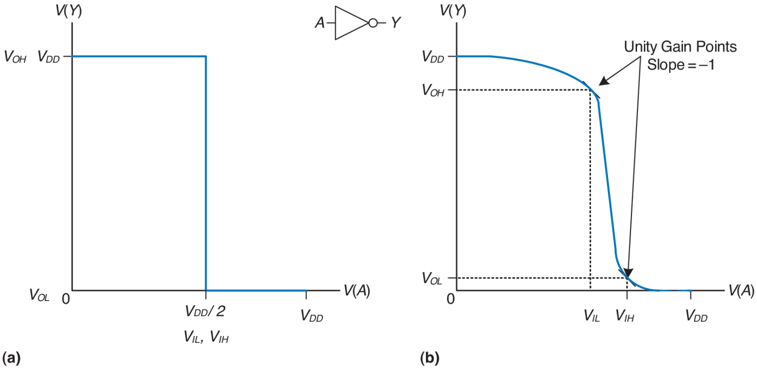
Noise Margins



Noise Margins

Logic Family	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75–5.25)	0.8	2.0	0.4	2.4
CMOS	5 (4.5–6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3–3.6)	0.8	2.0	0.4	2.4
LVCMOS	3.3 (3–3.6)	0.9	1.8	0.36	2.7

DC Transfer Characteristics



Power Consumption

- $P_{dynamic} = 1/2 * C * V_{DD}^2 * f$
- $P_{static} = I_{DD} * V_{DD}$
- $P_{total} = P_{dynamic} + P_{static}$

Week 2 Activity 2, 3

2. **Silicon manufacturing process**: How does silicon manufacturing process/technology affect performance, power and reliability factors of a microprocessor. (Read chapter 1 section 1.4 and 1.5 in from "Computer Architecture: A Quantitative Approach")
3. **Encoding**: In this lecture we saw how decimal numbers are encoded using binary number. Modern personal computer support many languages and are capable of displaying graphics. How are alphabets and pictures represented in a computer? Are there standards that governing how alphabets from various languages are represented? Support your answer with details.

Week 2 Activity 5, & 6

4. **Failure Analysis**: In the DE0-Nano demo, the result of a simple subtractions came out to be a unexpected number, in the demo you saw 2 - 5 = 65533. Is this an error? if it is an error, investigate and explain where the error has occurred. Is it a software or a hardware error? Support your answer with details. Also, suggest a possible solution to fix this error. [1]
5. **Power**: What are the advantages and disadvantages of lowering V_{DD} in digital electronics?
6. **Size**: What is the effect of capacitance and is it possible to reduce or even eliminate capacitance in a digital circuit ?

[1] NOTE: gui_front_2.tcl is considered to be part of software, and week1 schematics is hardware. Both are possible locations where error could have been introduced