# 14:332:231
# DIGITAL LOGIC DESIGN

Ivan Marsic, Rutgers University
Electrical & Computer Engineering
Fall 2013

## Organizational Matters (1)

- Instructor: Ivan MARSIC
  - Office: CoRE Building, room 711
  - Email: marsic@ece.rutgers.edu
  - Office hours: Tuesday 1:30 – 3 PM
- TAs:
  1. Mehrnaz Tavan mehrnaztavan1@gmail.com
  2. Talal Ahmed talal.ahmed@rutgers.edu
  3. Zahra Shakeri ze_shakeri@yahoo.com
  4. Xiangyi Gao xg45@eden.rutgers.edu
- TAs office hours T.B.A.

## Organizational Matters (2)

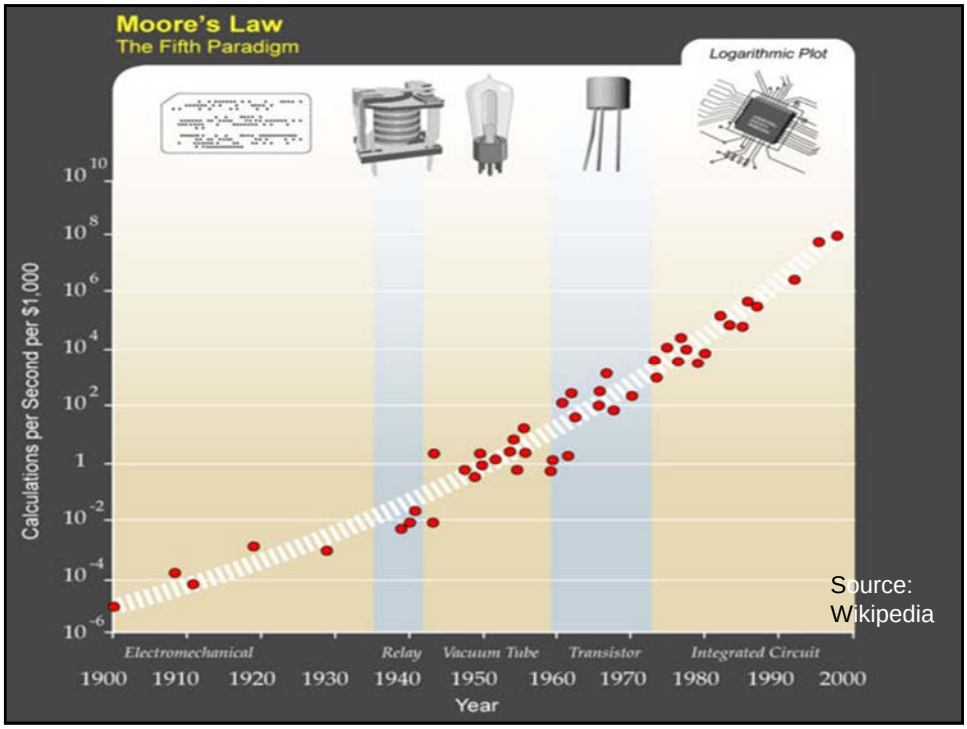
- Required Text:
  - John F. Wakerly, *Digital Design: Principles and Practices*, Prentice Hall: Englewood Cliffs, N.J., 4th Edition, 2005, ISBN 0-13-186389-4
- The laboratory notes will be available via the Web (on Sakai)
- Important announcements will be mailed on the class mailing list
- Labs will be run by the TAs in five (5) different sections

3 of 30

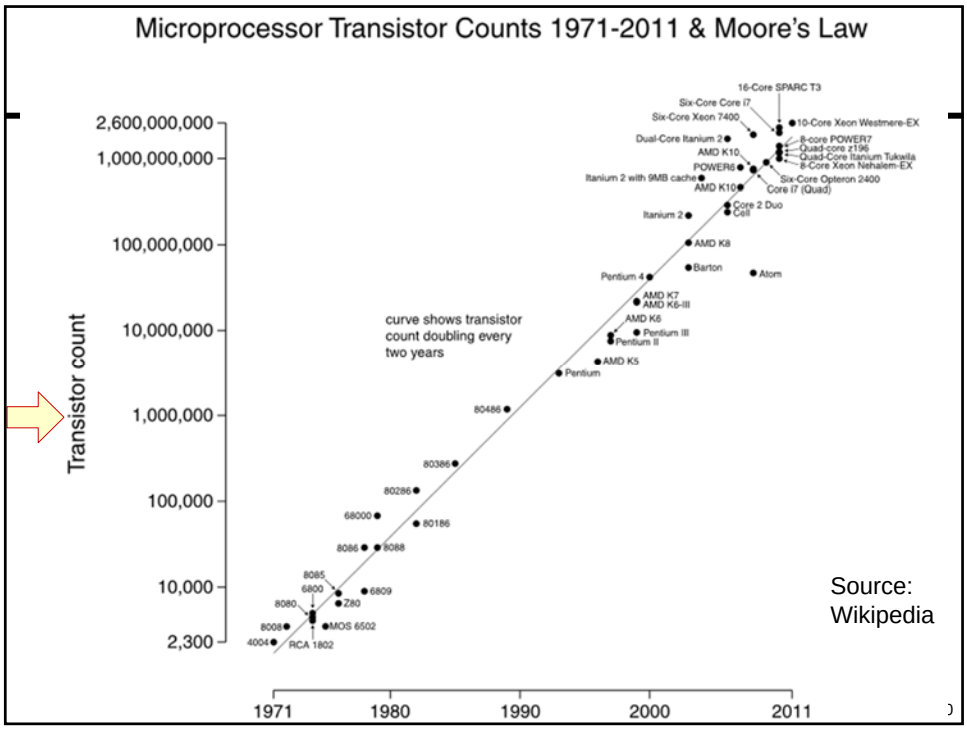
## Organizational Matters (3)

- Grading:
  - Homework 20 %
  - Midterm Exam 1 20 % (~Oct 11 or 18)
  - Midterm Exam 2 20 % (~Nov 19 or 27)
  - Final Exam 30 %
  - Random Quizzes 10 %

4 of 30

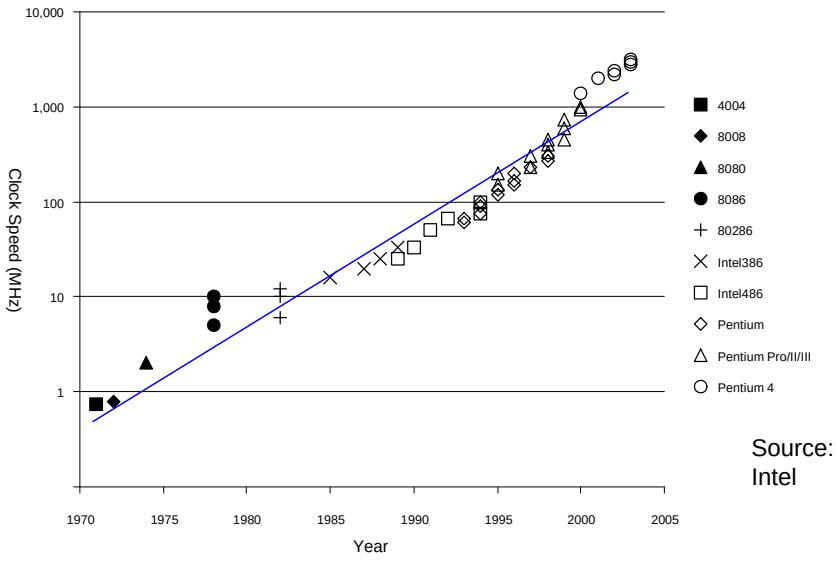



Source: Wikipedia




Source: Wikipedia

# Clock Rate Grows Exponentially




Source: Intel

7 of 30

# What is a Digital Circuit?

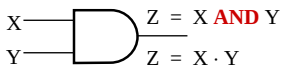
- Defined by the interpretation of the signals (waveforms, if in time) in the circuit.
- Analog
  - Continuous Values
  - Fast, economical, low accuracy
  - Susceptible to noise & distortion
- DIGITAL
  - Discrete Levels, less sensitive to noise
  - Accuracy related to cost (number of “bits”)
  - Less susceptible to noise
- Binary: 2 Levels or States
- Multi-valued: More than two Levels

8 of 30

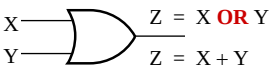
# Digital Logic Elements

- Binary system -- 0 & 1, LOW & HIGH
- Basic building blocks -- AND, OR, NOT *logic gates*

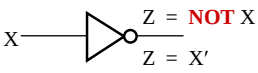
**AND gate**




**OR gate**




**NOT gate or inverter**




| X | Y | X AND Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

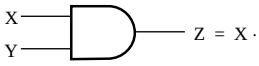
| X | Y | X OR Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

| X | NOT X |
|---|---|
| 0 | 1 |
| 1 | 0 |

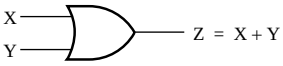


# Logic Gate Symbols and Behavior

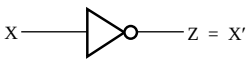
- Logic gates have special symbols:




AND gate

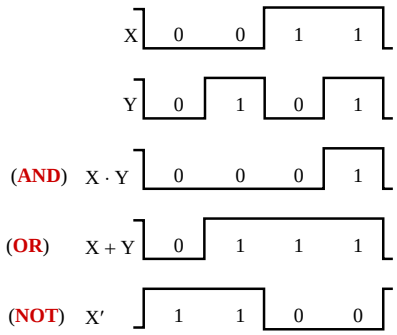



OR gate

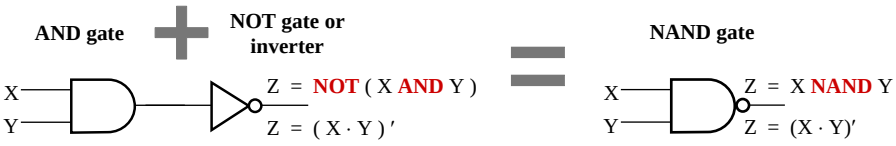



NOT gate or inverter

- And waveform behavior in time as follows:



# Digital Logic Elements (2)

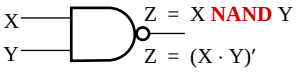



| X | Y | X NAND Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

11 of 30

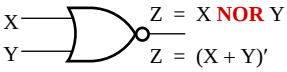
# Digital Logic Elements (3)

NAND gate

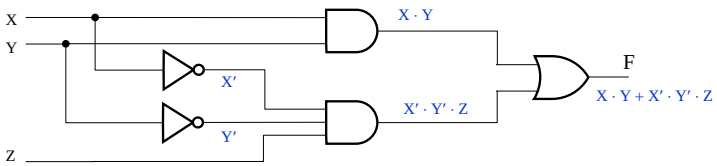



| X | Y | X NAND Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

NOR gate




| X | Y | X NOR Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |




12 of 30

# Truth Tables

- A **Truth Table** is a tabular notation for representing the logic value of a function for *all possible combinations* of the values of its arguments.

| X | Z = NOT X |
|---|---|
| 0 | 1 |
| 1 | 0 |

NOT

| X | Y | Z = X AND Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

AND

| X | Y | Z = X OR Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

OR

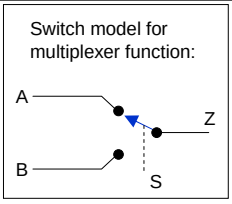
XOR = ??

| X | Y | Z = X XOR Y |
|---|---|---|
| 0 | 0 | **0** |
| 0 | 1 | **1** |
| 1 | 0 | **1** |
| 1 | 1 | **0** |



# Truth Tables & Logic Diagrams

- Conceptual problem representation




- Truth table: Truth table for the multiplexer function

| S | A | B | Z |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

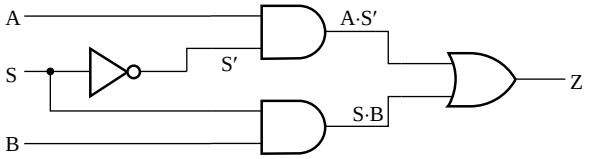
- Synthesize into a logic diagram

# Logic Equations & Logic Diagrams

- Logic equation for a multiplexer:

$$Z = S' \cdot A + S \cdot B$$

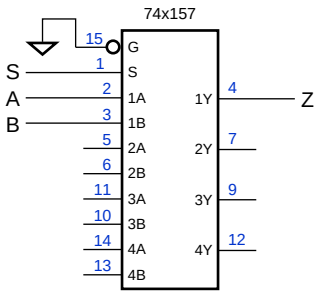
- Logic diagram:
  Gate-level logic diagram for multiplexer function





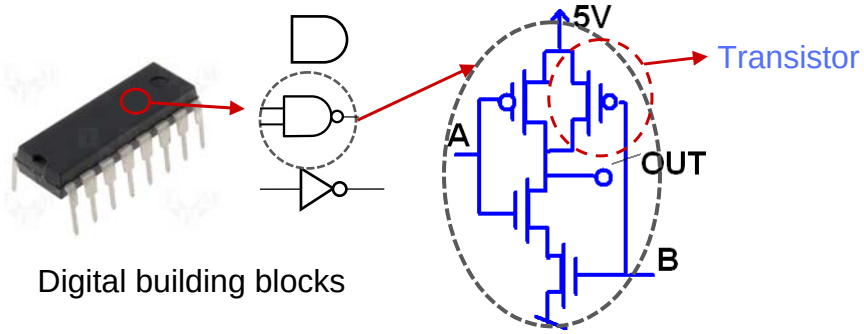

# Prepackaged Building Blocks

E.g., using an MSI building block to implement a multiplexer



# What are Logic Gates built from?

Transistors!

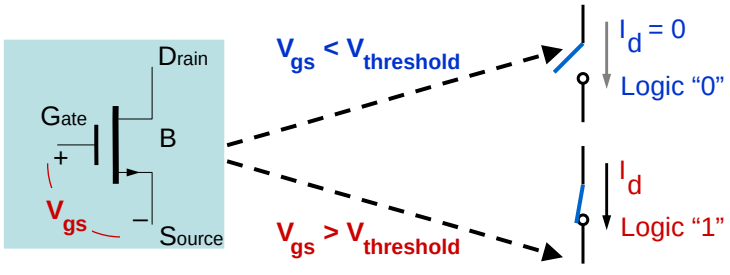



Digital building blocks

17 of 30

# Digital Model of a Transistor

- We make abstraction of the signals: 0 or 1
- As a result a transistors can be considered a switch (ON or OFF; 1 or 0)

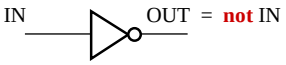


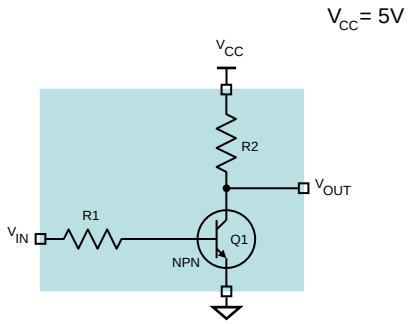

18 of 30

# Transistor-Transistor Logic (TTL)

E.g., bipolar transistor inverter (npn)

**NOT gate or inverter**

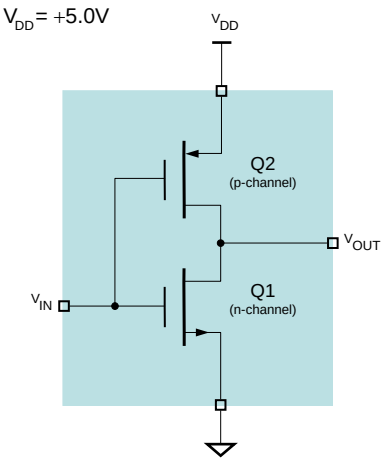


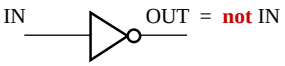

19 of 30

# Complementary Metal-Oxide Semiconductor Field-Effect Transistor (CMOS) Logic

E.g., CMOS inverter




**NOT gate or inverter**

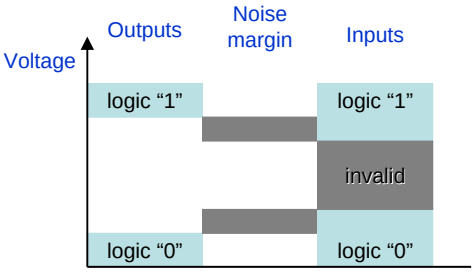



| $V_{IN}$ | $Q_1$ | $Q_2$ | $V_{OUT}$ |
|---|---|---|---|
| 0.0V (L) | off | on | 5.0V (H) |
| 5.0V (H) | on | off | 0.0V (L) |

20 of 30

# Logic Levels (1)

- Undefined region is inherent
  - "0" ≡ 0 - 0.8 xxxxx "1" ≡ 2 - 5 V TTL
  - "digital", not analog voltages

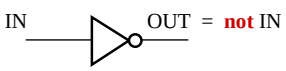


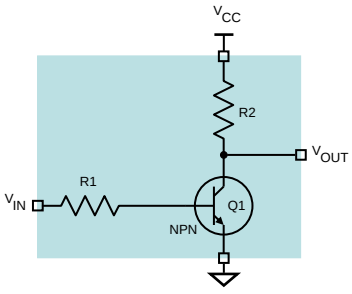

Logic values and noise margins

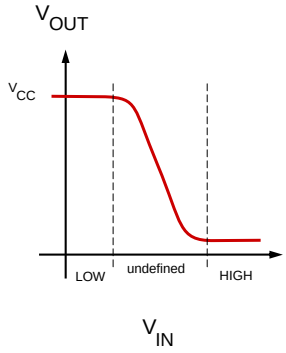


# Resistor-Transistor Inverter

**NOT gate or inverter**



# Logic Levels (2)

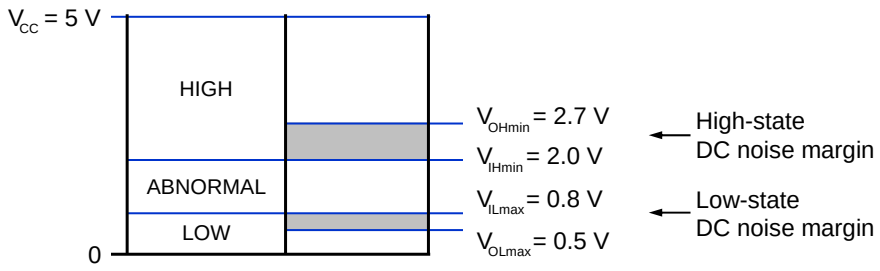
- Switching threshold varies with voltage, temperature, production process, phase of the moon, etc.
  - Need "noise margin" between the output and input
- The more you push the technology, the more "analog" it becomes
  - Accidental versus purposeful "push"
  - Logic voltage levels (CMOS) decreasing with progress in microelectronics

    $V_{CC}$: 5 → 3.3 → 2.5 → 1.8 V

23 of 30

# TTL Logic Levels & Noise Margins

- Asymmetric (unlike CMOS, which could also be *symmetric*)




- Some CMOS families can be made compatible with TTL

24 of 30

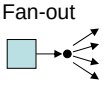
# Importance of Specifications

- Digital analysis works only if circuits are operated in specifications from the manuals:
  - Power supply voltage
  - Temperature
  - Input-signal quality
  - Output loading
- Must do some “analog” analysis too to prove that circuits are operated in specifications:
  - Fan-in and fan-out specifications
  - Timing analysis


25 of 30

# Input/Output Loading Specs

- Each gate input requires a certain amount of current to drive it in the LOW state and in the HIGH state:
  - $I_{IL}$ and $I_{IH}$
  - These amounts are specified by the manufacturer
- Fan-out calculation:
  - (LOW state) The sum of the $I_{IL}$ values of the driven inputs may not exceed $I_{OLmax}$ of the driving output
  - (HIGH state) The sum of the $I_{IH}$ values of the driven inputs may not exceed $I_{OHmax}$ of the driving output



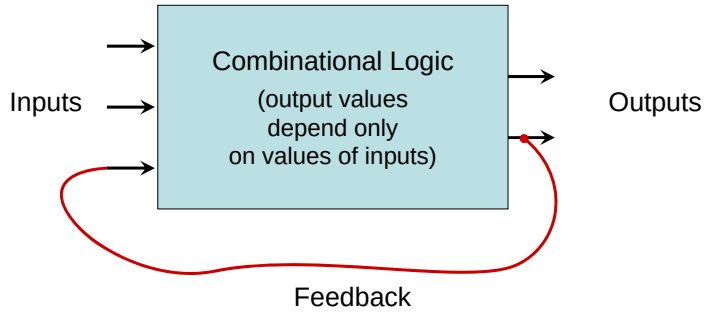

26 of 30

# Combinational Circuits

- **Combinational circuit**'s output depends only on the current input values (called an *input combination*)
  - IDEAL: Output responds instantly
  - REALITY: Propagation delay, spurious outputs ("glitches") that *must* eventually settle
- *Sequential circuit*'s output depends not only on its current input but also on the past sequence of inputs that have been applied to it.
  - I.e., a sequential circuit has *memory* of past events
  - REALITY: Output depends on the *duration* of the inputs

27 of 30

# Combinational & Sequential Systems



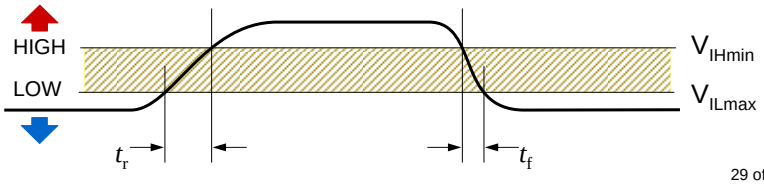

28 of 30

# Transition Times

(a) Ideal case of zero-time switching:

(b) A more realistic approximation:


(c) Actual timing, showing rise and fall times:






# Classification of Designs

- SYNCHRONOUS
  - All feedback occurs at the same time, as defined by a
- special signal called a ***clock***
- ASYNCHRONOUS
  - Feedback occurs whenever values change. No special clock signals

- Remember:
  Avoid asynchronous design if you can!
  Try to identify the unavoidable asynchronous interfaces and convert them to synchronous form
  as reliably and with as few gates as possible