EECS 3201: Digital Logic Design Lecture 3

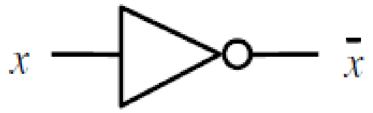
Ihab Amer, PhD, SMIEEE, P.Eng.

Logic Gates

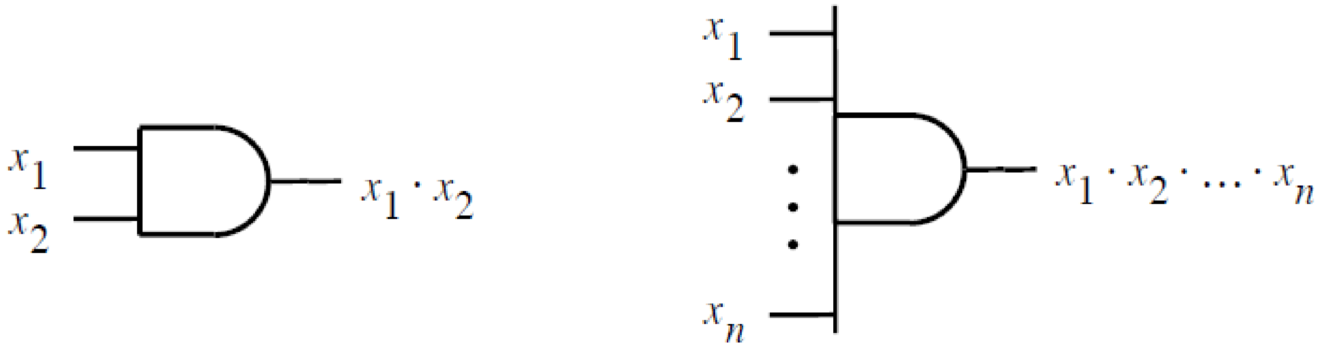
- Circuit elements
- A representation of the physical instantiation of a logic function
- Single-input
  - NOT gate, buffer
- Two-input
  - AND, OR, XOR, NAND, NOR, XNOR
- Multiple-input
- Combine these into a circuit schematic consisting of graphical symbols

Basic Gates

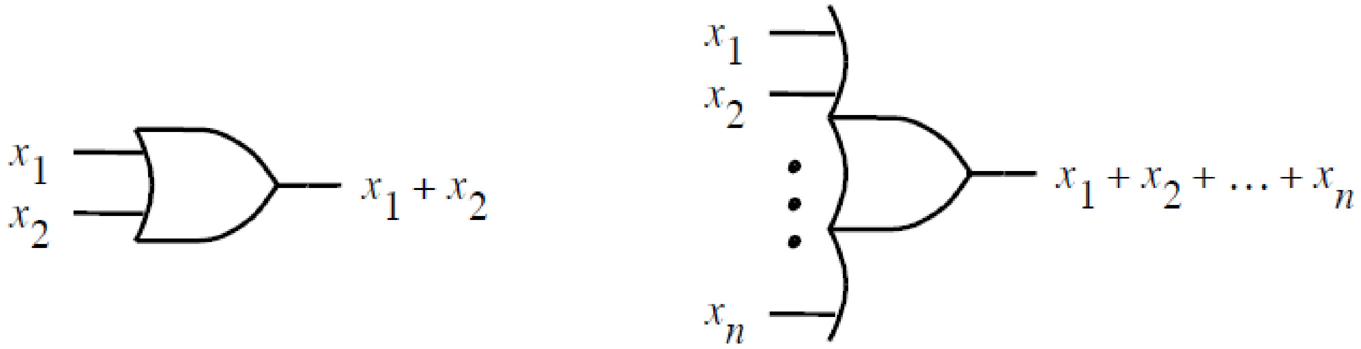
NOT gate



AND gates

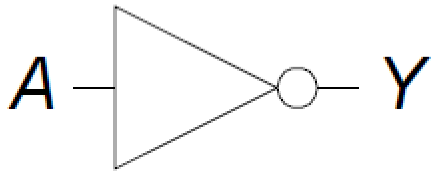


OR gates



Single-Input Logic Gate

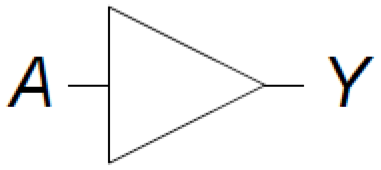
NOT



$Y = \overline{A}$

A	Y
0	
1	

BUF

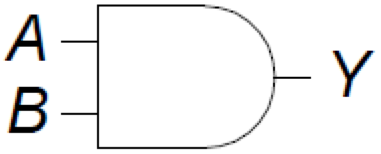


$Y = A$

A	Y
0	
1	

Two-Input Logic Gates

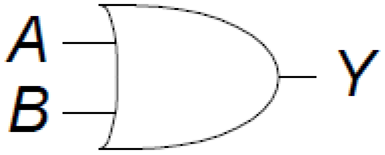
AND



$Y = AB$

A	B	Y
0	0	
0	1	
1	0	
1	1	

OR

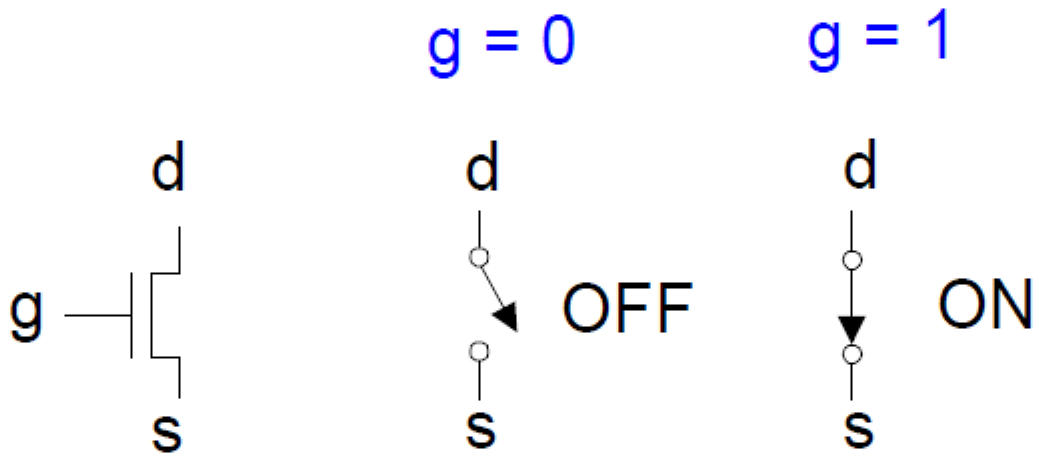


$Y = A + B$

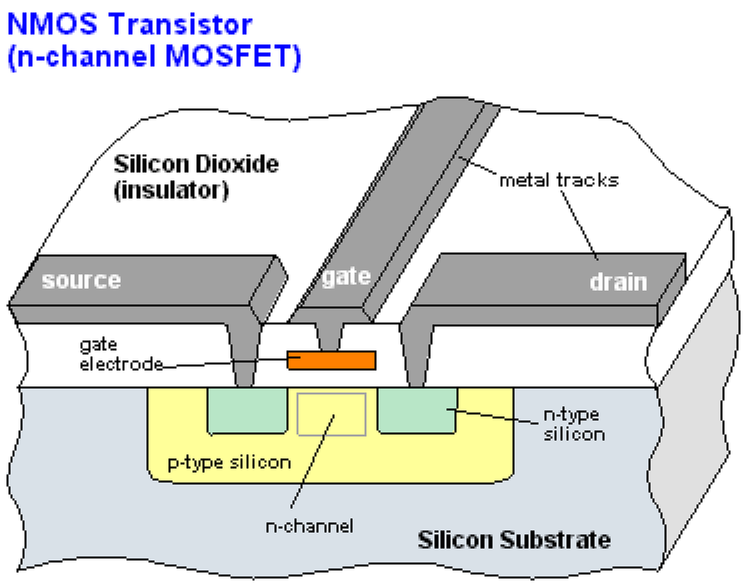
A	B	Y
0	0	
0	1	
1	0	
1	1	

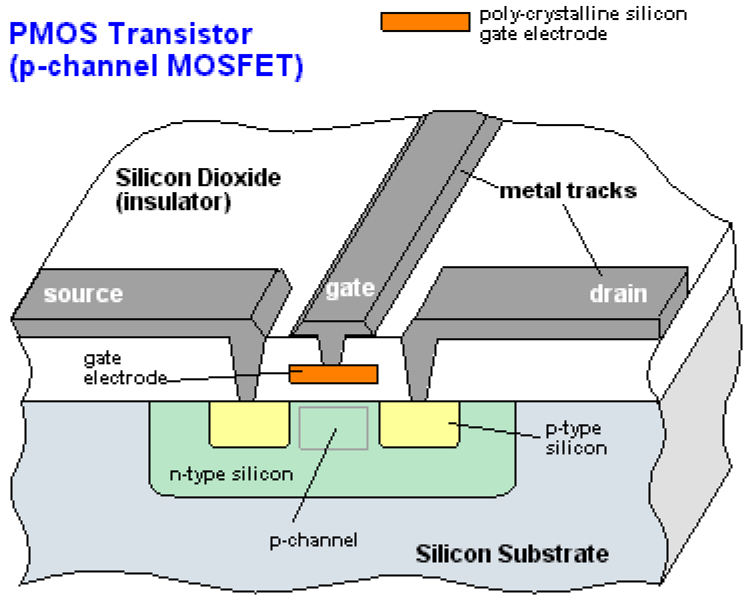
Transistors

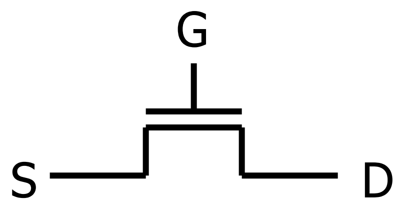
- Logic gates built from transistors
- 3-ported voltage-controlled switch
  - 2 ports connected depending on voltage of 3rd
  - d and s are connected (ON) when g is 1



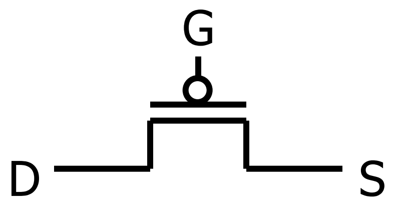
MOSFET





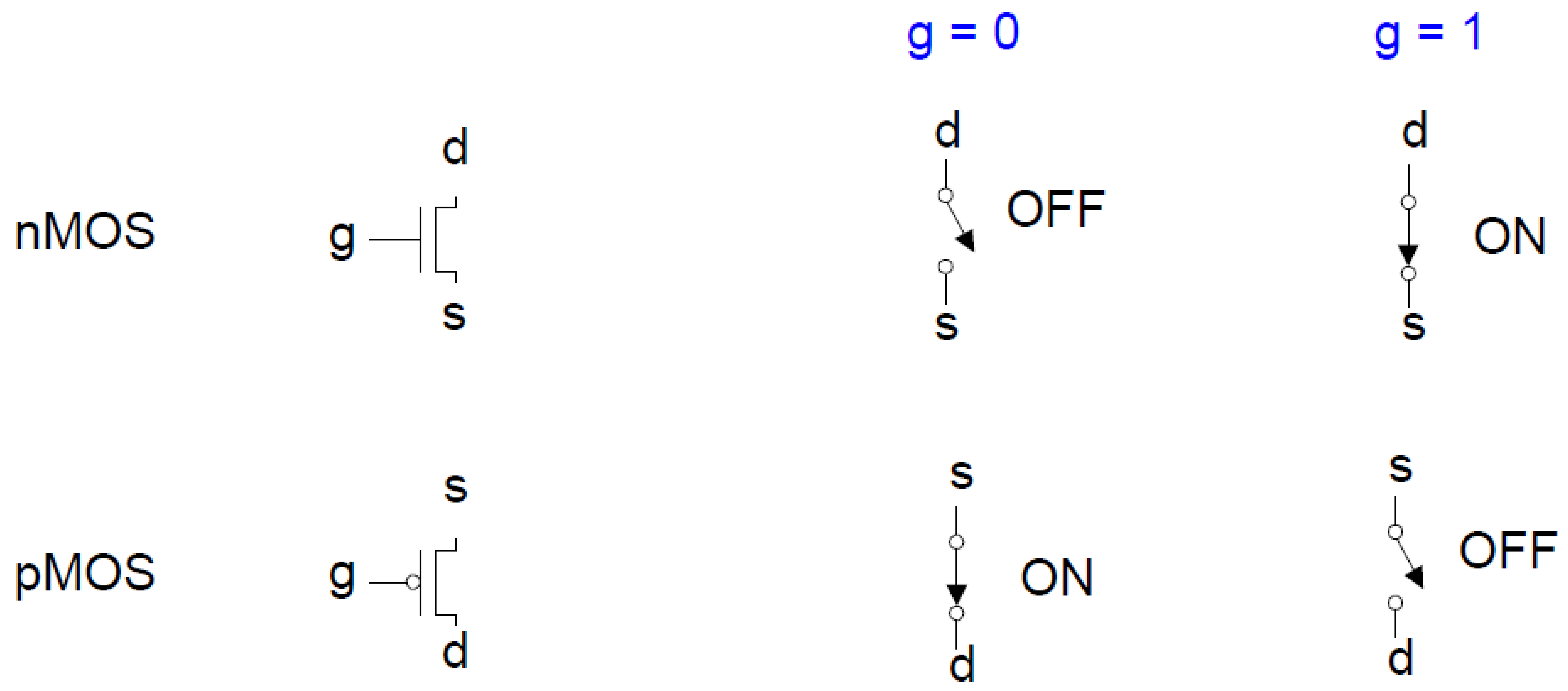


n-channel
open when voltage at G is low
closed when voltage at G is high



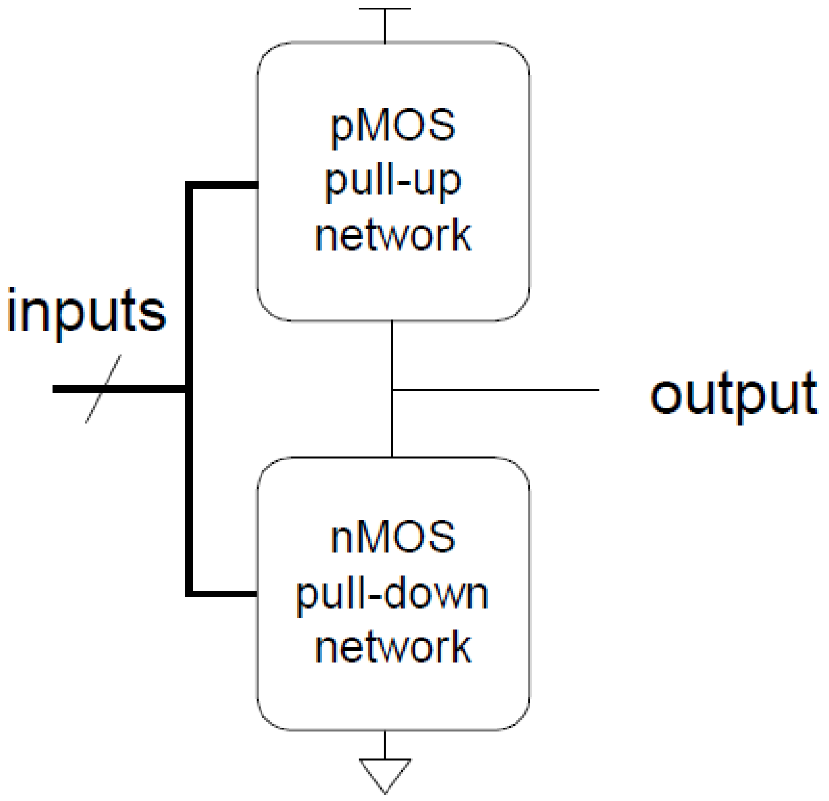
p-channel
closed when voltage at G is low
open when voltage at G is high

Two MOS Transistor Types (CMOS)

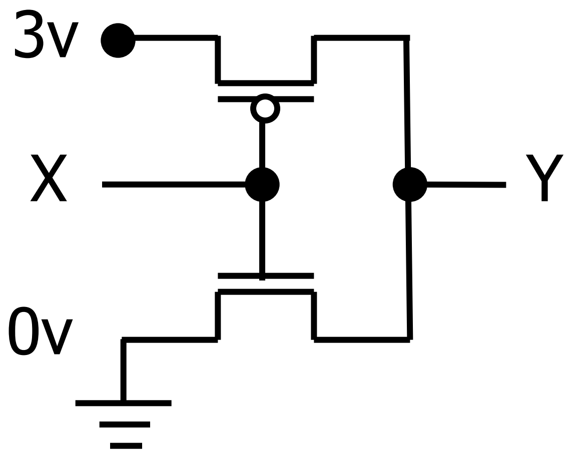


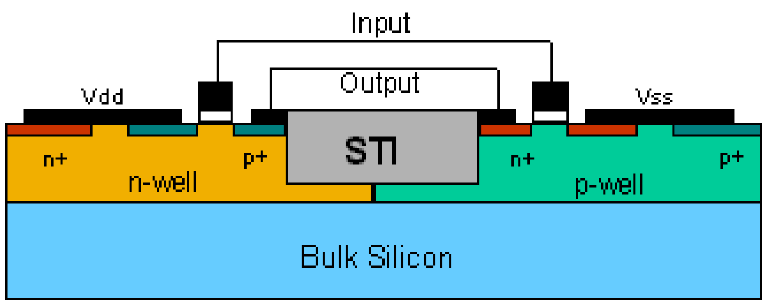
Transistor Function

- nMOS: passes “good” 0’s, so connect source to *GND*
- pMOS: passes “good” 1’s, so connect source to V_{DD}

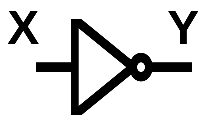


CMOS Basic Logic Gates

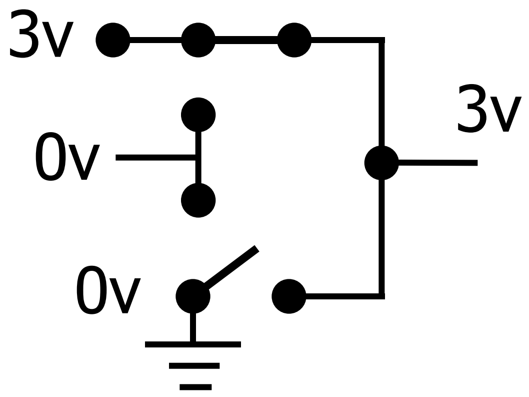


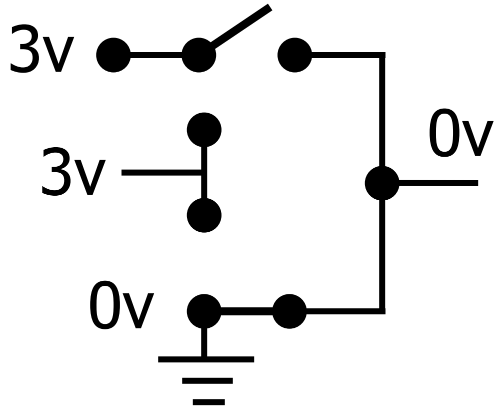


≡



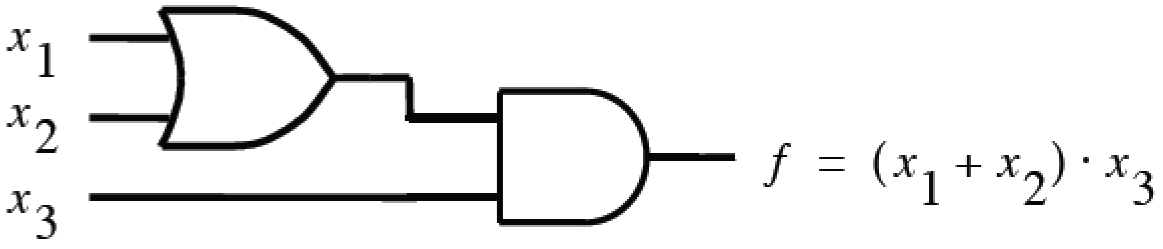
X	Y
0 volts	3 volts
3 volts	0 volts





Logic Networks

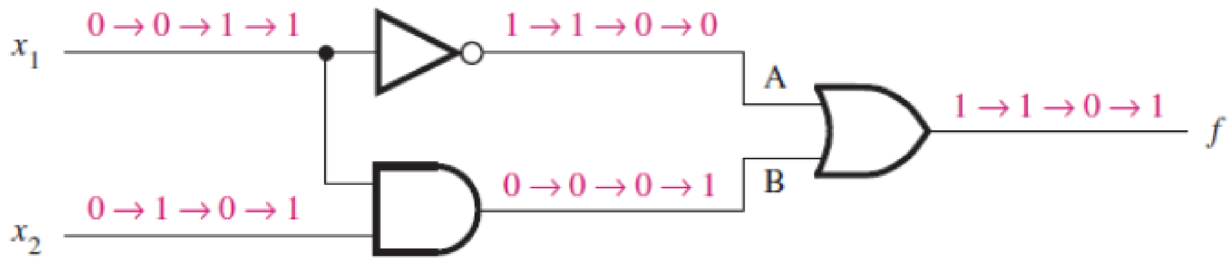
- Larger circuit implemented by network of gates
  - A Logic Network/logic circuit



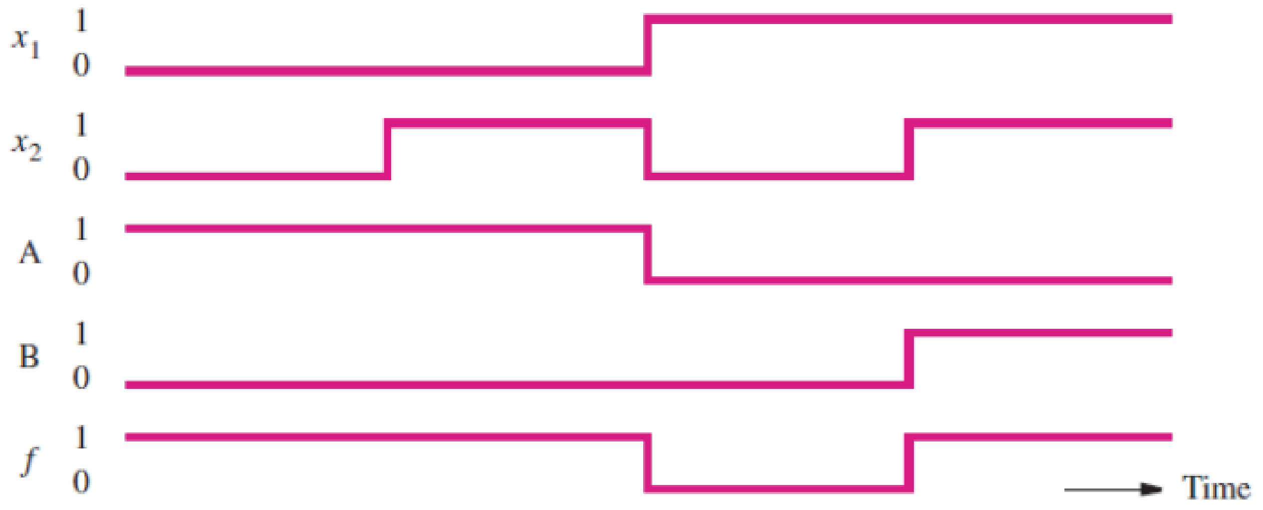
- One thing you must be able to do is analyze logic networks

Network Analysis

- Figuring out what a network does
- Describe with truth tables and timing diagrams



x_1	x_2	$f(x_1, x_2)$	A	B
0	0	1	1	0
0	1	1	1	0
1	0	0	0	0
1	1	1	0	1



Network Synthesis

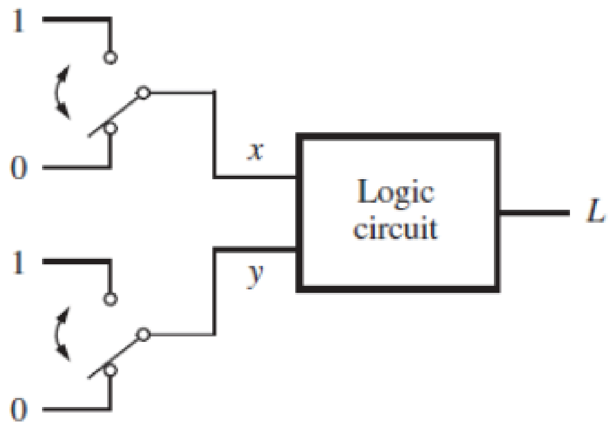
- Another thing you must be able to do is synthesize logic networks

- Go from truth tables/timing diagrams to circuits

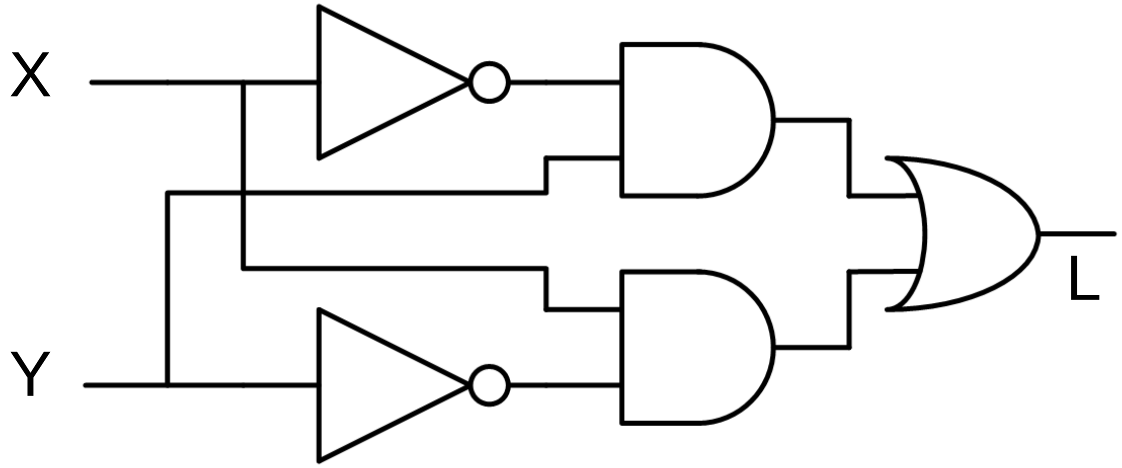
- Important to synthesize networks consisting of optimum arrangement of gates
  - minimize number of gates
  - reduce number of inputs
  - lower interconnect
  - reduce power, etc.

Basic Synthesis

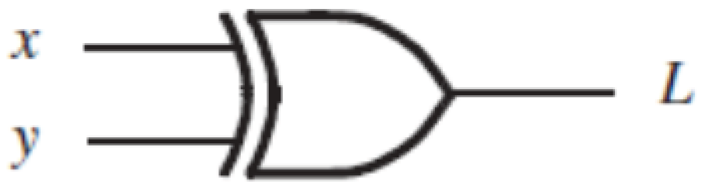
- Synthesize this…



x	y	L
0	0	0
0	1	1
1	0	1
1	1	0



XOR

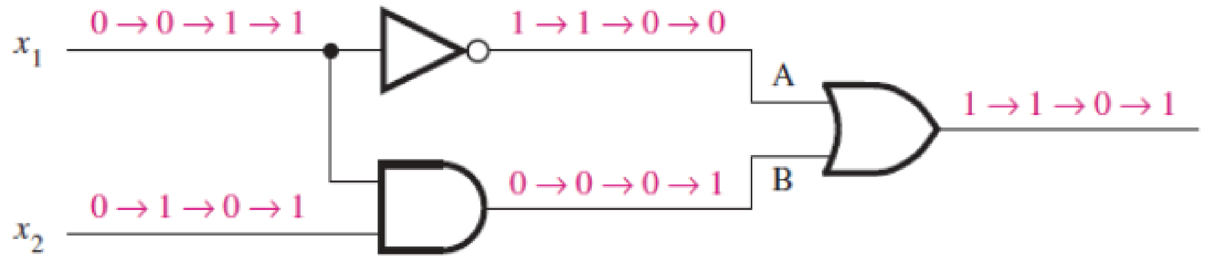


x	y	L
0	0	0
0	1	1
1	0	1
1	1	0

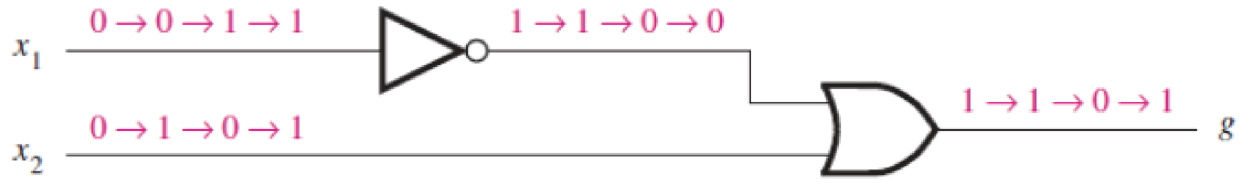
- $L = y \cdot x' + y' \cdot x$
- $L = y \oplus x$

Polymorphic Mapping

- A logic function can be implemented in more than one way



$$f = \bar{x}_1 + x_1 \cdot x_2$$



$$g = \bar{x}_1 + x_2$$

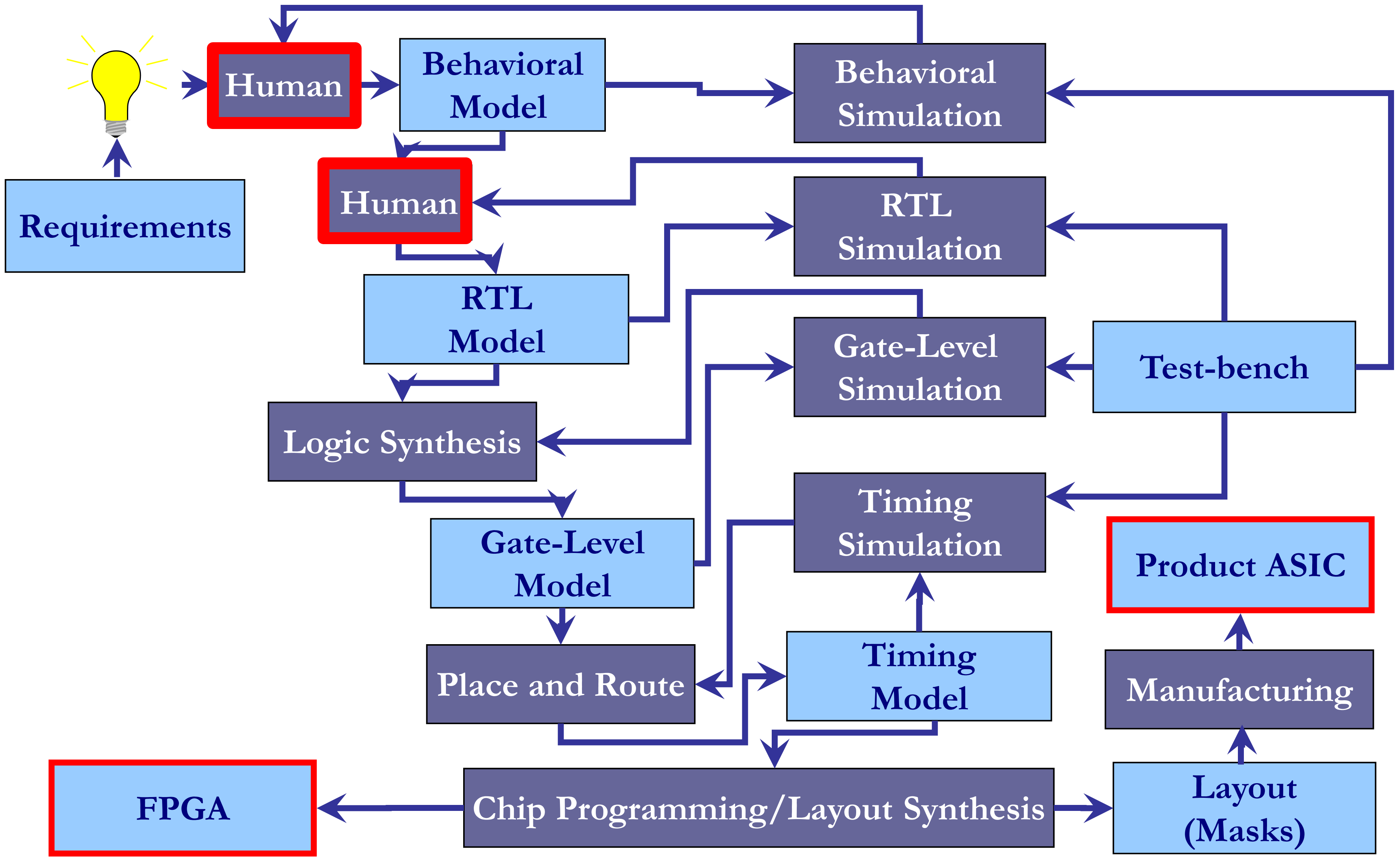
- How do you quickly find the best one?

- Boolean Algebra!

From Previous Lecture(s)

- Modern digital systems use ICs almost exclusively in their designs
- Modern VLSI and ULSI ICs typically contain millions of transistors
- Traditional design approach provides insight and understanding of the problem
- However, it is inadequate for real problems
- Hence, the other design approach (CAD) is an essence for large (real) problems

Basic Design Methodology



Our Task in the Flow

- *Modeling* the digital system based on the requirements
- Modeling can be at various levels of abstraction (will be discussed later)
- How to perform this modeling?

Using HDL: Hardware Description Language

Reasons for Modeling

- Requirements Specification
- Documentation
- Testing using Simulation
- Synthesis

<u>Goal:</u> Most reliable design process, with minimum cost and time

What is a HDL?

- A high-level computer language that can describe digital systems in textual form

- Two applications of HDL processing:
  - Logic Simulation
  - Logic Synthesis

HDL Applications

- Logic Simulation
  - A simulator translates the HDL description to a readable output such as *timing diagram*
  - It predicts how the hardware will work before it is actually fabricated
  - Functional errors can be corrected before actual fabrication
  - Stimulus that tests the design is called *test-bench (also written in HDL)*
- Logic Synthesis
  - Deriving the *gate-level netlist* from the HDL
  - Typically accompanied with optimization, and automated with computer software
  - Restrictions on coding style for RTL model
  - The outcome (netlist) is tool dependent

References

- Lecture Notes of Dr. Sebastian Magierowski – Fall 2013
- Digital Design, 3rd Edition, M. Morris, Mano
- Digital Design, 4th Edition, John Wakerly
- cpk.auc.dk/education/SSU-2007/mm10/ssu_mm10.pdf
- www.ece.cmu.edu/~thomas/VSLIDES.pdf
- http://ece.gmu.edu/coursewebpages/ECE/ECE448/S10/