Digital Design with the Verilog HDL

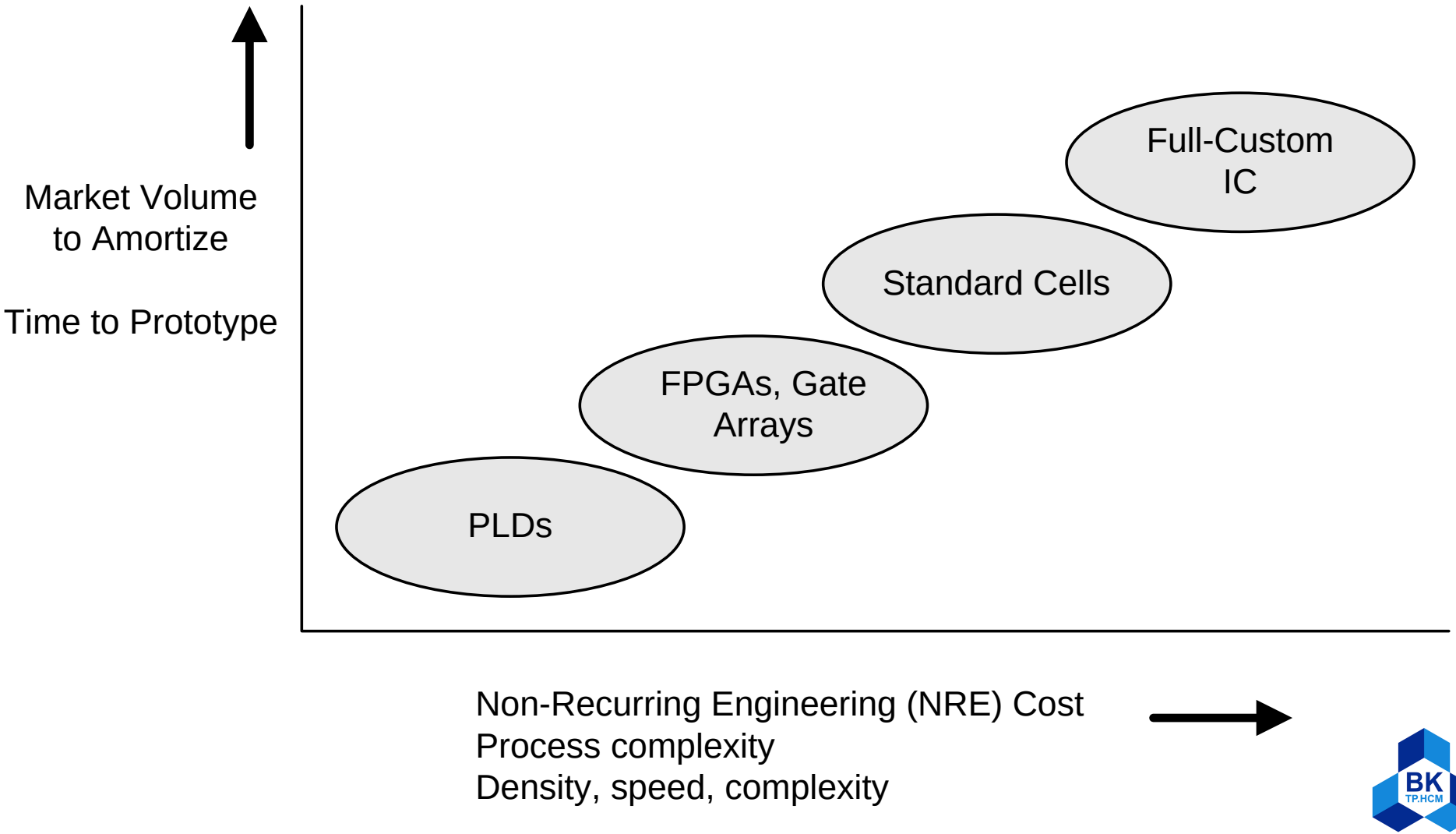
Chapter 1: Digital Design Review

Binh Tran-Thanh

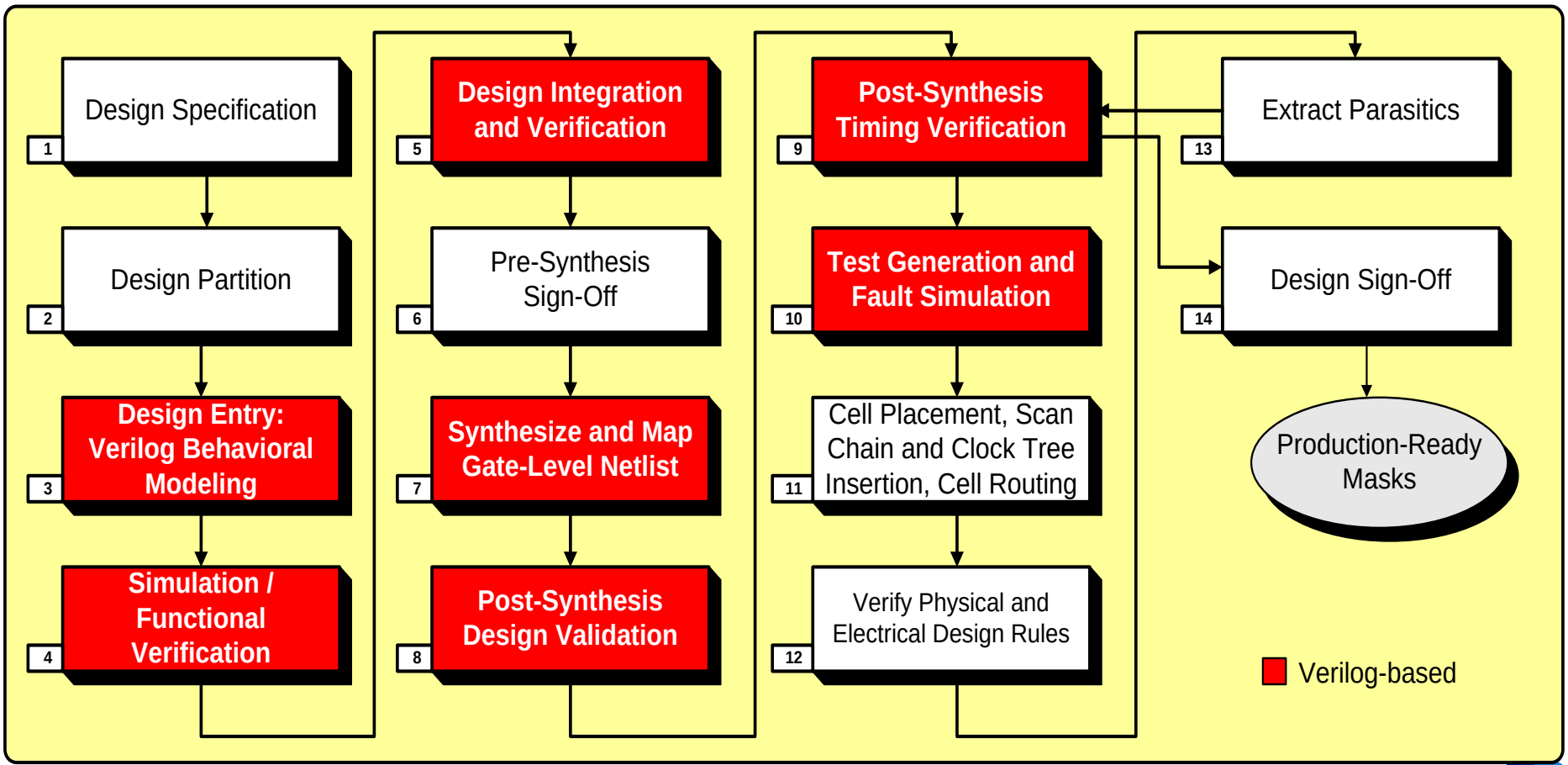
Department of Computer Engineering
Faculty of Computer Science and Engineering
Ho Chi Minh City University of Technology

January 15, 2024

Technology Tradeoffs

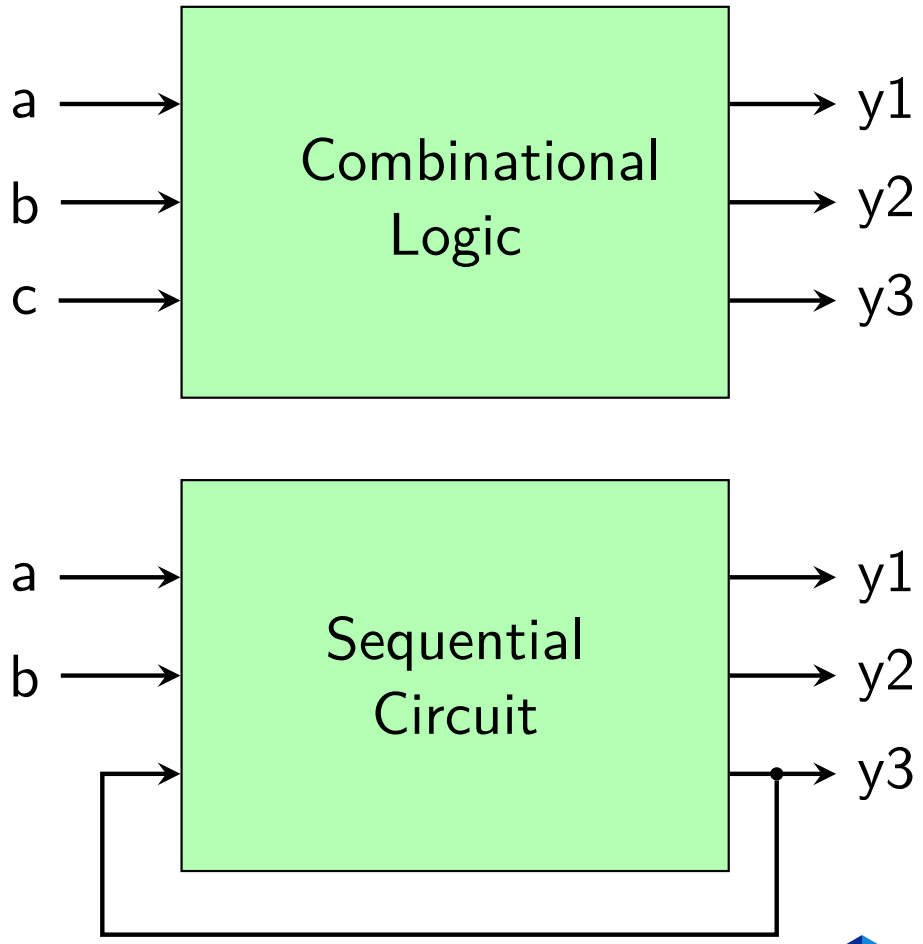


Design Methodology



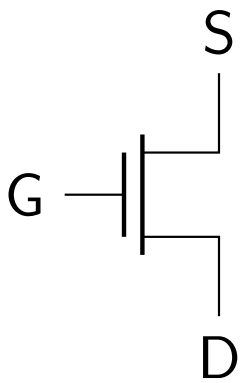
Combinational – Sequential Logic

- Combinational logic:
  - The outputs at any time, t, are a function of only the inputs at time t
- Sequential logic:
  - The outputs at time t are a function of the inputs at time t and **the outputs at time t-1**

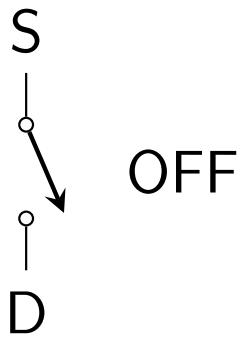


Transistor

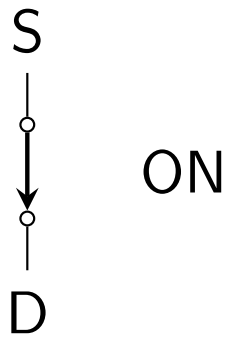
- nMos



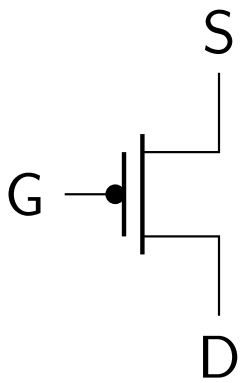
$G = 0$



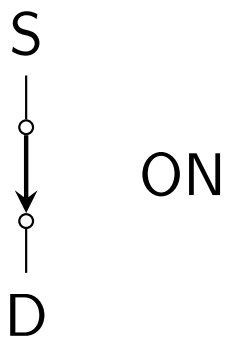
$G = 1$



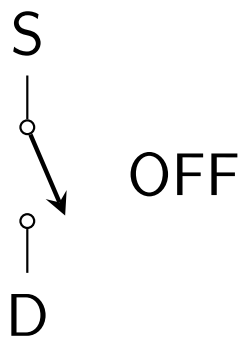
- pMos



$G = 0$

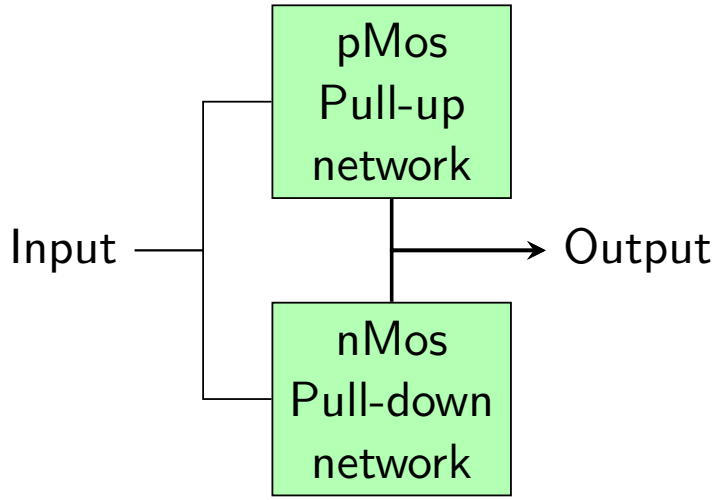


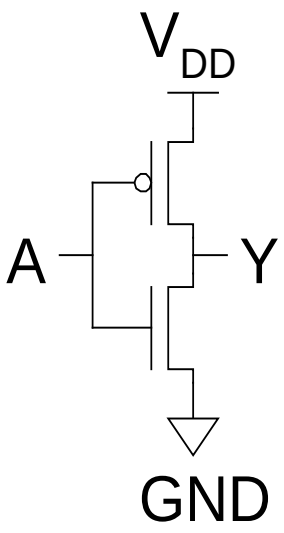
$G = 1$



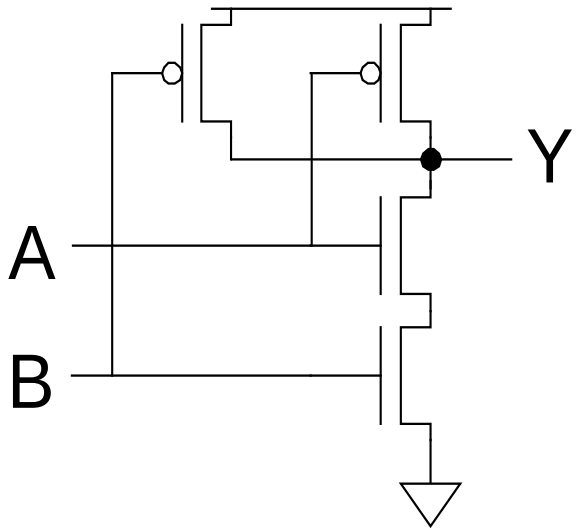
CMOS Technology

- Complementary metal-oxide semiconductor
- Outputs are always either 0 or 1

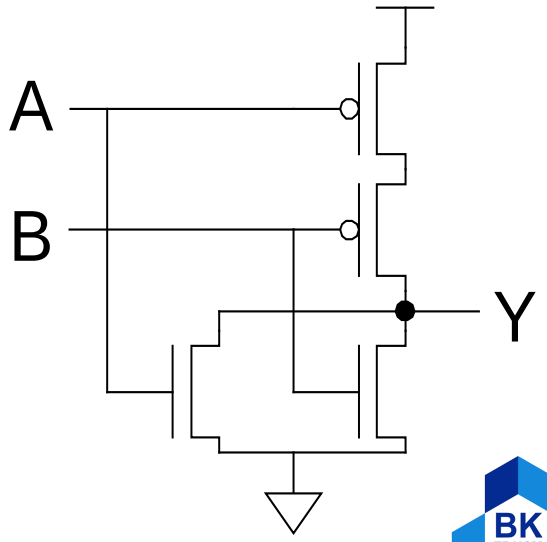




Inverter



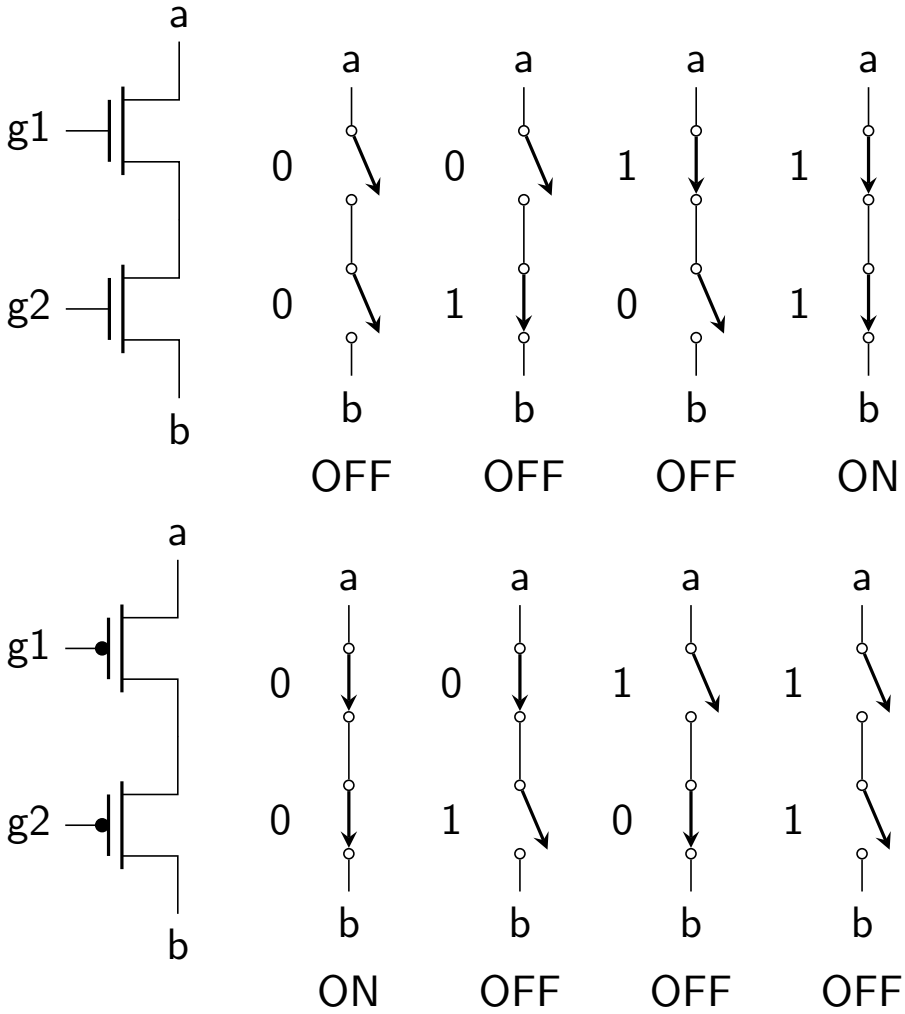
NAND gate



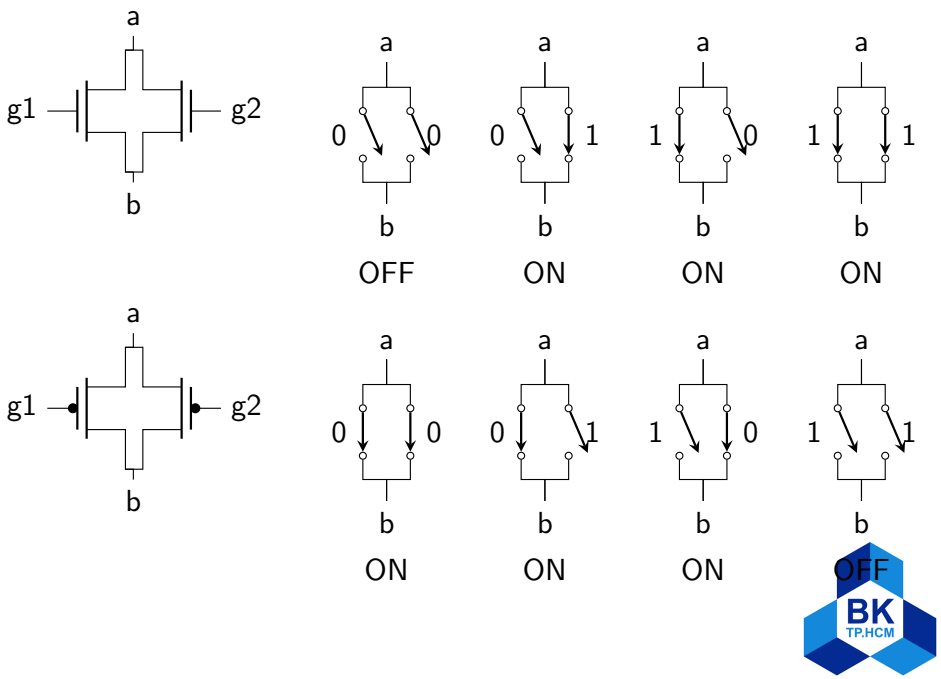
NOR gate

Parallel and Serial

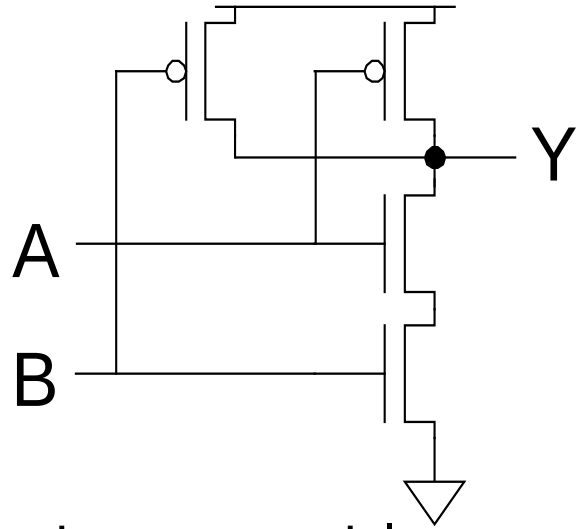
- nMOS: 1 = ON
- pMOS: 0 = ON



- **Series**: all transistors are on
- **Parallel**: at least one transistor is on



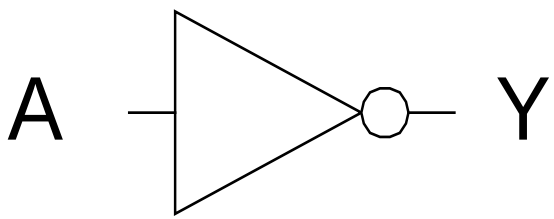
The "Conduction Complement" Rule

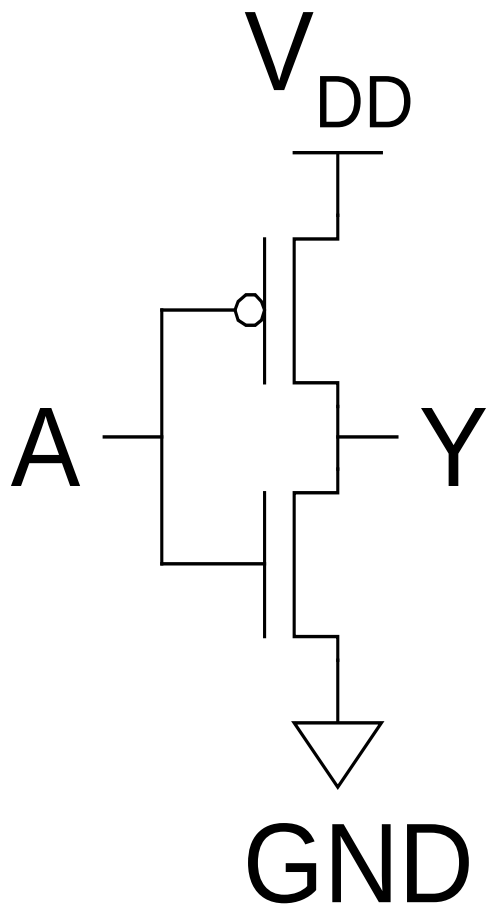


- CMOS gate's output is always either 0 or 1
- For example: NAND
  - Y=0 if and only if both inputs are 1
  - Y=1 if and only at least one input is 0
  - pMos transistors are parallel while nMos transistors are serial
- The "Conduction Complements" rule
  - The pull-up network always complements the pull-down network
  - **Parallel $\rightarrow$ Serial, Serial $\rightarrow$ Parallel**

CMOS Inverter

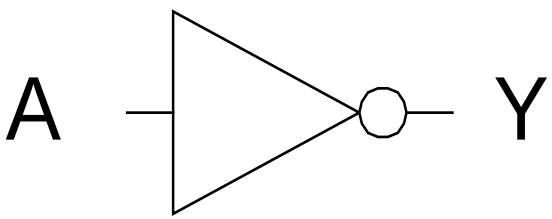
A	Y
0	
1	

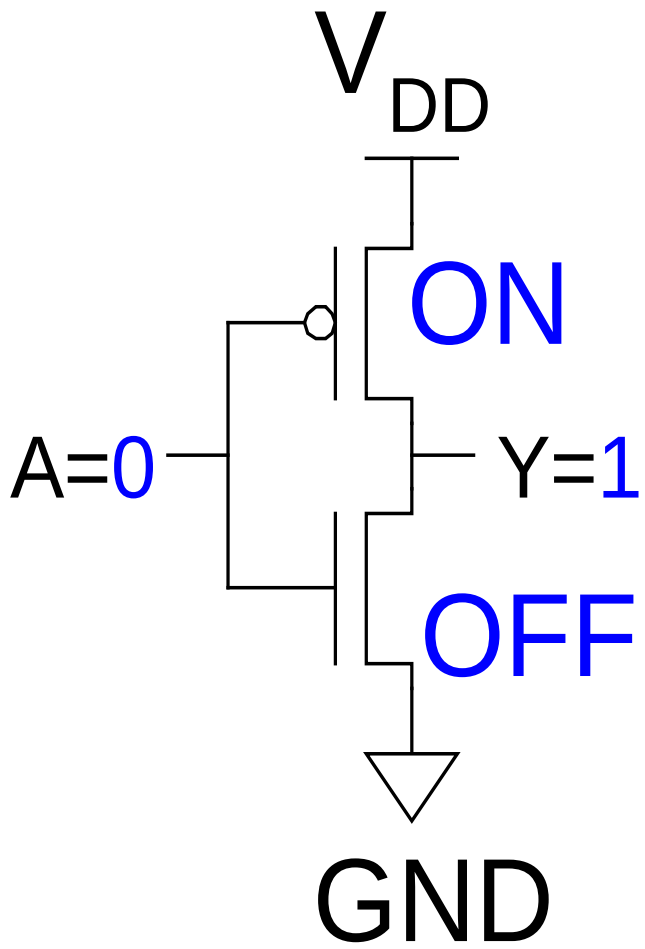




CMOS Inverter

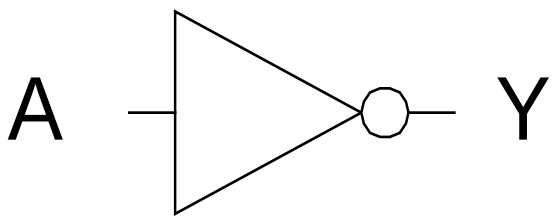
A	Y
0	1
1	

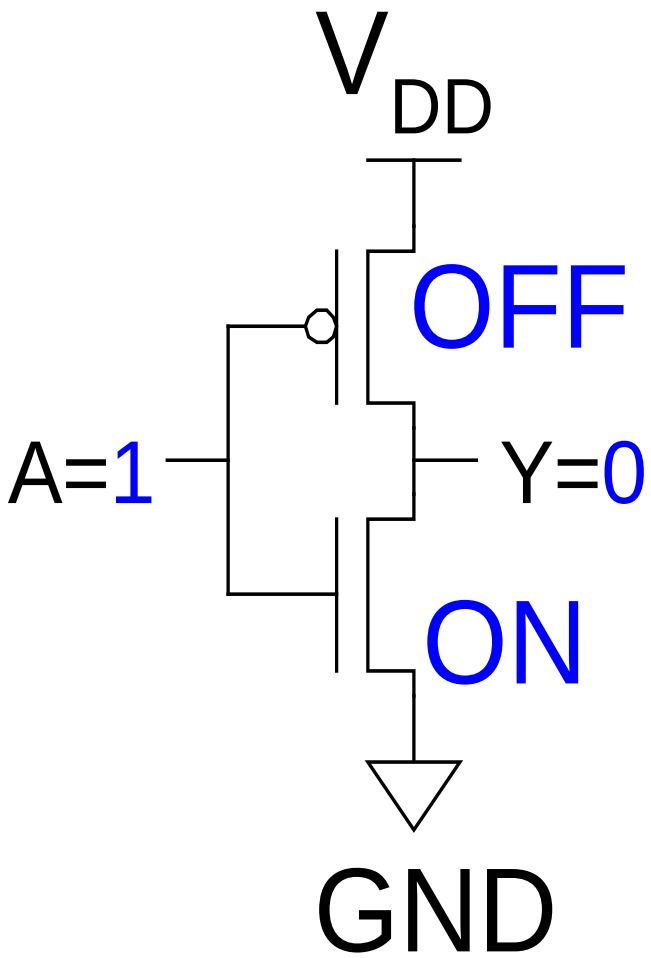




CMOS Inverter

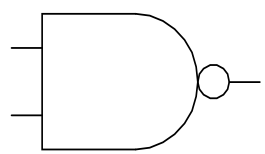
A	Y
0	1
1	0

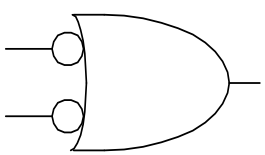


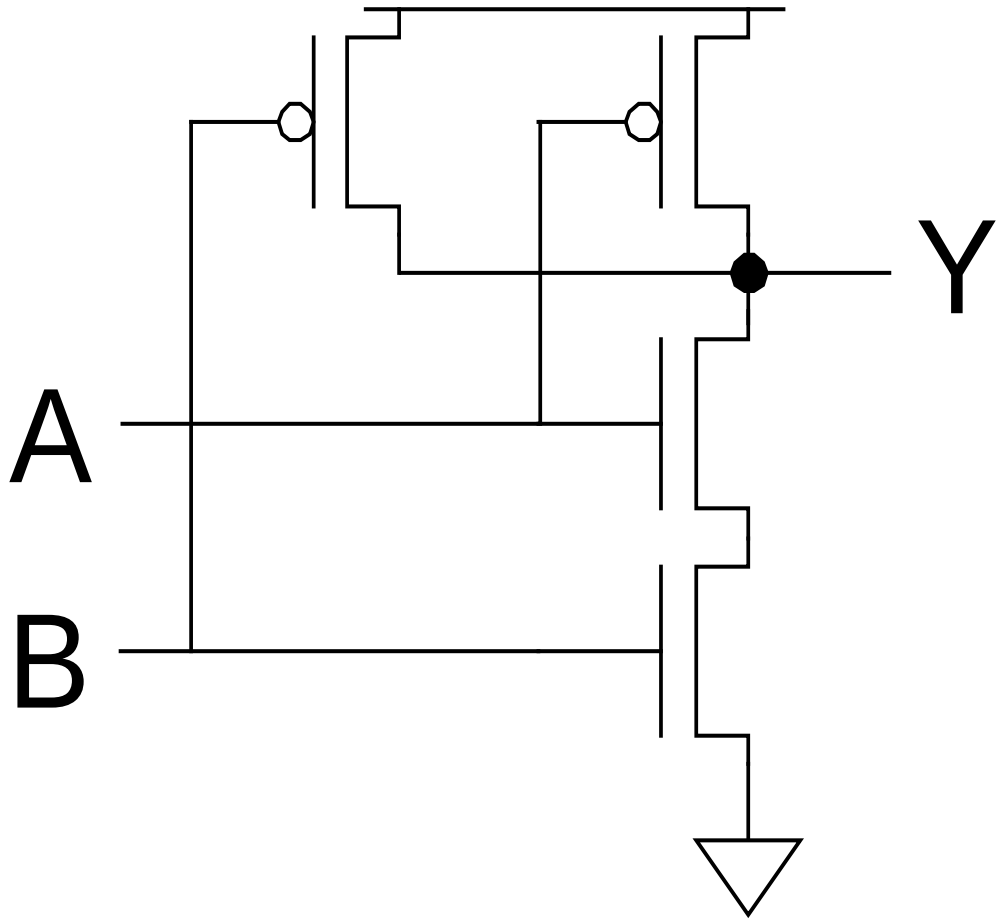


CMOS NAND Gate

A	B	Y
0	0	
0	1	
1	0	
1	1	

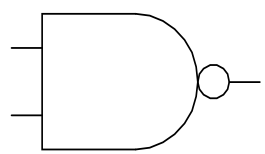


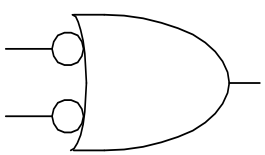


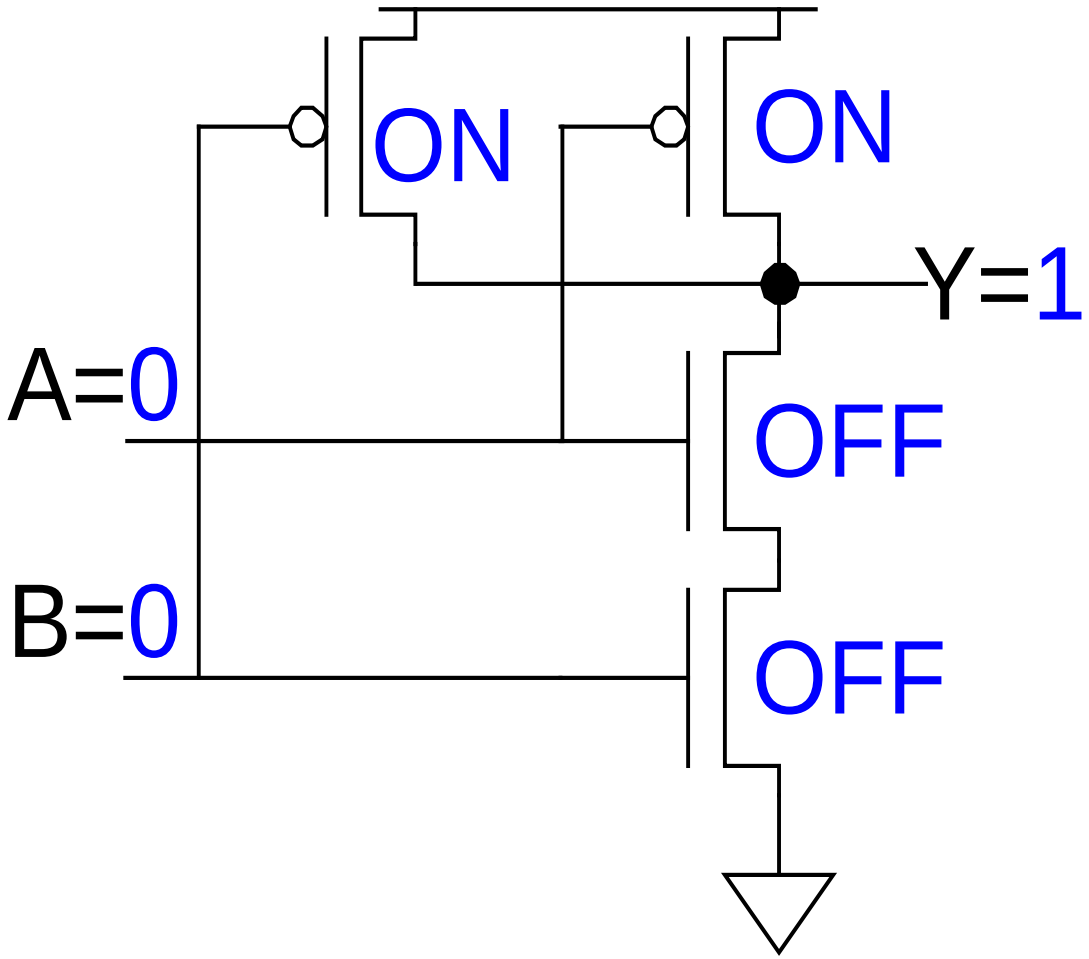


CMOS NAND Gate

A	B	Y
0	0	1
0	1	
1	0	
1	1	

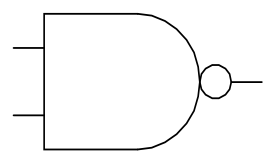


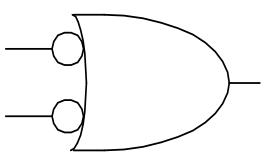


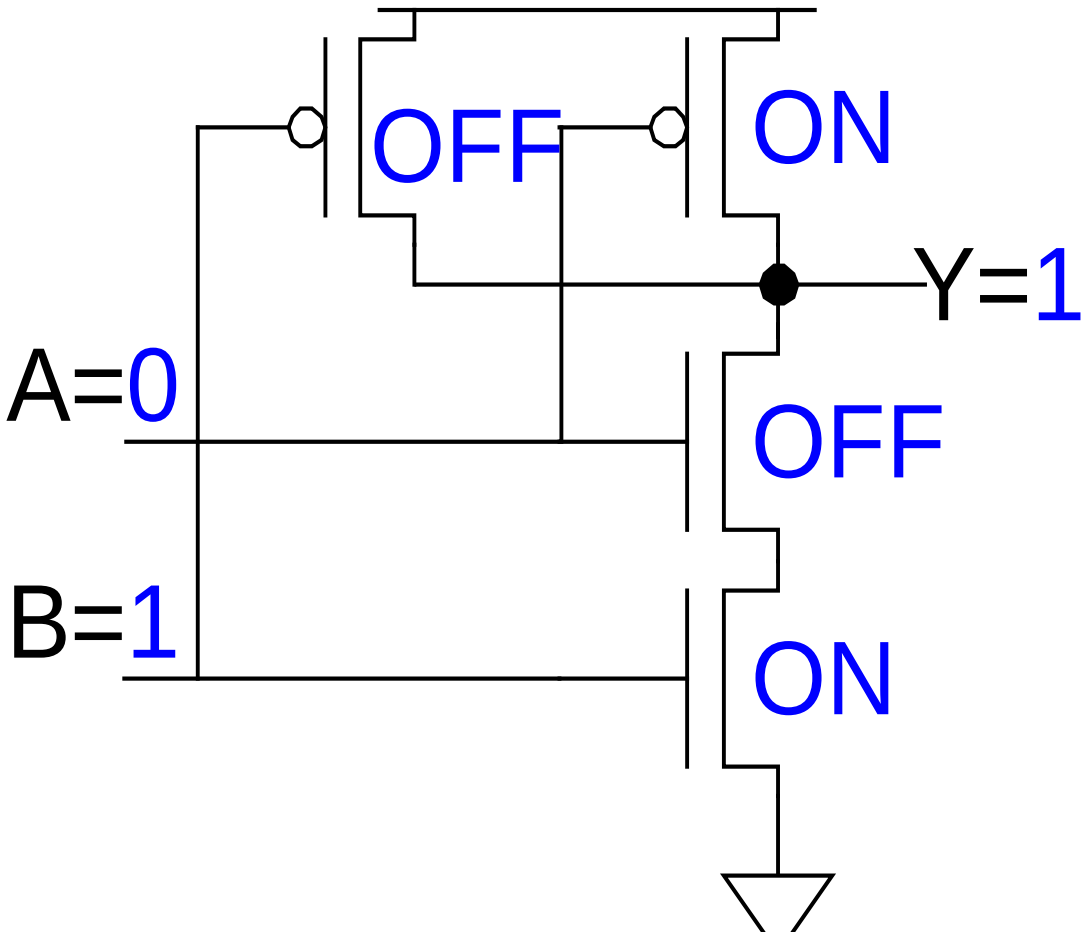


CMOS NAND Gate

A	B	Y
0	0	1
0	**1**	**1**
1	0	
1	1	

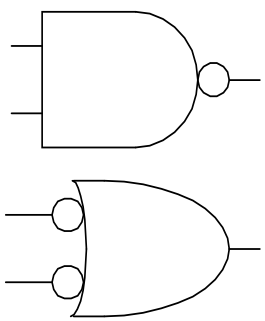


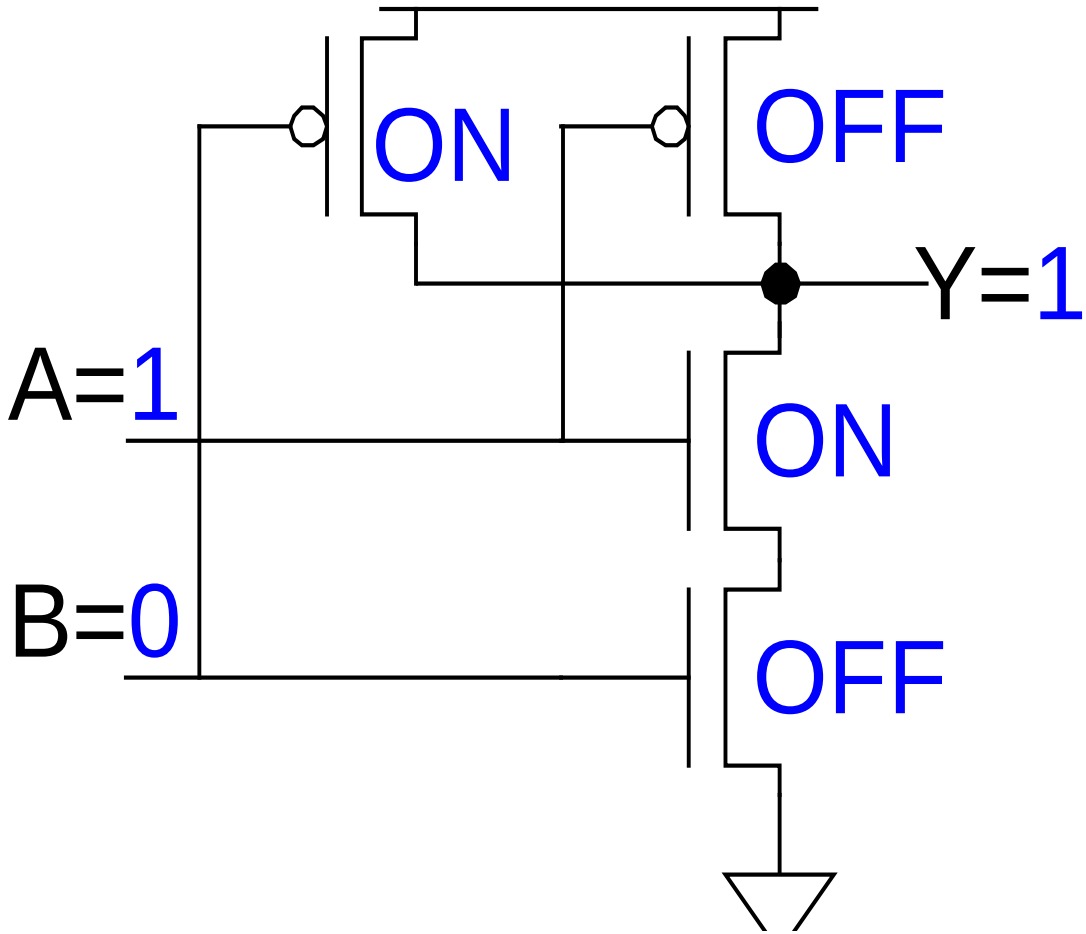




CMOS NAND Gate

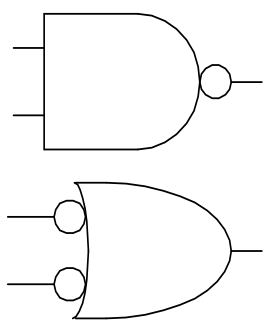
A	B	Y
0	0	1
0	1	1
1	**0**	**1**
1	1	

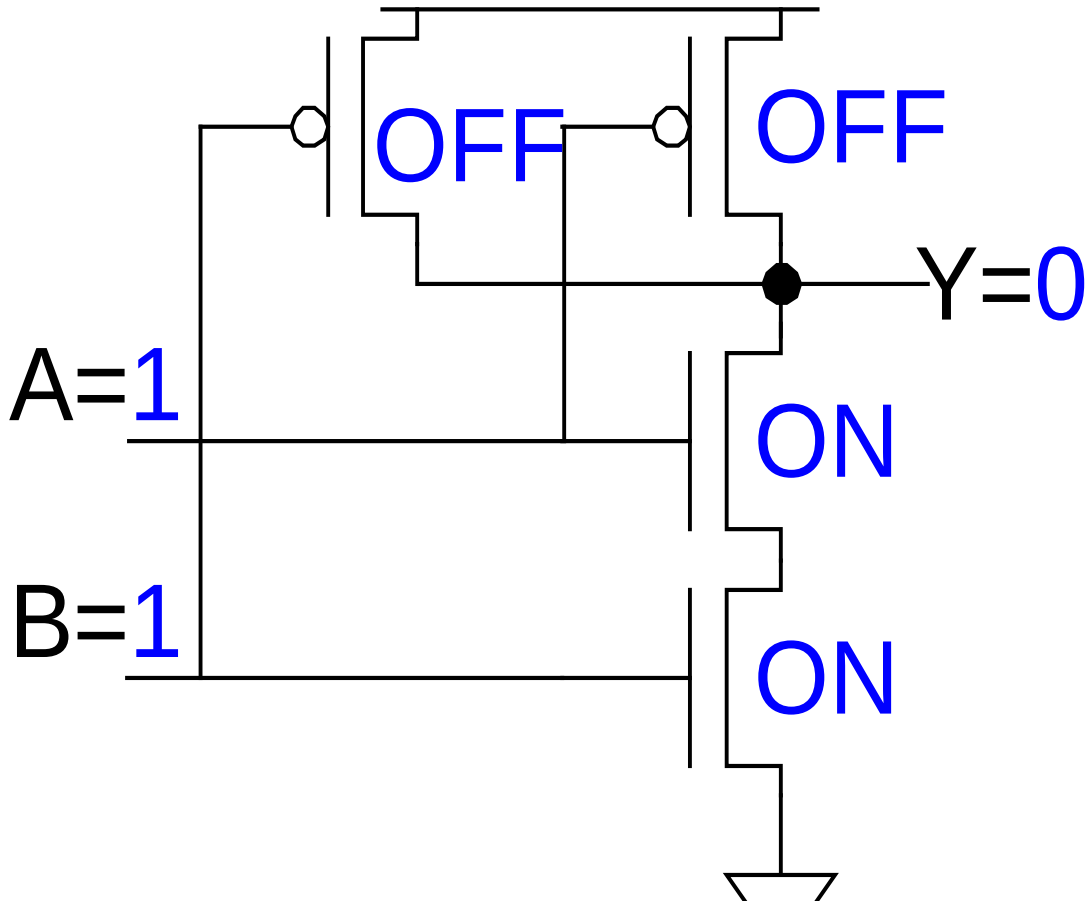




CMOS NAND Gate

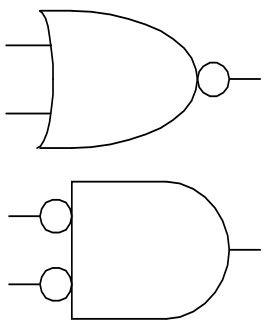
A	B	Y
0	0	1
0	1	1
1	0	1
1	**1**	**0**

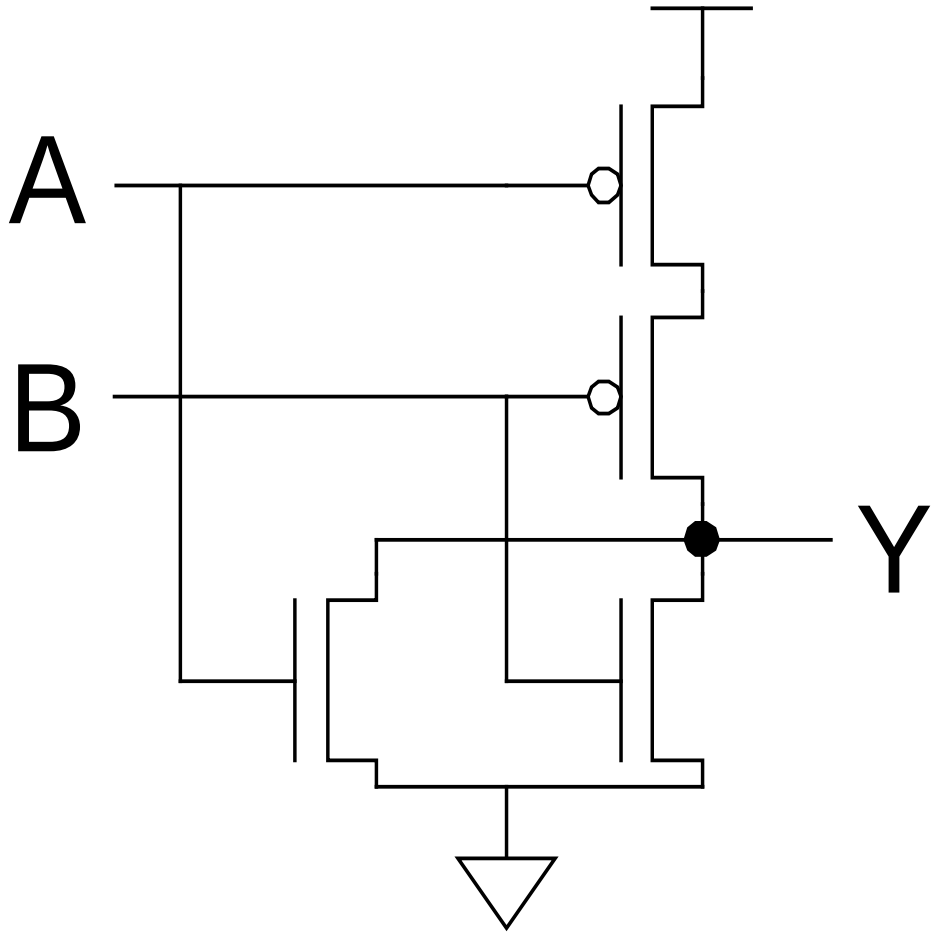




CMOS NOR Gate

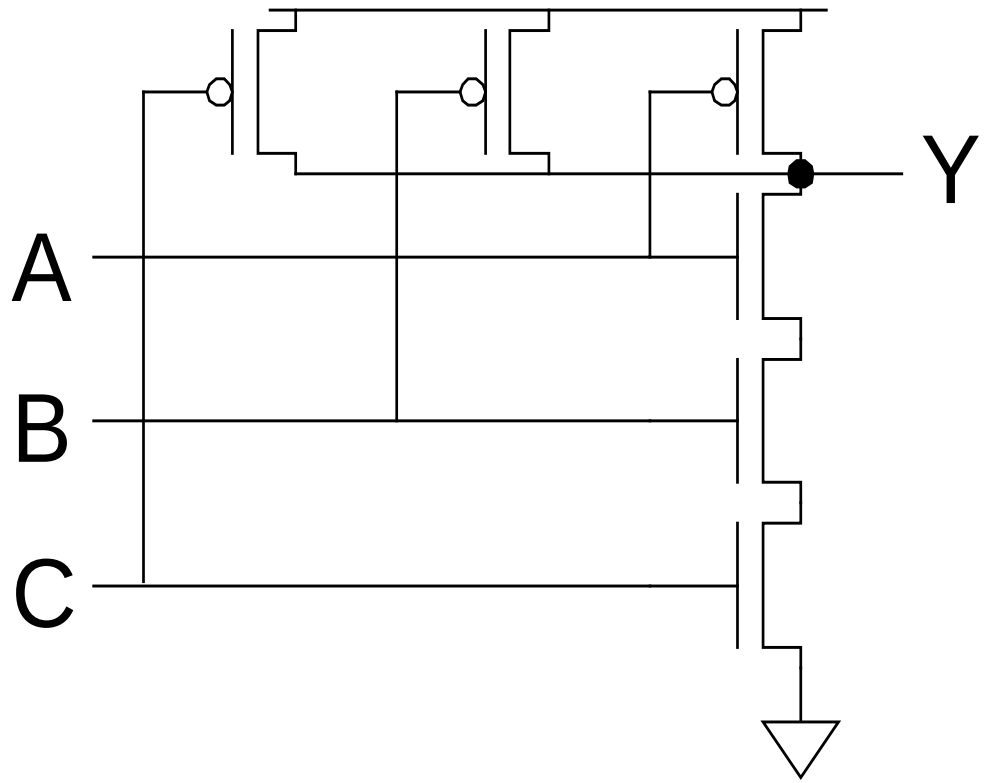
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0





3-input NAND Gate

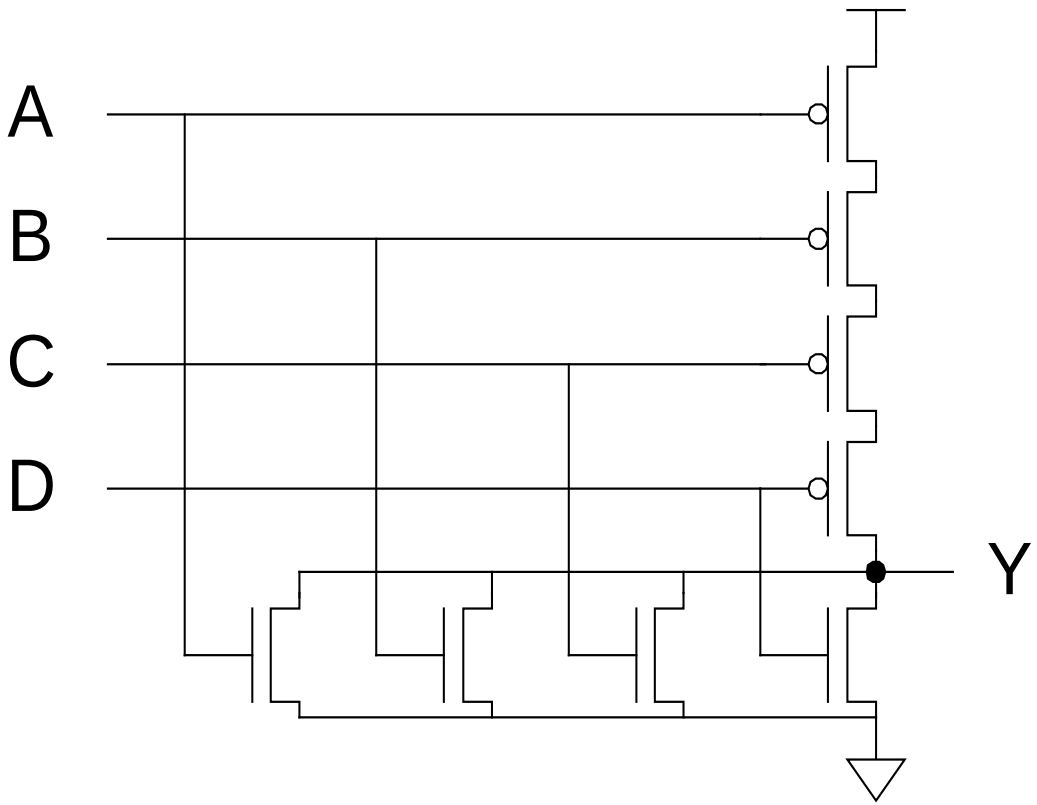
- Y is 0 if and only if ALL inputs are 1
- Y is 1 if and only if AT LEAST one input is 0



Design CMOS Gates

Example

Using the CMOS Technology, draw transistor structure of a 4-input NOR gate



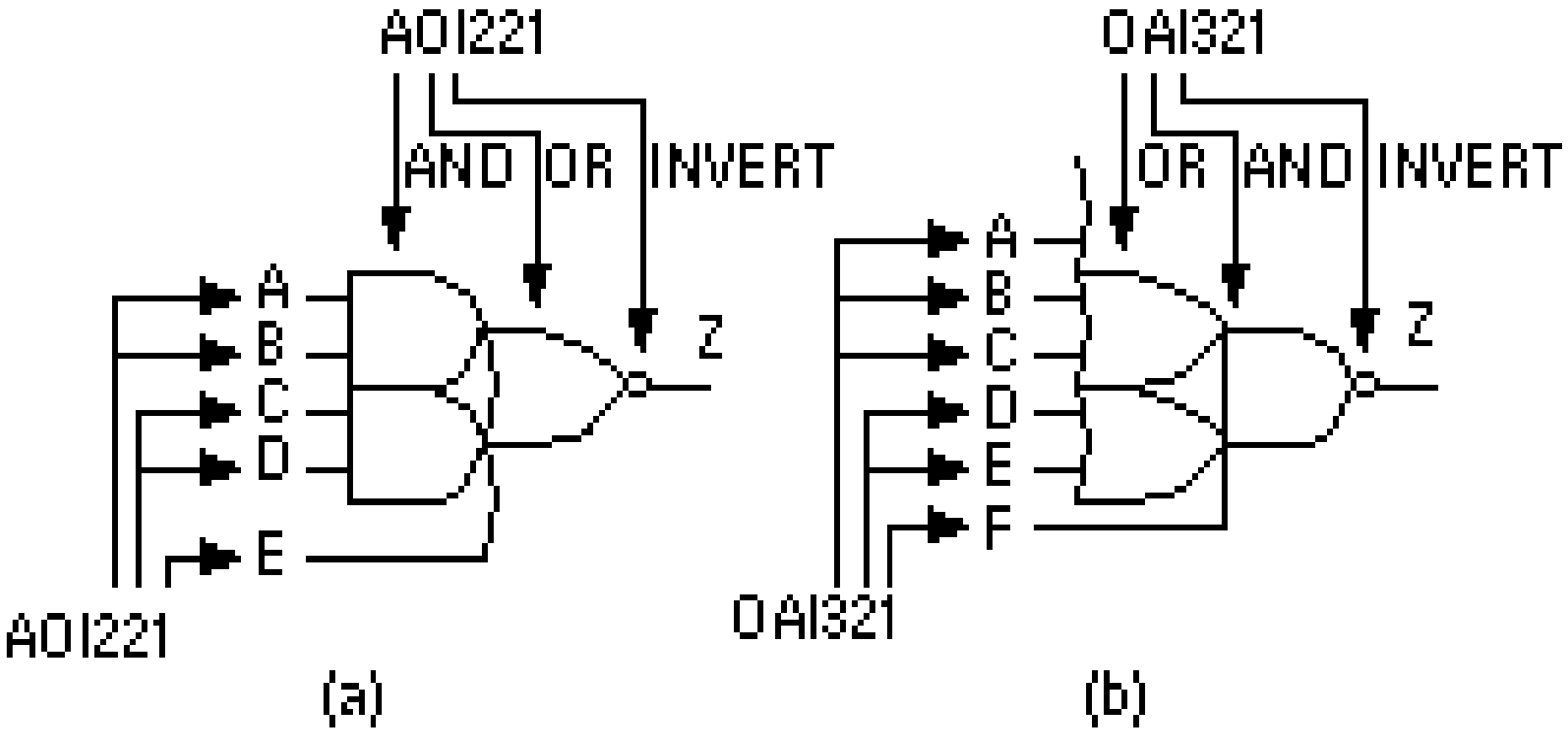
Design CMOS Gate (cont.)

Example 2 (Homework)

Using the CMOS Technology, draw transistor structure of a 4-input NAND gate

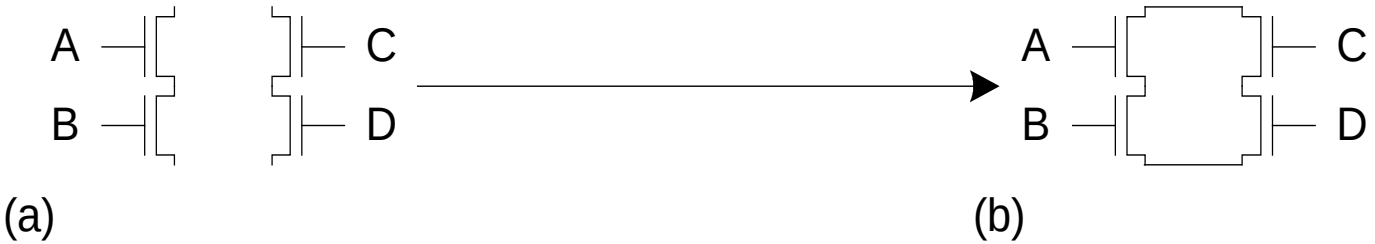
Compound Gates

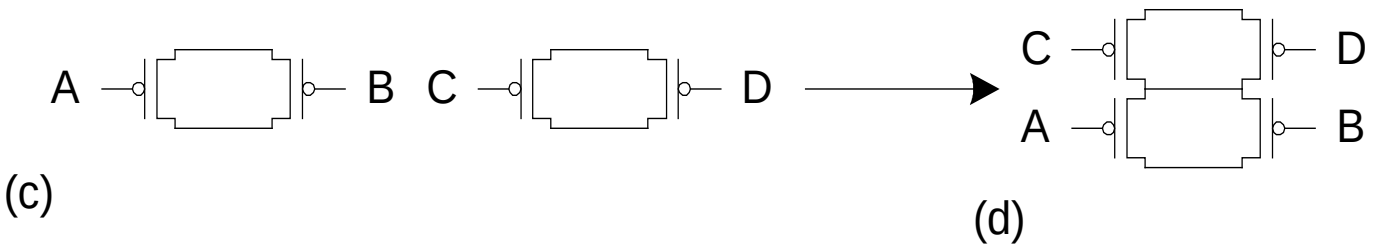
Compound gates: can describe any inverter function (not function)

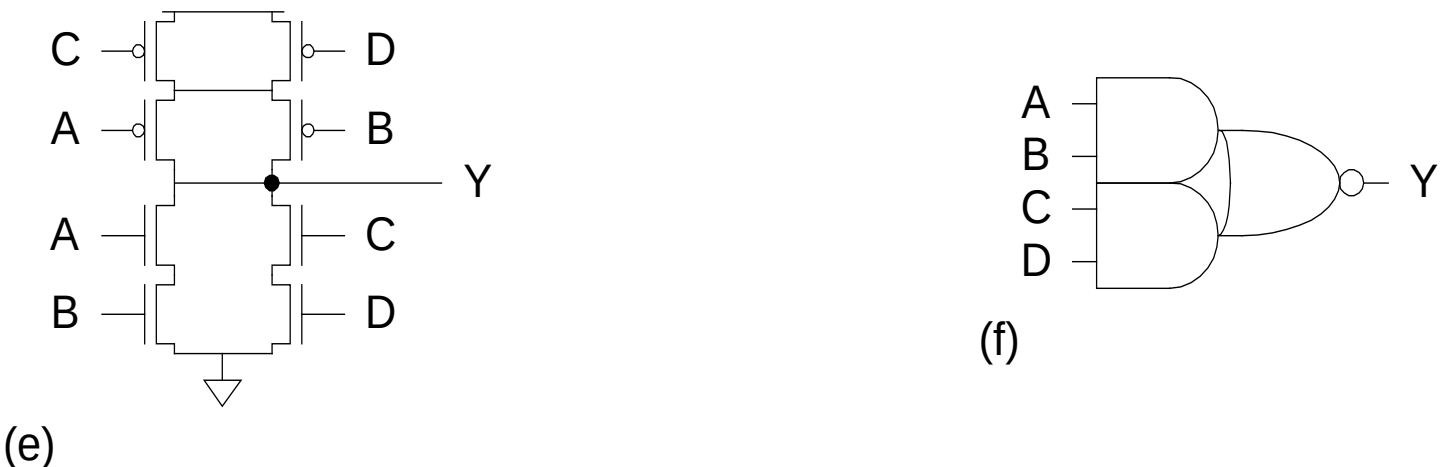


Example: AOI22

$$Y = \overline{(A \bullet B) + (C \bullet D)}$$



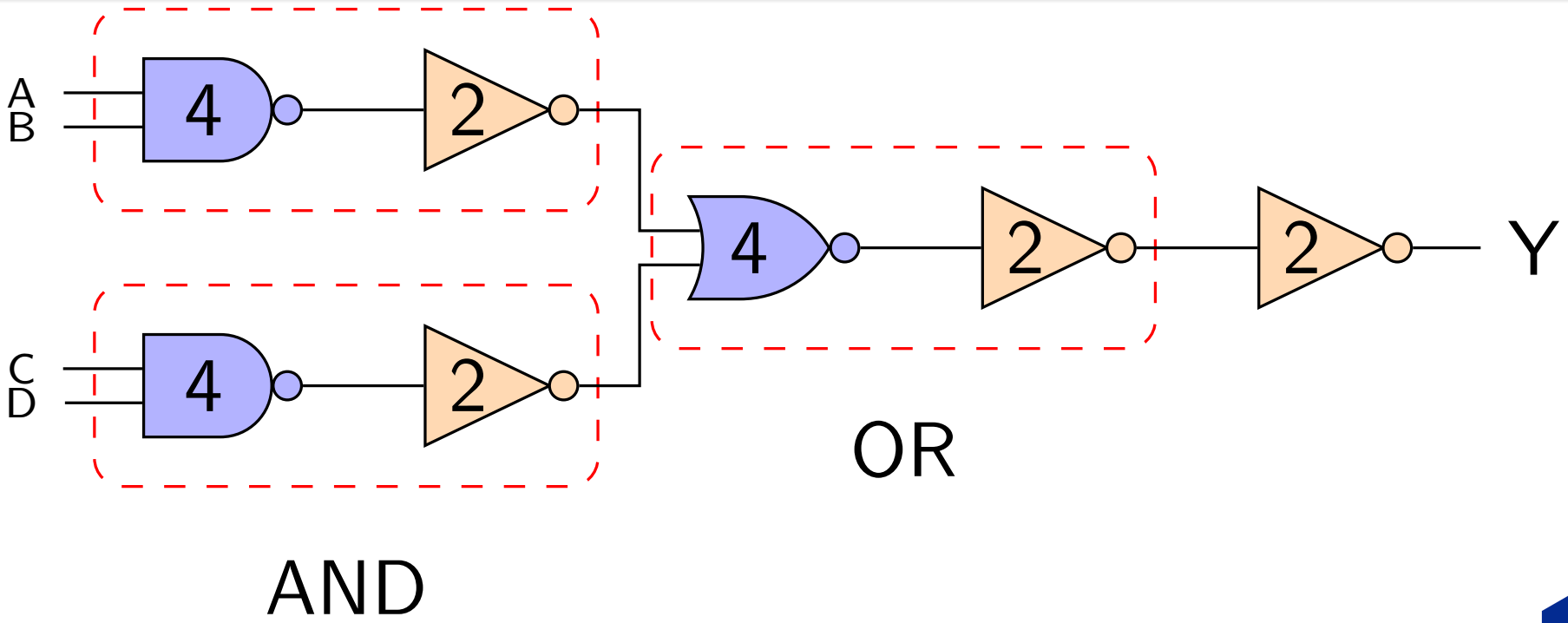




AOI22

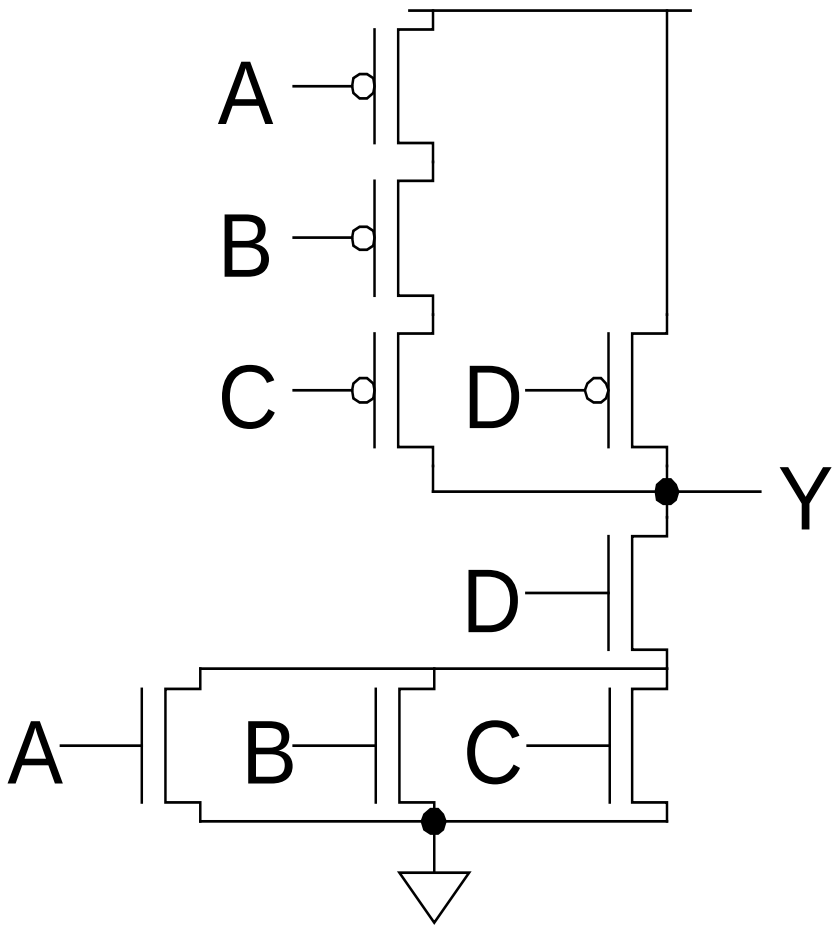
Use AND/OR gate to implement?

- 20 transistors



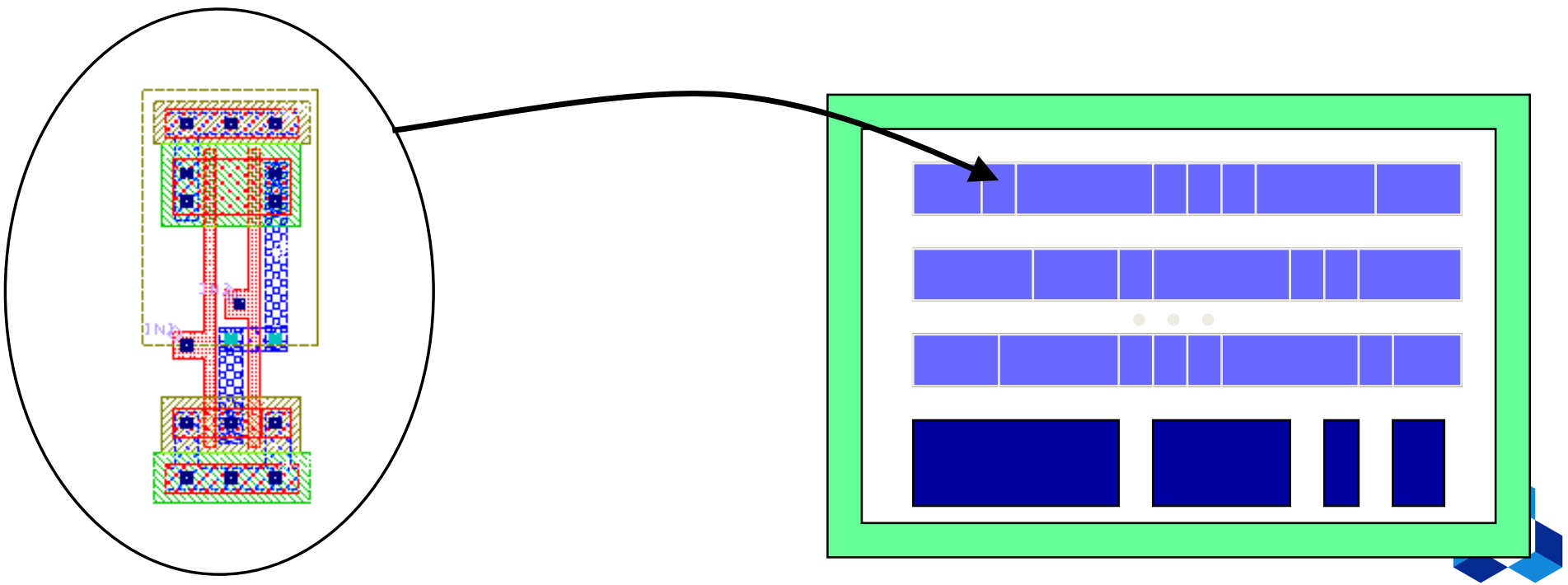
Example: O3AI

$$Y = \overline{(A + B + C) \bullet D}$$



Standard Cells

- Library of common gates and structures (cells)
- Decompose hardware in terms of these cells
- Arrange the cells on the chip
- Connect them using metal wiring



FPGAs

- "Programmable" hardware
- Use small memories as truth tables of functions
- Decompose circuit into these blocks
- Connect using programmable routing
- SRAM bits control functionality