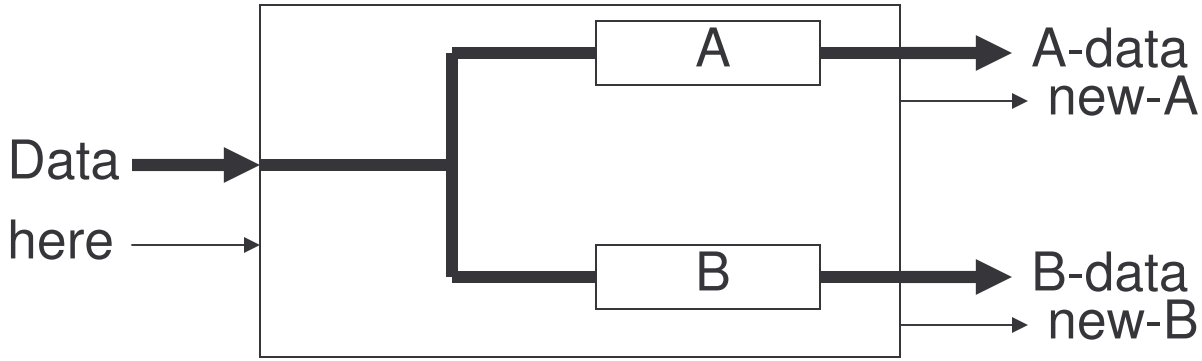
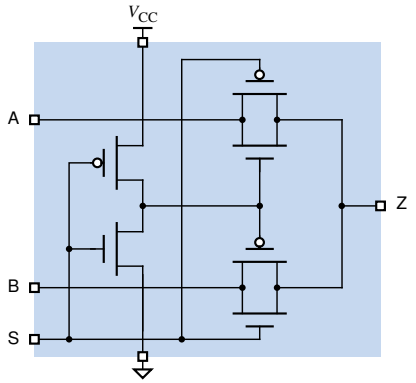


Digital Circuits

ECGR2181

Chapter 3 Notes



Reading: Chapter 3

What is a digital system?

It is a organized collection of digital elements which is designed to perform specified operations on a set of digital inputs and to generate a set of digital responses.

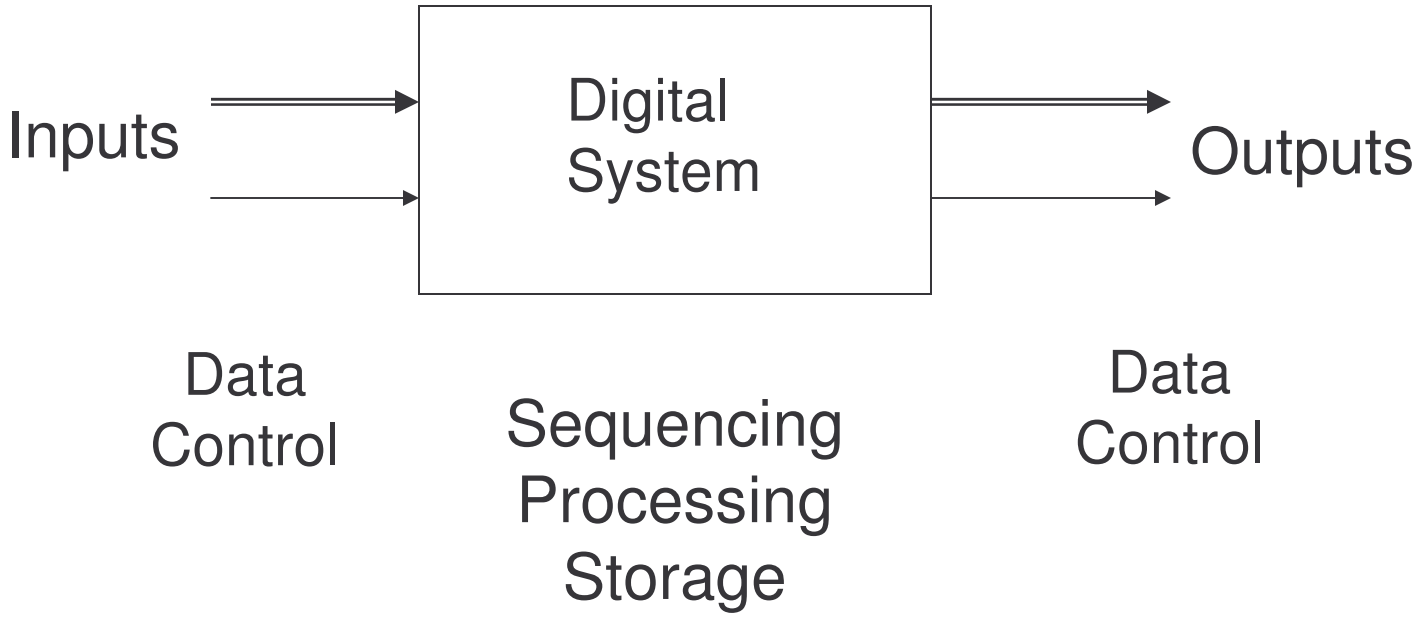
A digital system can be as simple as a block of combinational logic or as complex as a microprocessor.

Characteristics of Digital Systems

What are the characteristics of a digital system?

- Coordinate and sequence its internal operations.
- Data processing and storage.
- Cooperate in transferring data to & from itself.
- Sequences operations of external entities.

Overview of a digital system



Input & Output Signals

Data:

- Multi-bit: “values”
- Single-bit: decision-making / information

Control: {generally single-bit signals}

- Sequencing operations of system
- Coordinating operations with external units

Nomenclature: (Terms to know.)

Word: A group of binary bits. Typically represents some element of data. The number of bits in a word is indeterminate unless specified. [Example: “24-bit word”]

Byte: An 8-bit word.

Nibble: A 4-bit word.

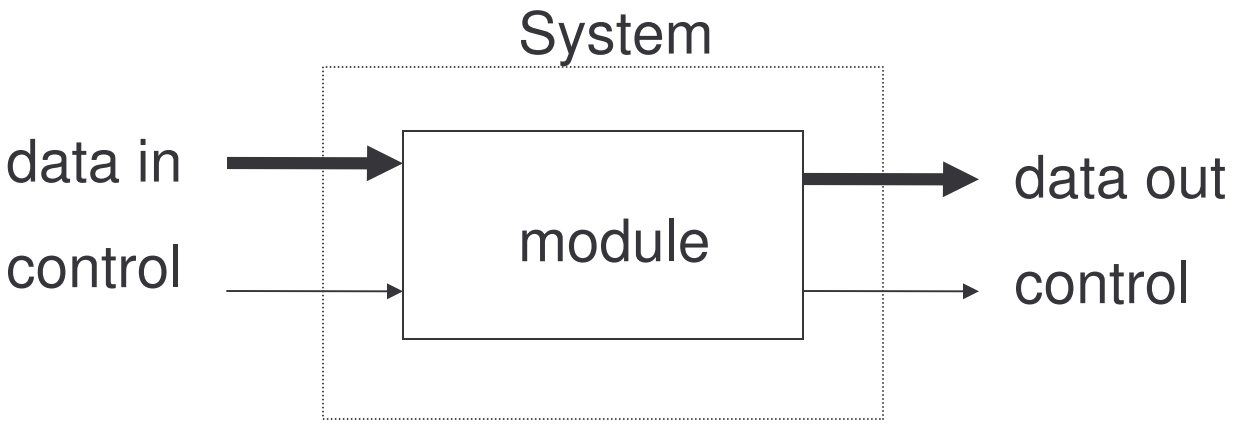
Introduction to Digital Systems

Structure of digital systems: "system" vs. "module"

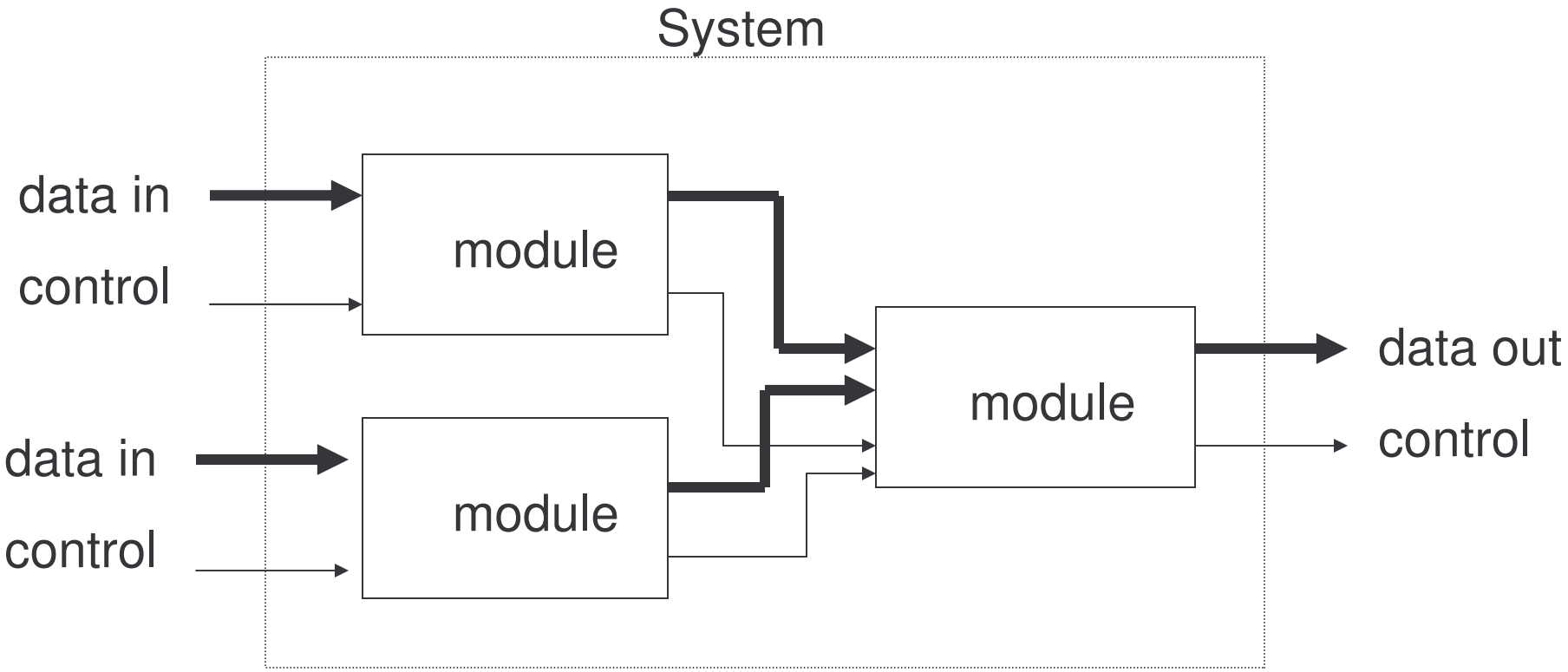
- A digital system can be created as a monolithic structure.
- Complex systems often need to be partitioned into some number of subsystems -- "modules"
- For small systems which can be conveniently designed monolithically the terms "system" and "module" may be used interchangeably.

Introduction to Digital Systems

Single module system:



Multiple module system:



Examples of digital systems

- Data Selector: Route input data to one of two outputs.
- Data Converter: Inputs a 32-bit data word and outputs it as 4 bytes.
- Message Generator: Outputs a fixed message when a "start" command is received
- Communications Buffer: Receives and stores a "block" of data. When the block is complete, it resends the stored data.
- Microprocessor: "Does everything!"

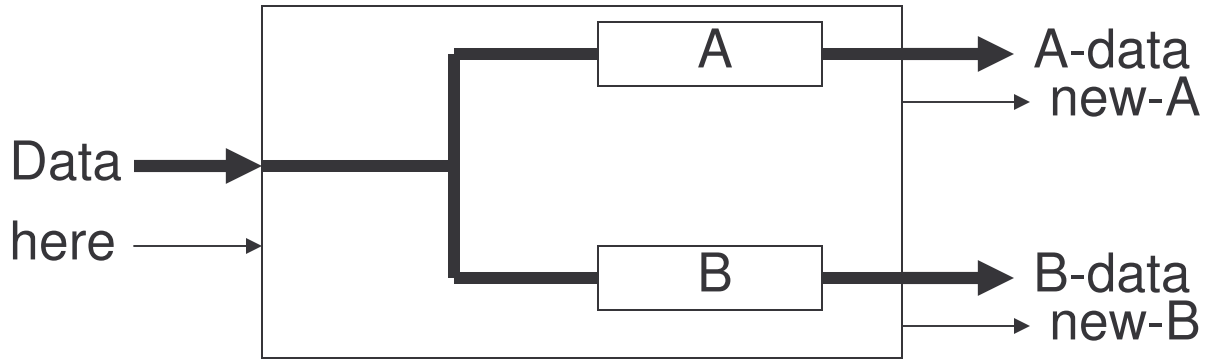
A first look at the design process

1. Understand the functional specification.
2. Create a block diagram from the external viewpoint.
3. Fill in the major internal components.
4. Determine the sequence of operations which must occur within the module

Data Selector

Route input data to one of two outputs.

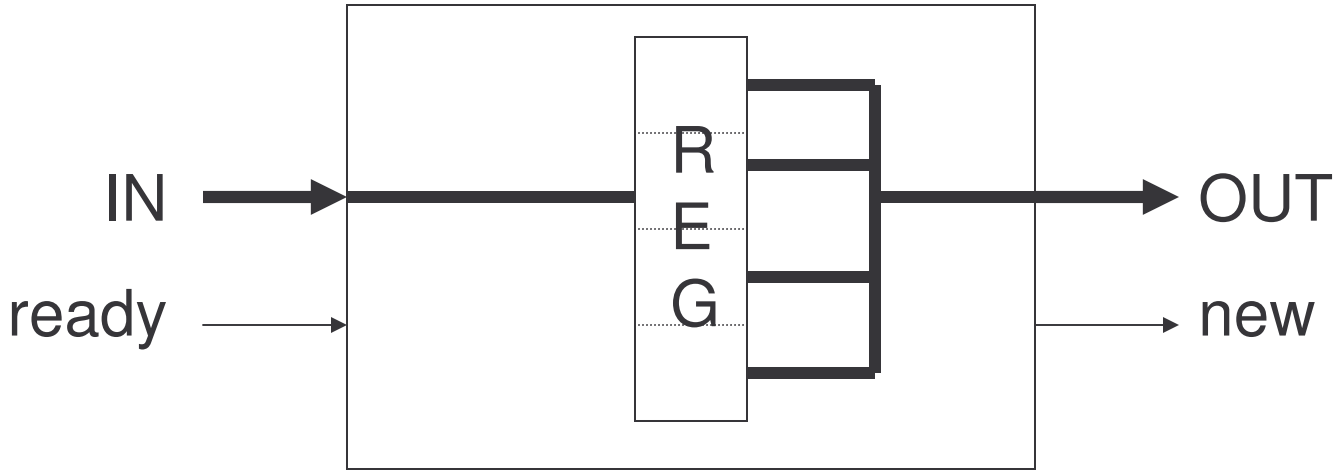
Specification: When a new data word arrives at the input, the module inspects the state of the most significant bit and routes the data to output A if the bit is true and to B if the bit is false. The last value sent to either output is retained until replaced.



Data Converter

Inputs 32-bit data word and outputs it as 4 bytes.

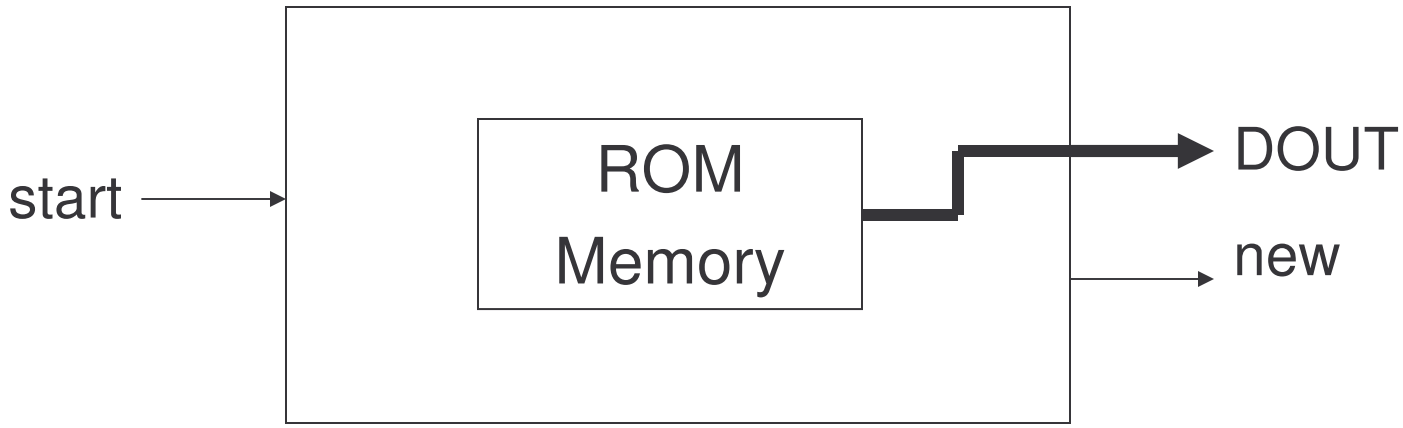
Specification: When a new data word arrives at the input, the module accepts it and then outputs the word as 4 bytes.



Message Generator

Outputs a fixed message when a “start” command is received.

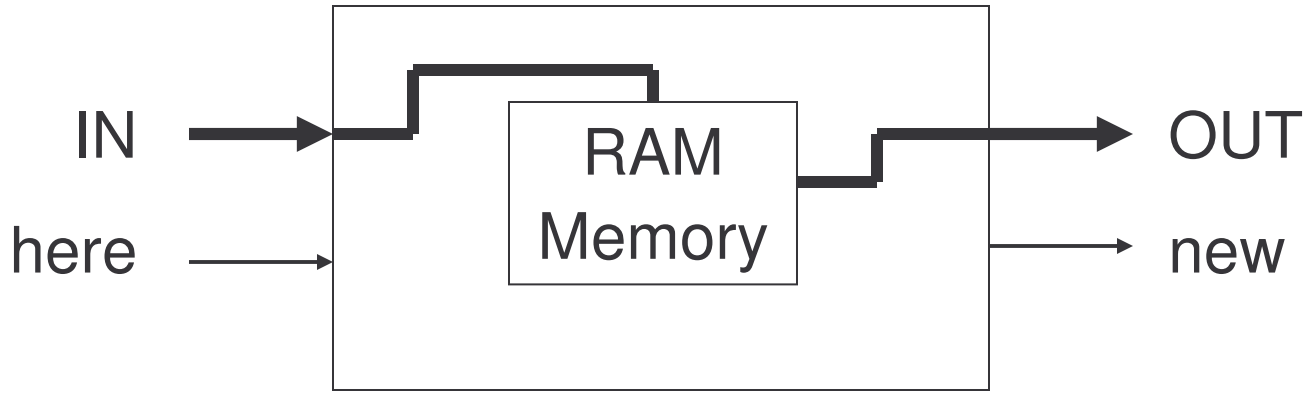
Specification: When a “start” command is received, the module retrieves the bytes of a message stored in an internal ROM and outputs them sequentially.



Communications Buffer

Receives and stores a “block” of data. When the block is complete, it resends the data.

Specification: The module receives a series of data bytes and stores them in an internal memory. Intake of data stops when a byte of all 1’s is received. Then it resends the message with pairs of bytes packed in 16-bit words.

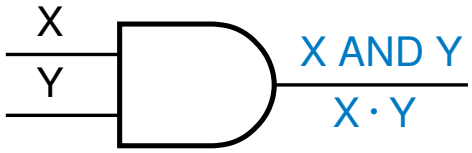


Digital Logic

Binary system -- 0 & 1, LOW & HIGH, negated and asserted.

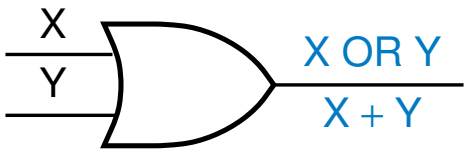
Basic building blocks -- AND, OR, NOT

(a)



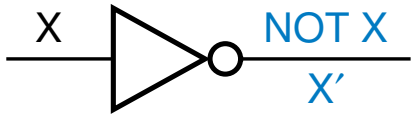
X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

(b)



X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

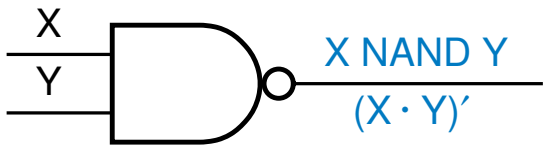
(c)



X	NOT X
0	1
1	0

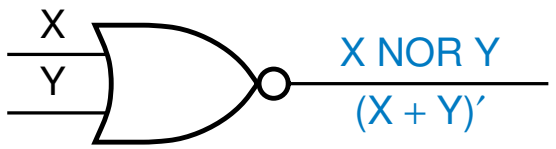
NAND and NOR

(a)



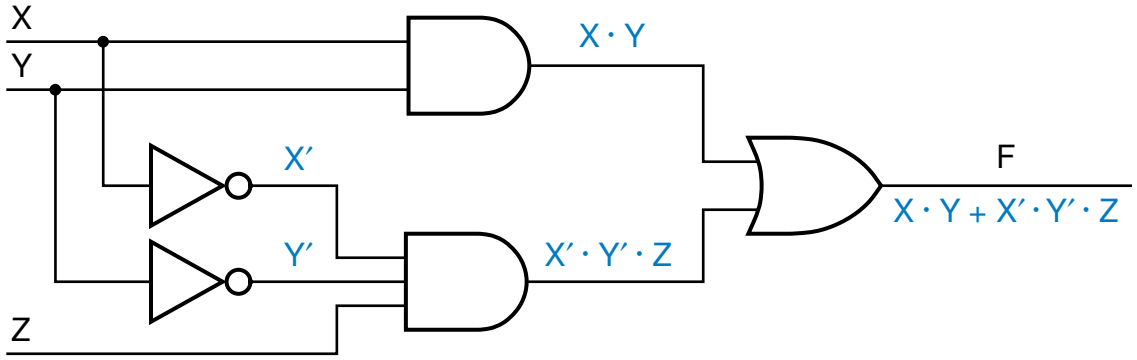
X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0

(b)



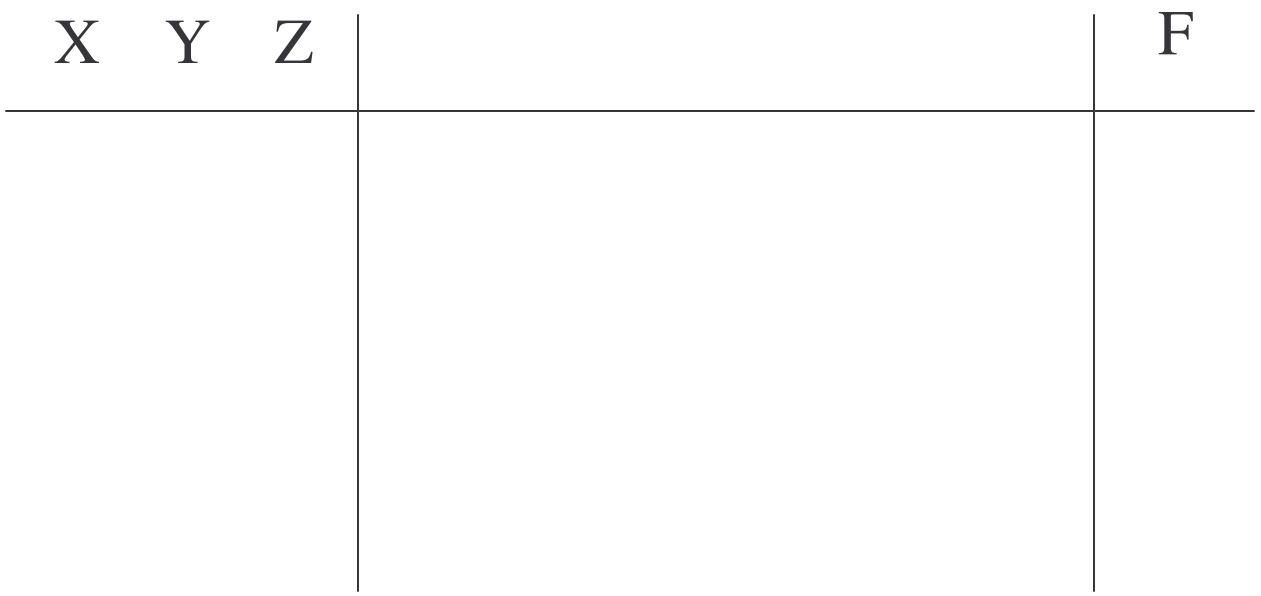
X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0

Truth Tables



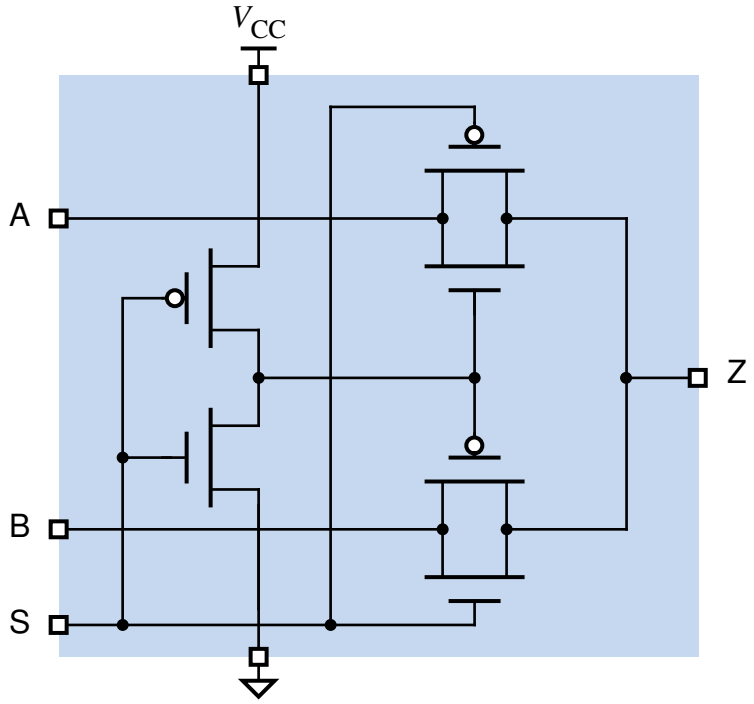
X	Y	Z	XY	X'	Y'	X'+Y'+Z	F

More Practice



Many representations of digital logic

Transistor-level
circuit diagrams



Truth tables

Logic diagrams

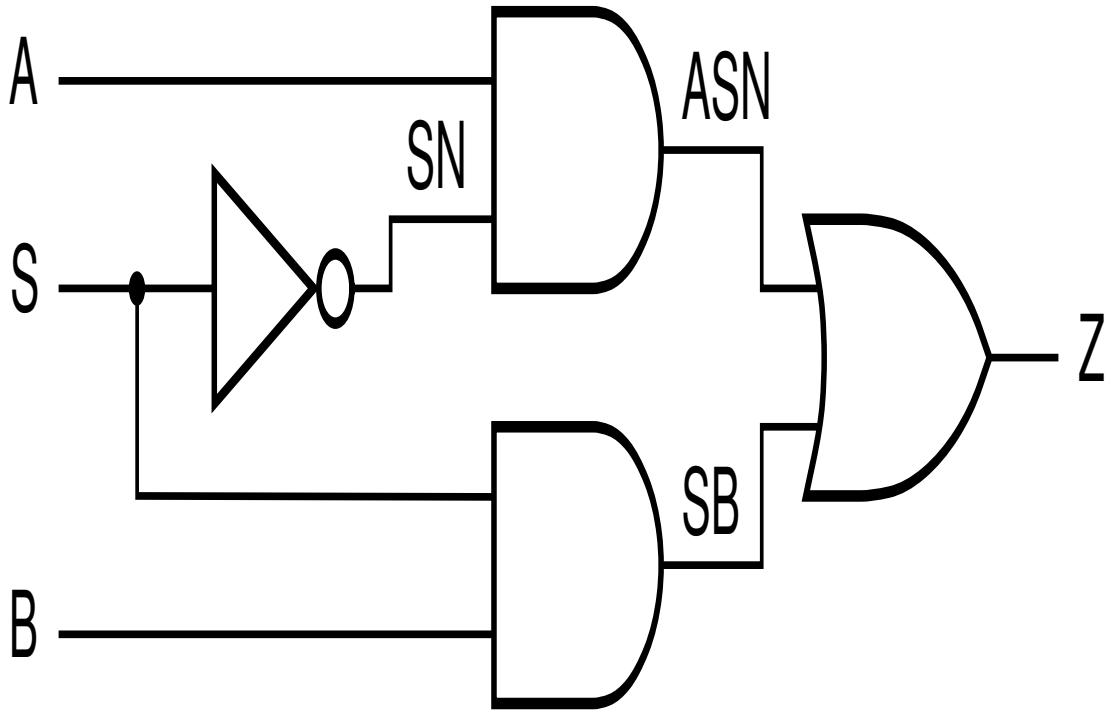
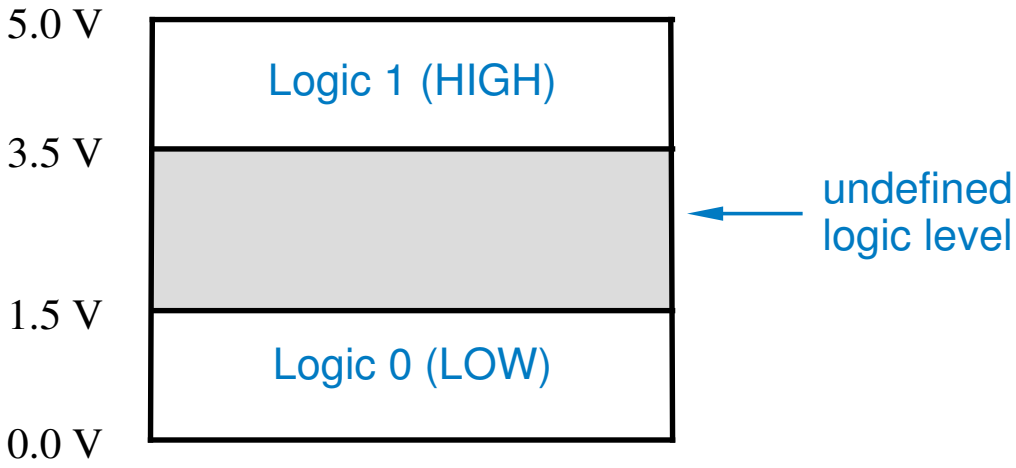


Table 1-1
Truth table for the multiplexer function.

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

Logic levels



Switching threshold varies with voltage, temp, process, etc.

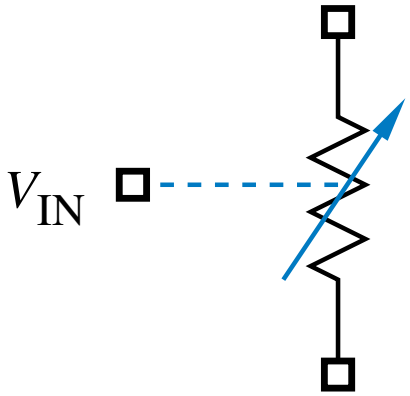
- need “noise margin”

Operating closer to the tolerances requires an increase in attention to “analog” behavior.

Logic voltage levels decreasing with process

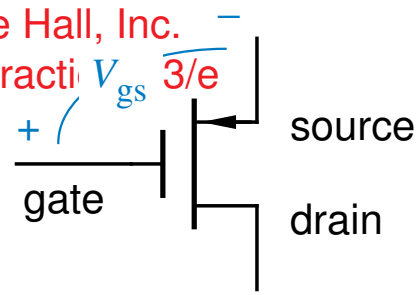
- 5 -> 3.3 -> 2.5 -> 1.8 V

MOS Transistors



Voltage-controlled resistance

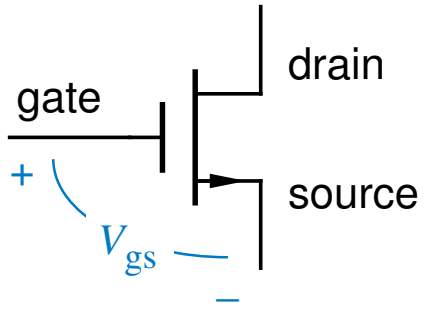
PMOS



Voltage-controlled resistance:
decrease V_{gs} ==> decrease R_{ds}

Note: normally, $V_{gs} \leq 0$

NMOS

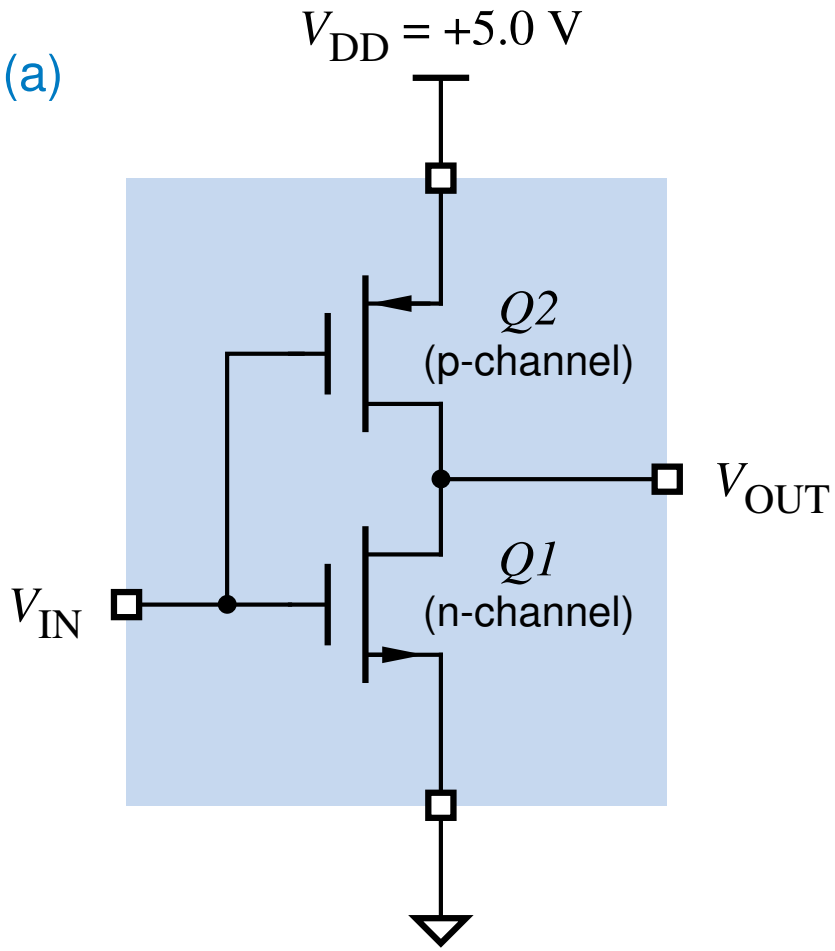


Voltage-controlled resistance:
increase V_{gs} ==> decrease R_{ds}

Note: normally, $V_{gs} \geq 0$

CMOS Inverter

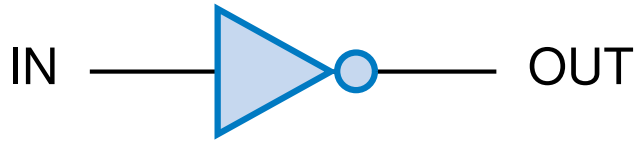
(a)



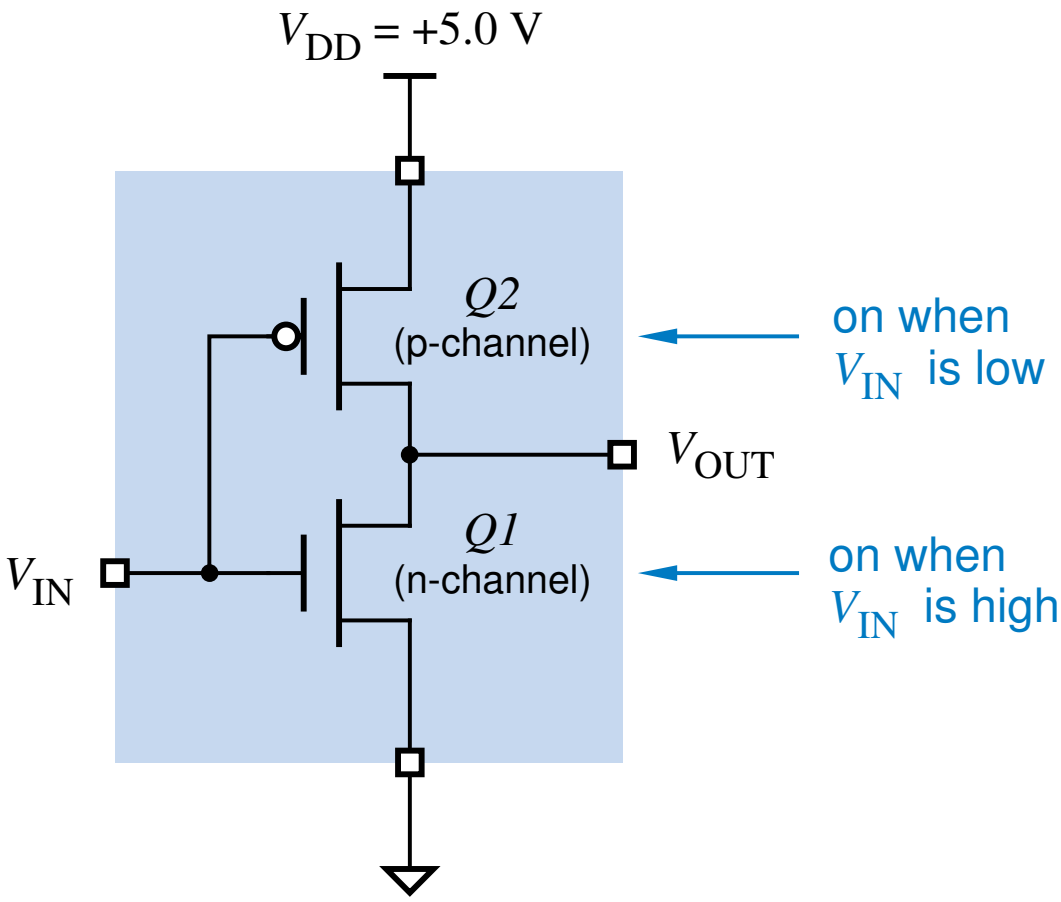
(b)

V_{IN}	*Q1*	*Q2*	V_{OUT}
0.0 (L)	off	on	5.0 (H)
5.0 (H)	on	off	0.0 (L)

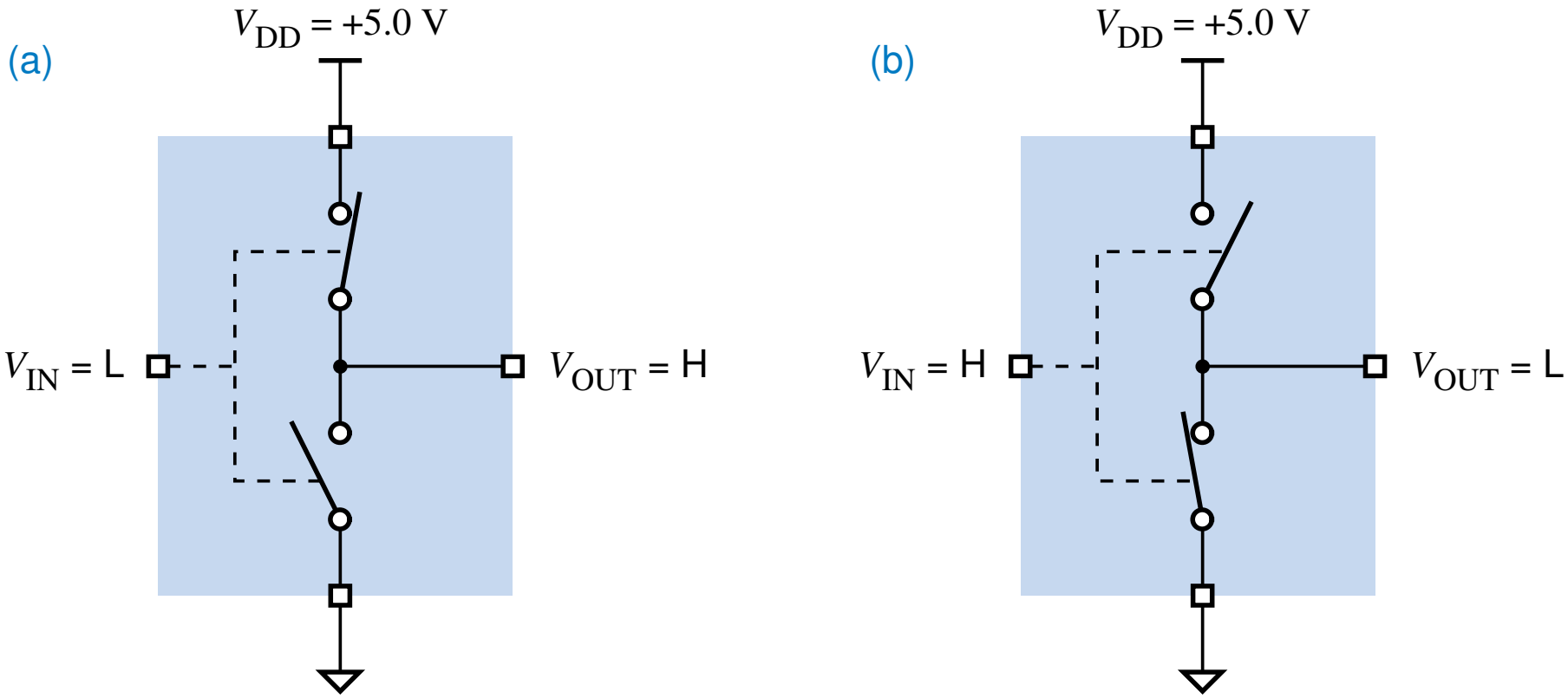
(c)



Alternate transistor symbols



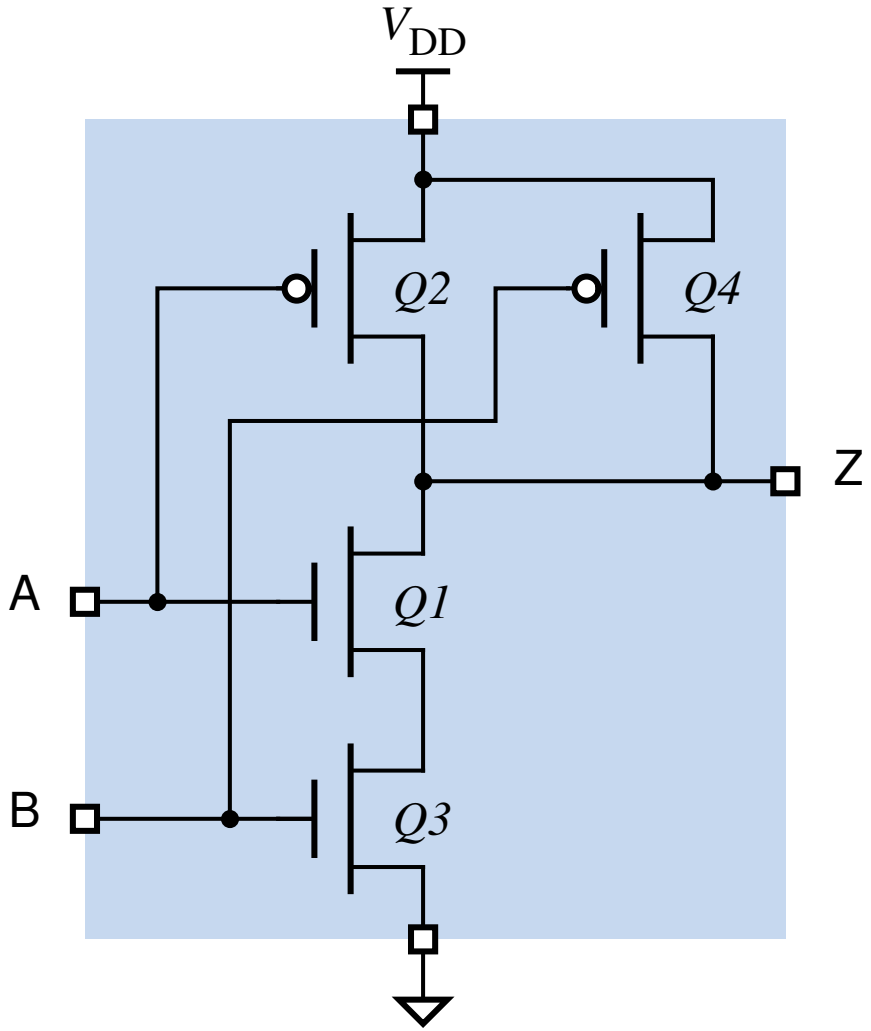
Switch model



CMOS NAND Gates

Use 2*n* transistors for *n*-input gate

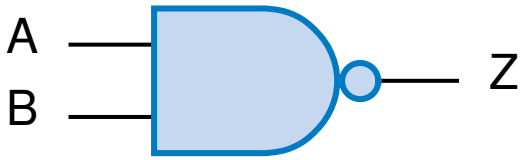
(a)



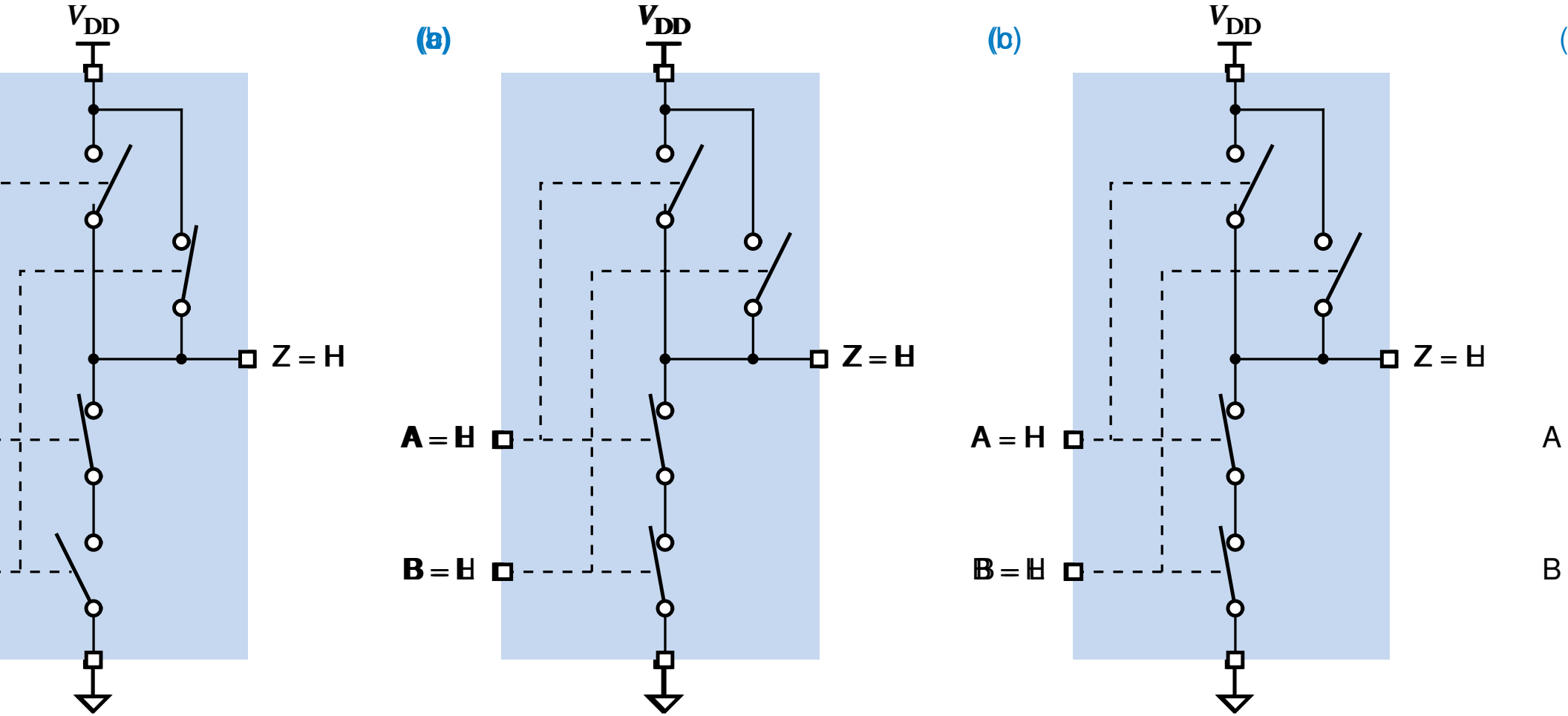
(b)

A	B	*Q1*	*Q2*	*Q3*	*Q4*	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	H
H	L	on	off	off	on	H
H	H	on	off	on	off	L

(c)

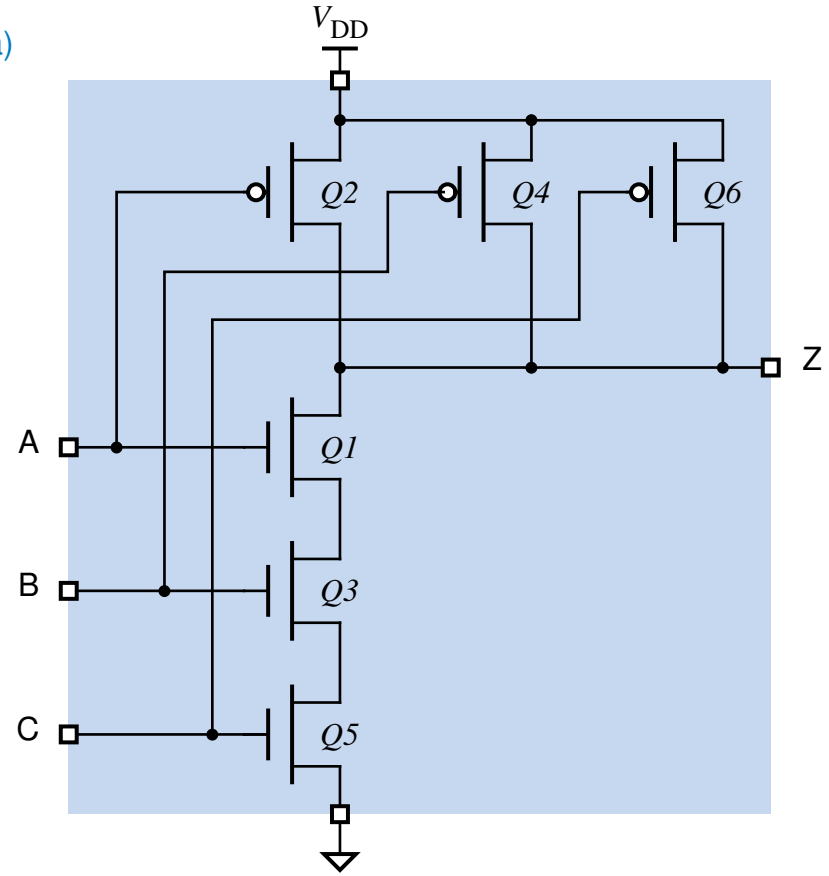


CMOS NAND -- switch model



CMOS NAND -- more inputs (3)

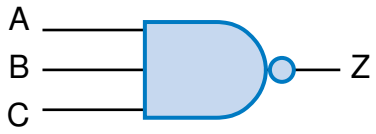
(a)



(b)

A	B	C	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	Z
L	L	L	off	on	off	on	off	on	H
L	L	H	off	on	off	on	on	off	H
L	H	L	off	on	on	off	off	on	H
L	H	H	off	on	on	off	on	off	H
H	L	L	on	off	off	on	off	on	H
H	L	H	on	off	off	on	on	off	H
H	H	L	on	off	on	off	off	on	H
H	H	H	on	off	on	off	on	off	L

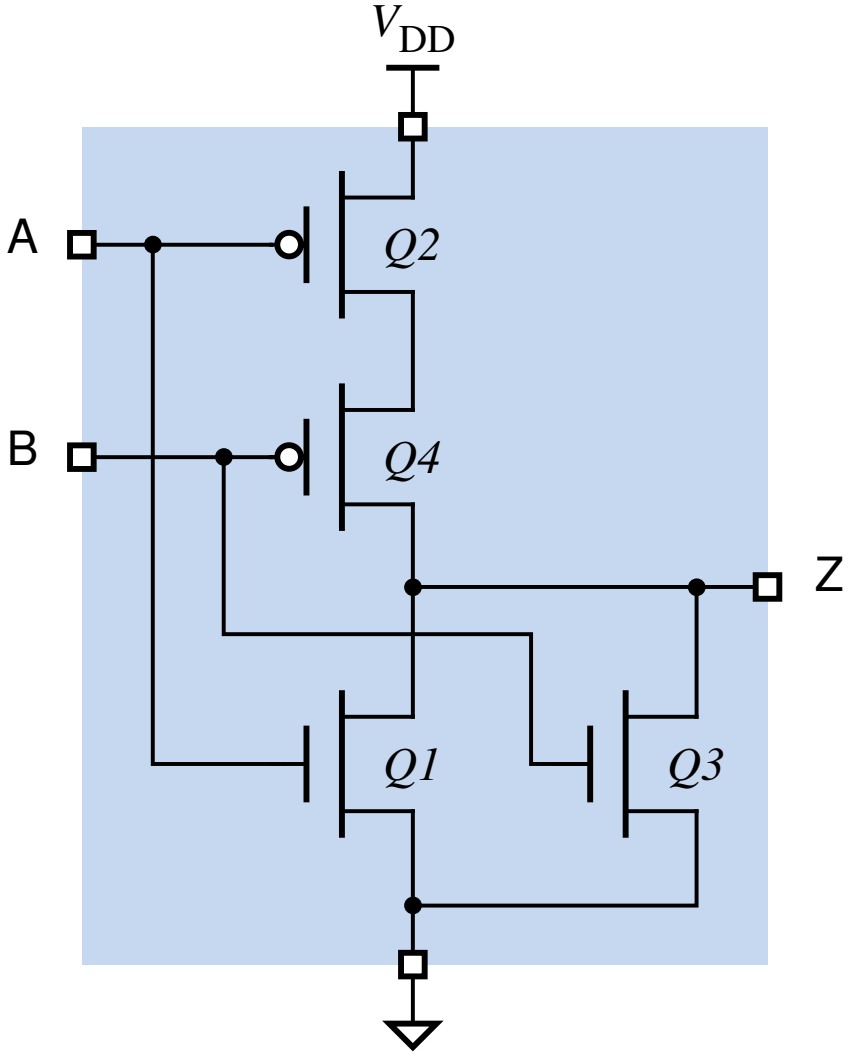
(c)



CMOS NOR Gates

Like NAND -- 2*n* transistors for *n*-input gate

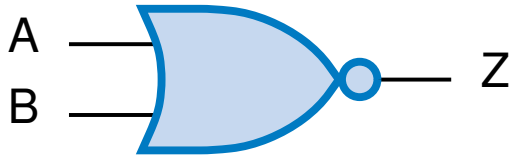
(a)



(b)

A	B	*Q1*	*Q2*	*Q3*	*Q4*	Z
L	L	off	on	off	on	H
L	H	off	on	on	off	L
H	L	on	off	off	on	L
H	H	on	off	on	off	L

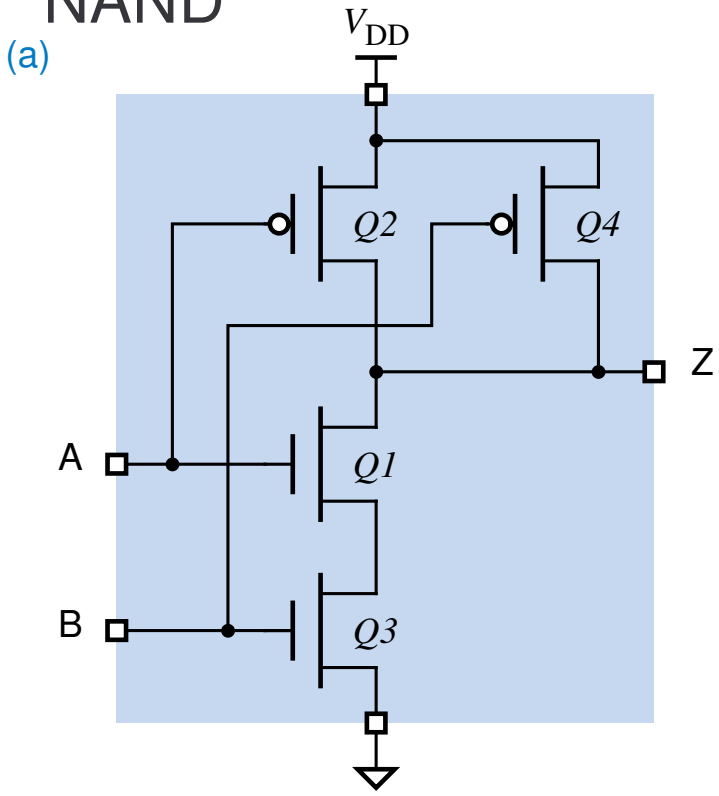
(c)



NAND vs. NOR

PMOS transistors have higher “on” resistance than NMOS transistors.

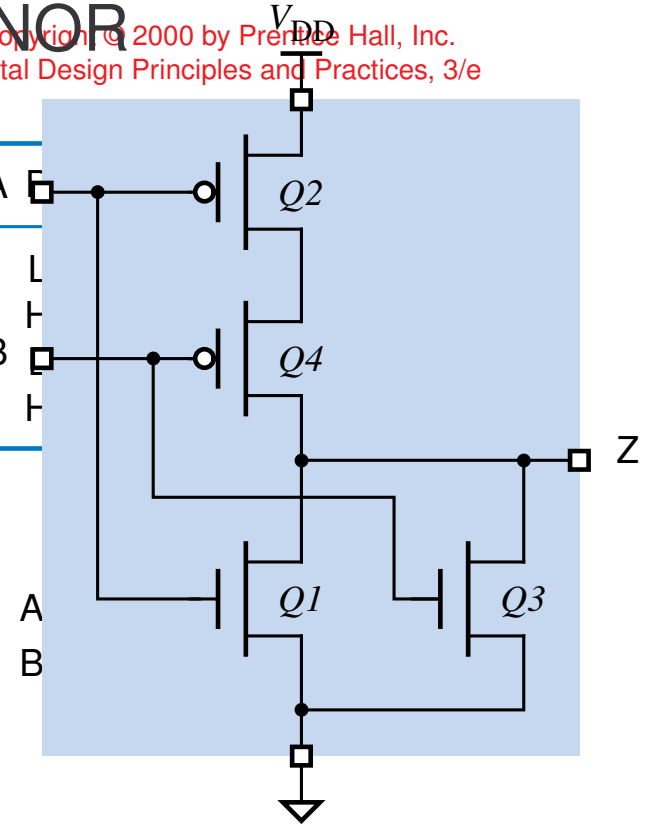
NAND



(b)

A	B
L	L
L	H
H	L
H	H

(c)



NOR

(a)

(b)

A	B
L	L
L	H
H	L
H	H

(c)

Result: NAND gates are preferred in CMOS.