EE321

Computer Architecture

Chap. 01 : Review of Digital Design

Dr. Abdelhakim Khouas

Email : akhouas@hotmail.com

ab.khouas@univ-boumerdes.dz

IGEE (ex. INELEC)

University M'hamed Bougara of Boumerdes

Course chapters

1. Review of Digital Design
2. Top Level of Computer
3. Central Processing Unit (CPU)
4. Control Unit
5. Memory
6. Instruction Set and Addressing Modes
7. Input/Output Devices

Lecture Outline

1. Digital systems
2. Binary representations
3. Combinational logic circuits
4. Sequential circuits
5. Logic families

1. Digital Systems

What is Digital Circuit (Logic Circuit)?

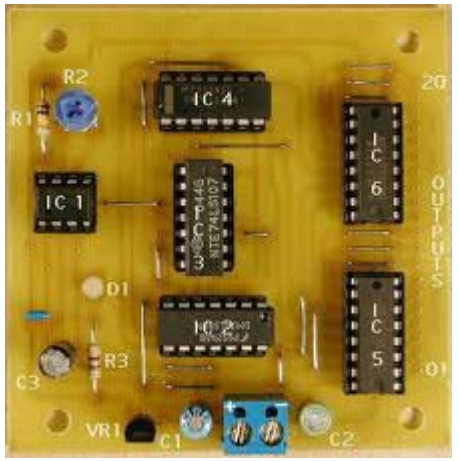
- To solve some everyday problems, we use electronic circuits.
- Two large groups of electronic circuits:

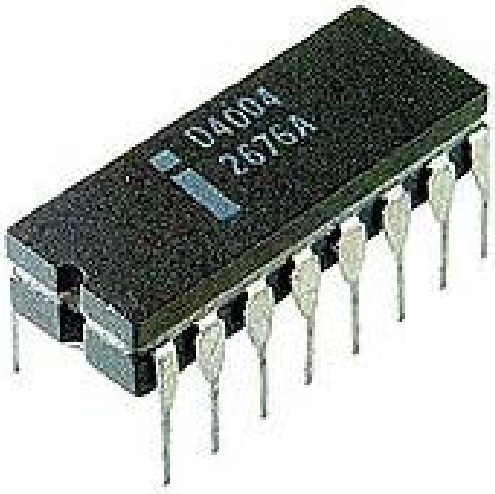
1. Analog circuits: continues-valued signals
2. Digital (logic) circuits: discrete-values signal
   - Word is analog, we use DAC and ADC to convert analog and digital signals

1. Digital Systems

Different designs to solve the same problem:

1. Software design: We write a program than can be executed by a processor (uprocessor, ucontroller, DSP, GPU, ... etc.)
2. Hardware Design: We design electronic circuit
    1. Printed-Circuit Board (PCB)
    2. Application Specific Integrated Circuit (ASIC or IC)
    3. Programmable Logic Devices (PLD, CPLD, FPGA, ...)
3. Hardware/Software Design





1. Digital Systems

Why Digital Circuit?

1. Easy to design
2. High performances
3. Robustness to noise
4. Low cost
5.
6.
7.

1. Digital Systems

Bits and Bytes

- In digital systems we use a two-level logic component called a bit to represent digital signal
  - One bit can take two logic values: ‘0’ and ‘1’ or High (H) and Low (L)

1. Digital Systems

Bits and Bytes

- A group of bit is called bit-vector
- Leftmost bit is referred to as MSB (Most Significant Bit)
- Rightmost is LSB (Least Significant Bit)
- A group of 8 bits (10011101) is called Byte
- 1 Kbyte = 1024 Bytes = 8192 bits
- 1 Mbyte = 1024 Kbyte = 1048576 Bytes
- 1 Gbyte = 1024 Mbyte

Logic value vs. Physical value

- Logic values ‘L’ and ‘H’ are represented in the real circuit by physical values
- In fact, the physical values for ‘L’ and ‘H’ are ranges of values
  - VOH: minimum output voltage for High
  - VOL: maximum output voltage for Low
  - VIH : minimum input voltage guaranteed to be High
  - VIL : maximum input voltage guaranteed to be Low

1. Digital Systems

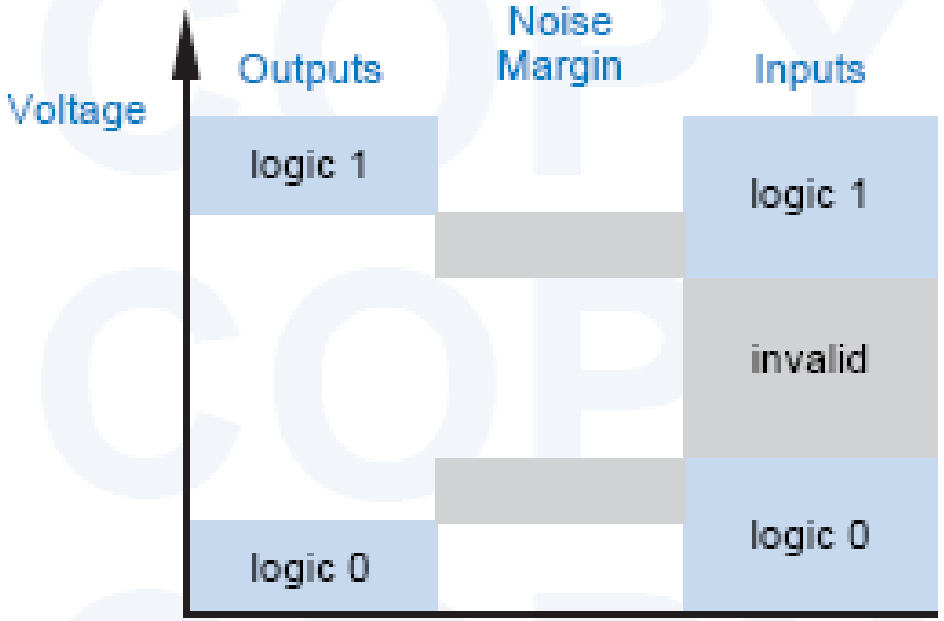
Logic value vs. Physical value

- OH = [VOH,VDD]
- OL= [0 , VOL]
- IH = [VIH,VDD]
- IL = [0,VIL]

Example:

5V HCMOS

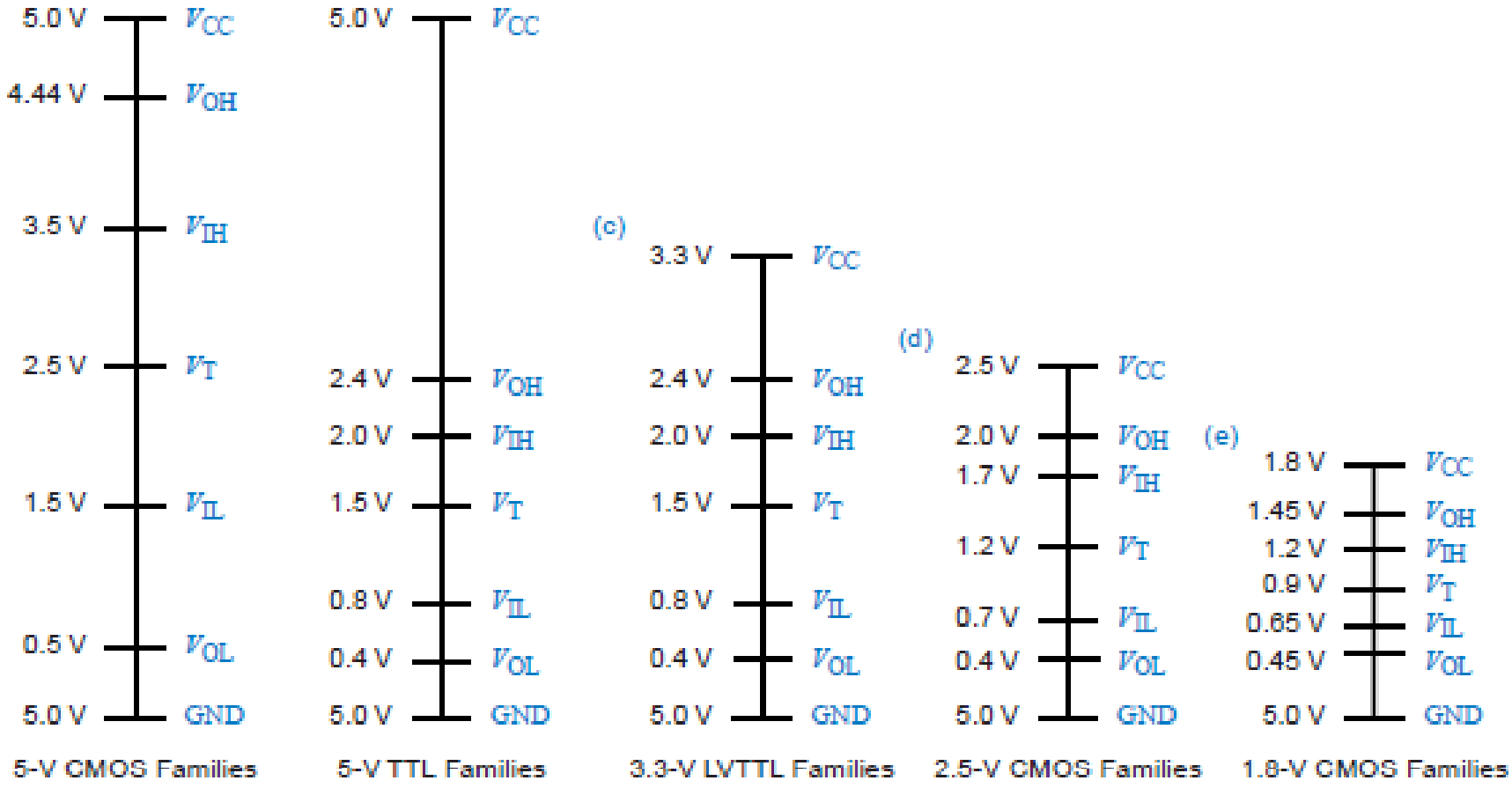
- VOL = 0.26 V
- VOH = 4.48 V
- VIL = 1V
- VIH = 3.5V
- Noise Margin L = 0.74
- Noise Margin H = 0.96V



Source: Digital Design Principles and practices, by J. F. Wakerly

1. Digital Systems

Comparison of logic levels of some technologies



Source: Digital Design Principles and practices, by J. F. Wakerly

2. Binary Representation

Positional Number Systems

- In positional number system, each position has a weight based on powers of the base (e.g. 57 = 5x10+7)

Base 2

$$N = a_k a_{k-1} \ldots a_1 a_0 \big|_2 = \sum_{i=0}^{k} a_i * 2^i \qquad avec \qquad 0 \le a_i \le 1$$

- 1101 = 1+0+4+8 = 13

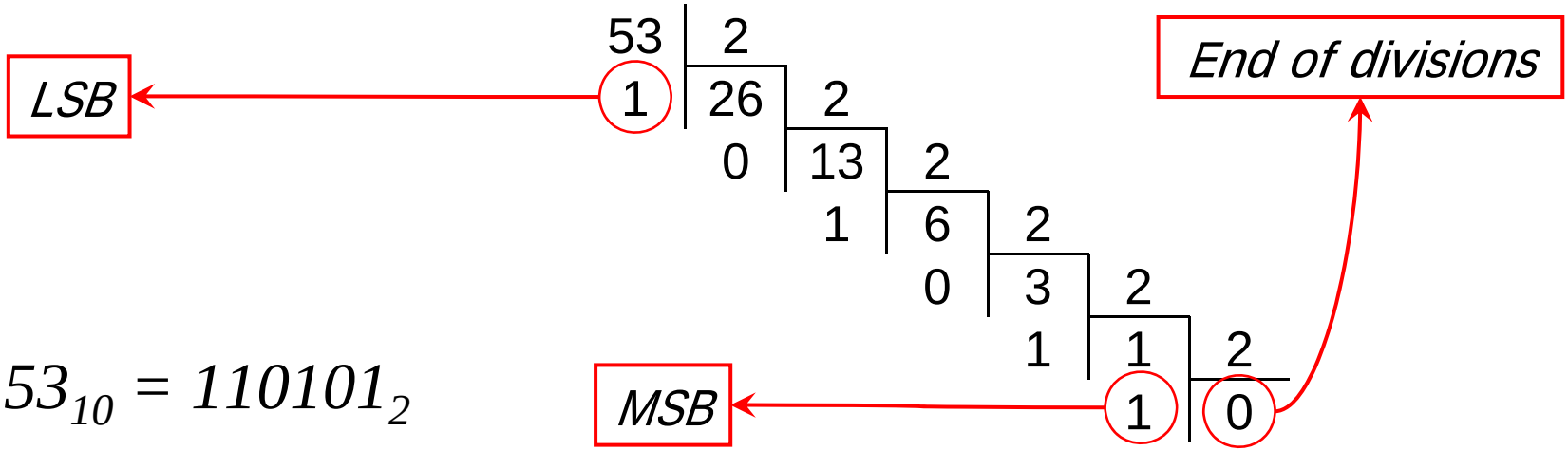
Hexadecimal Numbers (base 16)

- Bits are grouped by 4
- Hex is used just to simplify Human-system communication

2. Binary Representation

Conversion: Decimal --> Base 2

- Division successives par 2

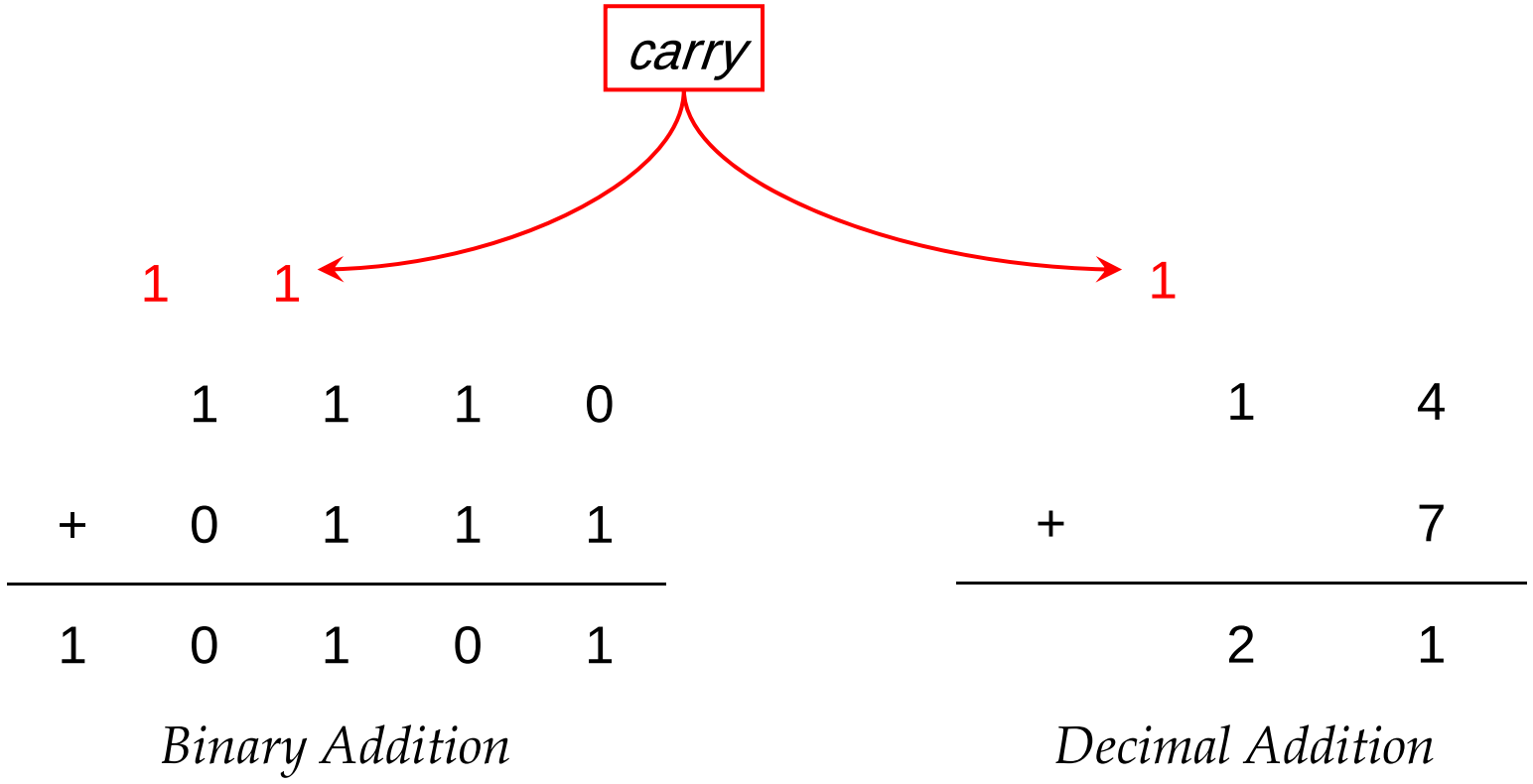


$53_{10} = 110101_2$

Conversion: Base 2 -->Decimal

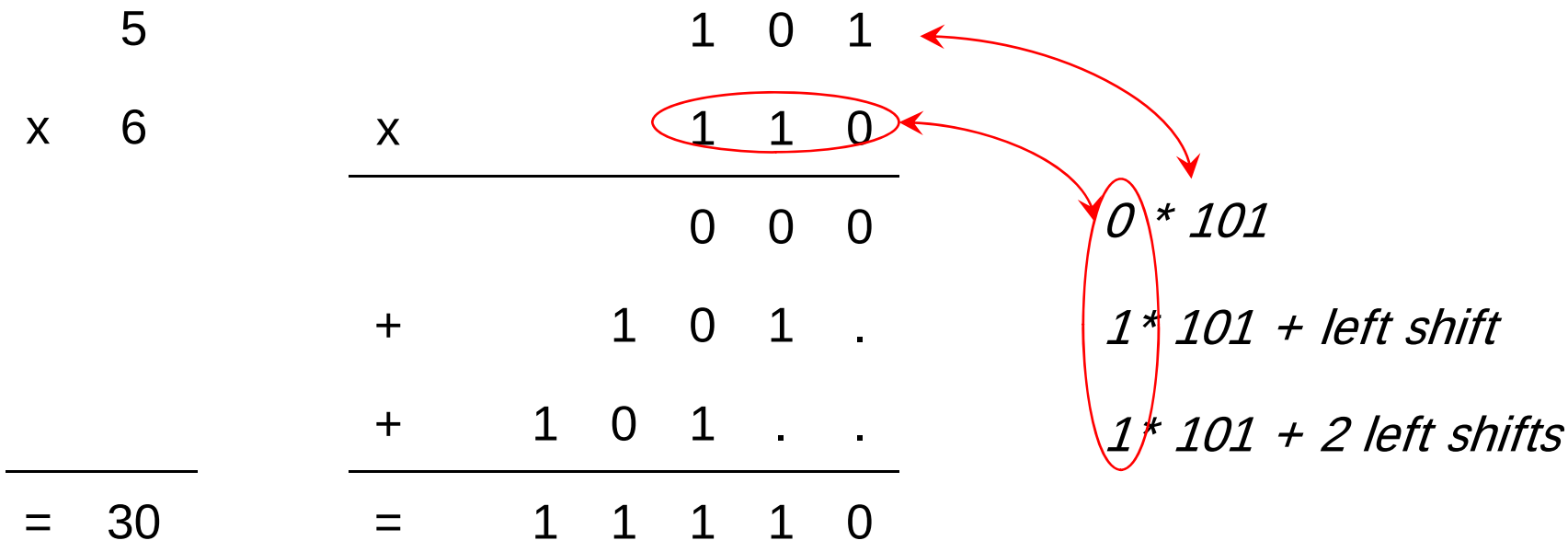
- Multiplication by power of 2
- $1101_2 = 1*2^0 + 0*2^1 + 1*2^2 + 1*2^3 = 1+4+8 = 13_{16}$

2. Binary Representation



Binary addition

2. Binary Representation



Binary multiplication

2. Binary Representation

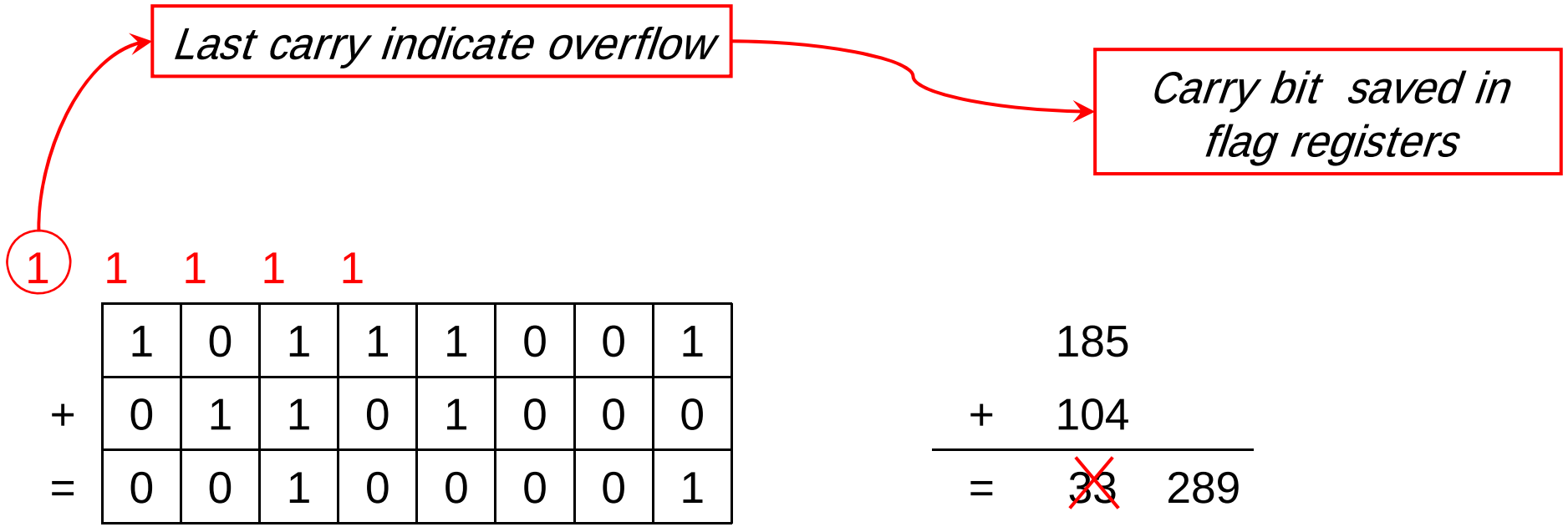
In computers and digital systems, the number of bits used to store binary numbers is limited

==> operation overflow occurs when the result is too large to be represented by the limited number of bits

Solution:

In unsigned arithmetic the Carry bit generated at the MSB is used for overflow

2. Binary Representation



8-bit binary addition

Decimal addition

8-bit unsigned binary addition with overflow

2. Binary Representation

Signed binary numbers

- Sign-magnitude representation
- 1's complement representation
- 2's complement representation
- Excess 2^{n-1}-1

Example: 4-bit

- $(5)_{10} = (0101)_{sm} = (0101)_{1'C} = (0101)_{2'C} = (1100)_{E7}$
- $(-5)_{10} = (1101)_{sm} = (1010)_{1'C} = (1101)_{2'C} = (0010)_{E7}=$

2. Binary Representation

2's complement is the preferred representation for arithmetic operations

- We have:

$$Z + \bar{Z} = 1...1$$

$$\Rightarrow Z + \bar{Z} + 1 = 0...0 = 0 \text{ (ignore MSB carry bit)}$$

$$\Rightarrow -Z = \bar{Z} + 1$$

- 2's complement numbers can be added by ordinary binary addition ignoring MSB carry bit

2. Binary Representation

Signed binary addition using 2's complement representation

+3	0011	−2	1110
+ +4	+ 0100	+ −6	+ 1010
+7	0111	−8	11000
+6	0110	+4	0100
+ −3	+ 1101	+ −7	+ 1001
+3	10011	−3	1101

Examples of 4-bit signed binary addition

Source: Digital Design Principles and practices, by J. F. Wakerly

2. Binary Representation

How to detect overflow in signed arithmetic addition?

- Overflow can occur only with the addition of two numbers of same sign
- If overflow occur, the sign of the result is different from the sign of the numbers. Also overflow can be detected if the carry in and the carry out of the LSB bit are different

−3	1101		+5	0101	
+ −6	+ 1010		+ +6	+ 0110	
−9	10111	= +7	+11	1011	= −5

Source: Digital Design Principles and practices, by J. F. Wakerly

3. Combinational Logic Circuits

Combinational vs. Sequential circuits

- Combinational circuits: output depends only on its present input (memoryless)
- Sequential circuits: output depends on its present and previous inputs (depends on its state)

3. Combinational Logic Circuits

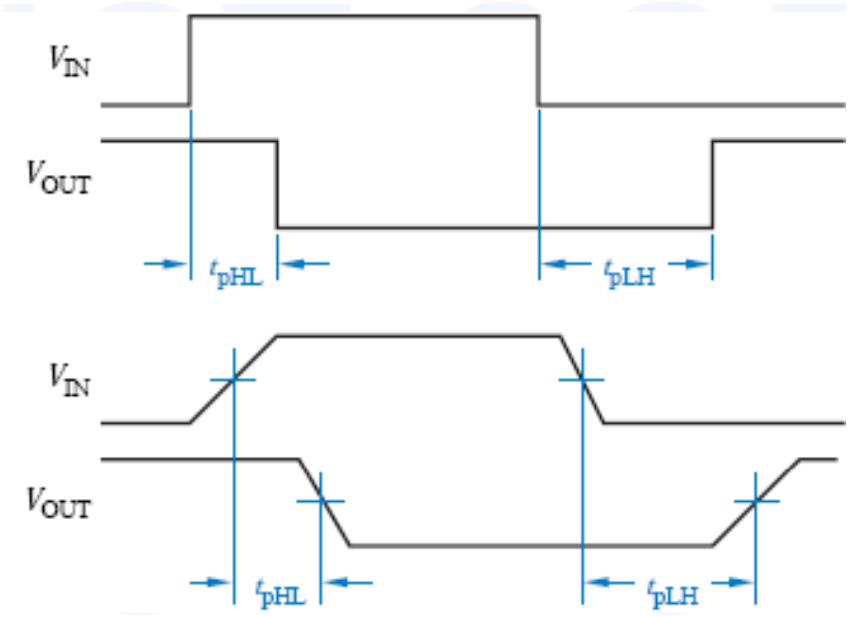
Each combinational circuit can be described as:

- Equation : sum of products, product of sums, factorial form
- Truth table
- Logic diagram
- Binary Decision Diagram
- Timing diagram
- ...etc

3. Combinational Logic Circuits

Timing diagram of logic circuit

- It shows how the circuit respond to a pattern of input signals
- Trans., delay
- tr: rising delay
- tf: falling delay
- tpHL:
- tpLH:



3. Combinational Logic Circuits

Timing diagram of logic circuit

- Why propagation delays are important in digital design?

- Glitch example: s = A and not(A)

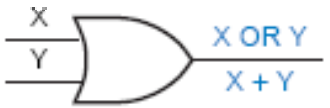
3. Combinational Logic Circuits

Basic logic gates

Circuits implementing basic functions are called gates. Any digital function can be realized with the following gates:

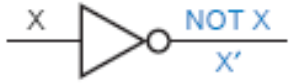
X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

(b)



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

(c)

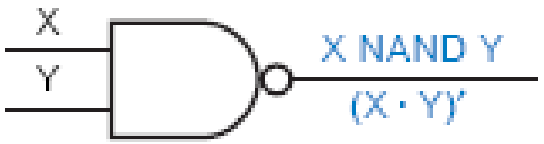


X	NOT X
0	1
1	0

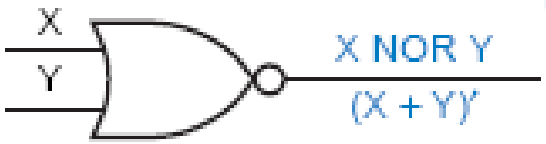
Source: Digital Design Principles and practices, by J. F. Wakerly

3. Combinational Logic Circuits

Basic logic gates



(b)



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

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

- NAND() ⇔ NOT(AND ()) or
- AND() ⇔ NOT (NAND ()) ?

Source: Digital Design Principles and practices, by J. F. Wakerly

3. Combinational Logic Circuits

Basic logic gates

- 74LS00: Quad 2-inputs NAND gate
- 74LS02: Quad 2-inputs NOR gate
- 74LS04: Hex inverter
- 74LS08: Quad 2 inputs AND gate
- 74LS32: Quad 2 inputs OR gate
- 74LS86: Quad 2 inputs XOR gate

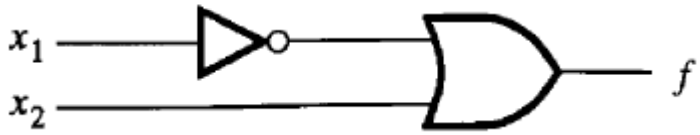
Source: Digital Design Principles and practices, by J. F. Wakerly

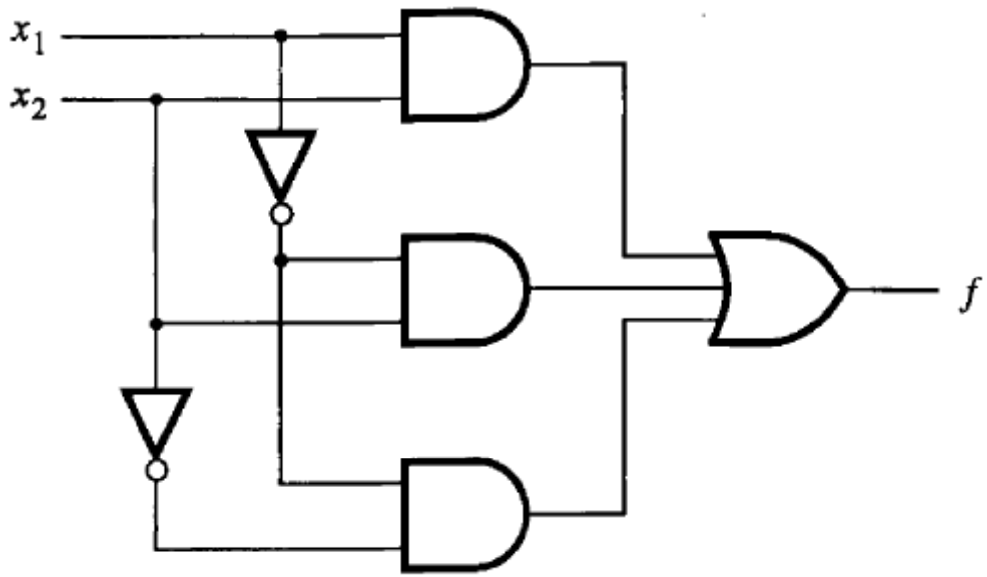
3. Combinational Logic Circuits

Circuit synthesis and optimization using Karnaugh-map

$$f(x_1, x_2) = x_1x_2 + \bar{x}_1\bar{x}_2 + \bar{x}_1x_2$$

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



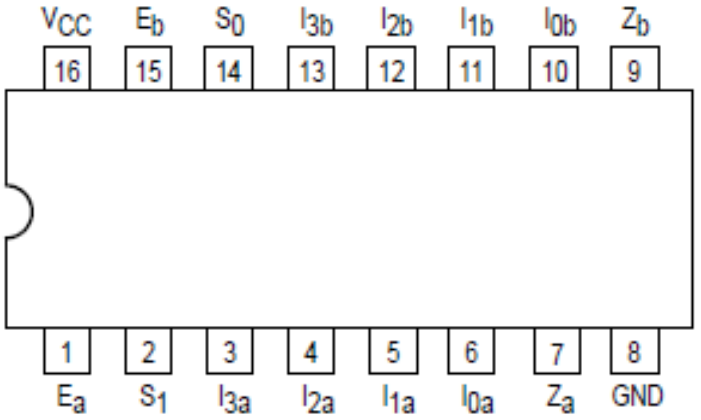


3. Combinational Logic Circuits

Multiplexer/Demultiplexer

SELECT INPUTS		INPUTS (a or b)					OUTPUT
S_0	S_1	$\overline{E}$	I_0	I_1	I_2	I_3	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care



$$Z_a = \overline{E}_a \cdot (I_{0a} \cdot \overline{S}_1 \cdot \overline{S}_0 + I_{1a} \cdot \overline{S}_1 \cdot S_0 + I_{2a} \cdot S_1 \cdot \overline{S}_0 + I_{3a} \cdot S_1 \cdot S_0)$$

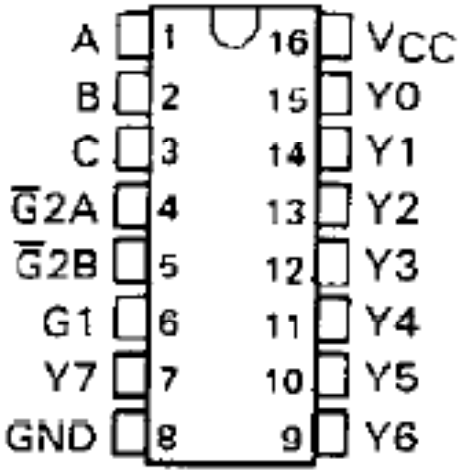
$$Z_b = \overline{E}_b \cdot (I_{0b} \cdot \overline{S}_1 \cdot \overline{S}_0 + I_{1b} \cdot \overline{S}_1 \cdot S_0 + I_{2b} \cdot S_1 \cdot \overline{S}_0 + I_{3b} \cdot S_1 \cdot S_0)$$

Connection diagram, truth table and equations of the dual 4-input multiplexer (74LS153)

3. Combinational Logic Circuits

Decoder/Encoder

Inputs					Outputs							
Enable		Select										
G1	G2 (Note 1)	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L



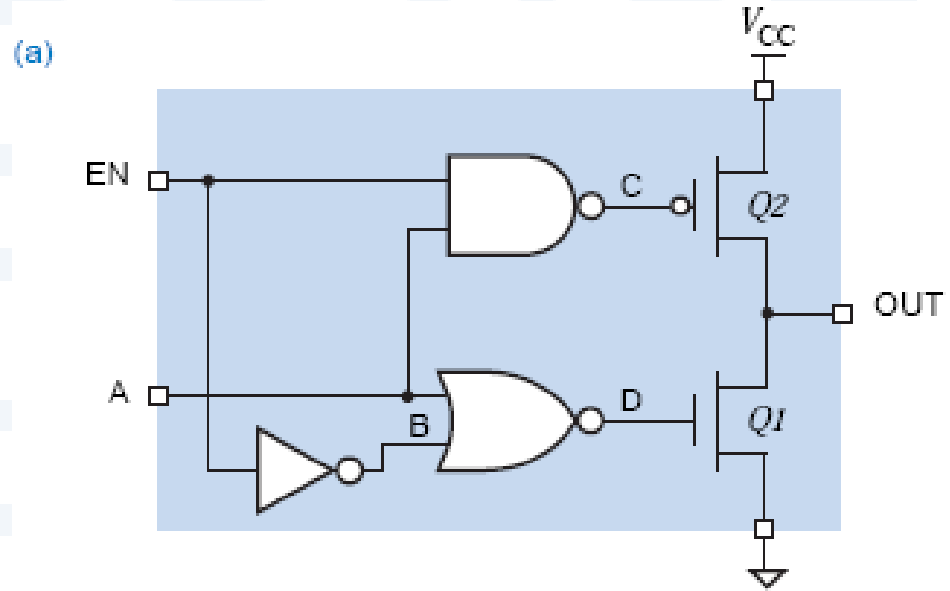
Pin diagram and truth table of the 3-to-8 decoder (74LS138).

Note 1:

$G2 = \overline{G2A} + \overline{G2B}$

3. Combinational Logic Circuits

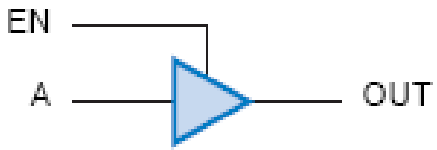
Tristate buffer



(b)

EN	A	B	C	D	Q1	Q2	OUT
L	L	H	H	L	off	off	Hi-Z
L	H	H	H	L	off	off	Hi-Z
H	L	L	H	H	on	off	L
H	H	L	L	L	off	on	H

(c)



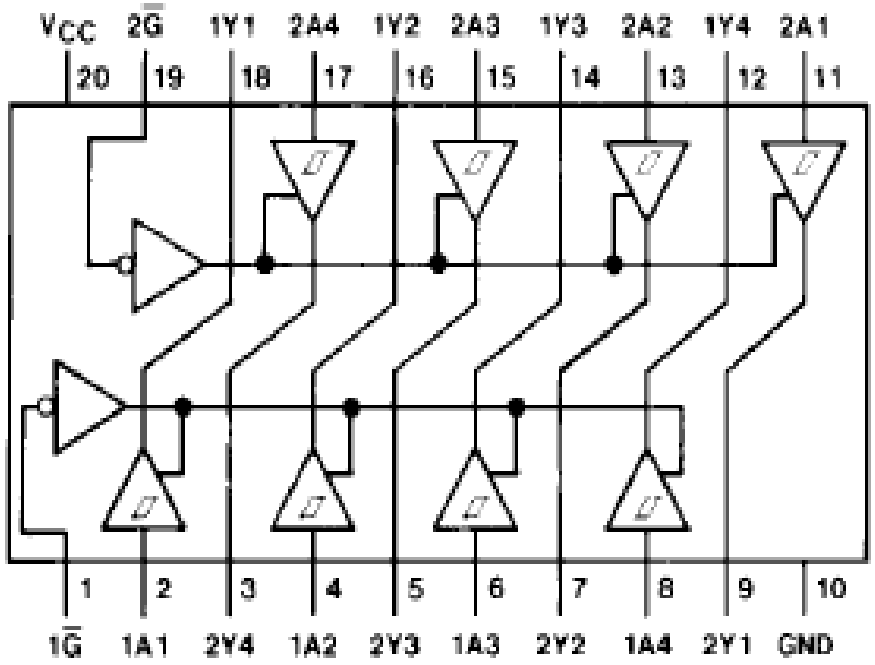
Circuit diagram, function table and logic symbol of the CMOS 3-state buffer.

Source: Digital Design Principles and practices, by J. F. Wakerly

3. Combinational Logic Circuits

Tristate buffer

Connection Diagram



Function Table

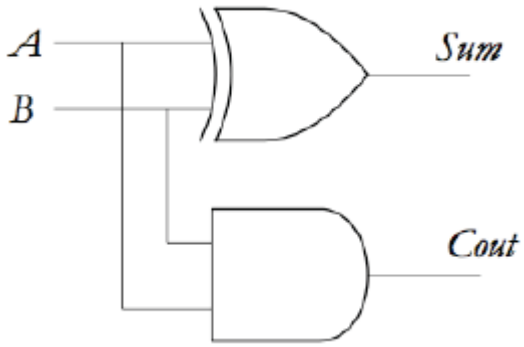
Inputs		Output
$\overline{G}$	A	Y
L	L	L
L	H	H
H	X	Z

L = LOW Logic Level
H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
Z = High Impedance

Pin diagram and function table of the 3-state octal buffer (74LS244).

Binary addition: Half Adder (HA)

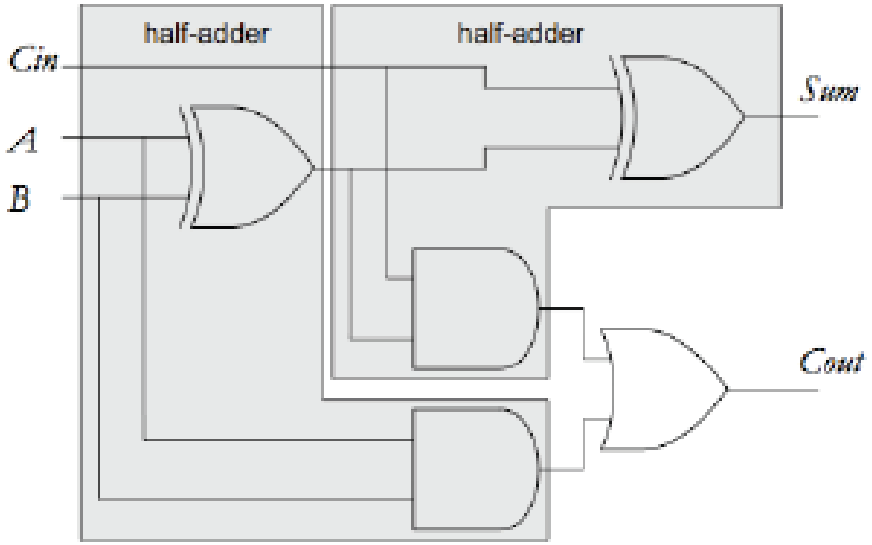
A + B	Sum	Cout
0 + 0	0	0
0 + 1	1	0
1 + 0	1	0
1 + 1	0	1



3. Combinational Logic Circuits

Binary addition: Full Adder (FA)

- Sum = A xor B xor Cin
- Cout = AB+CinB+CinA
- = AB+Cin(AxorB)

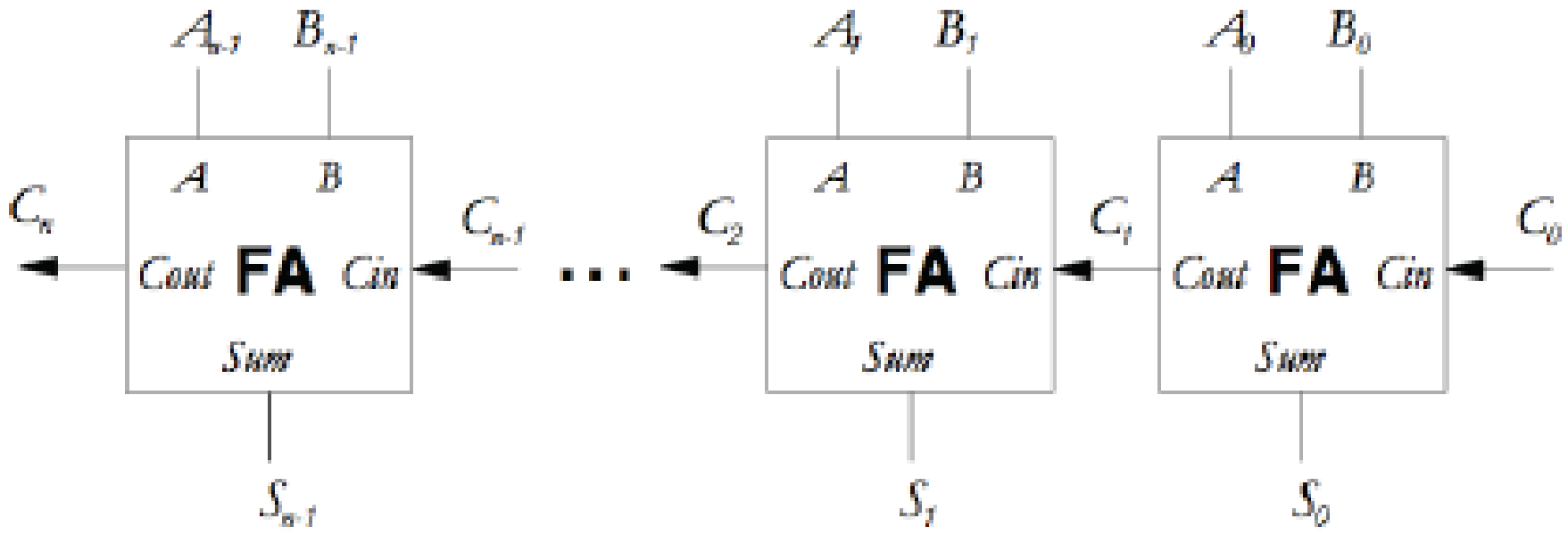


- Karnaugh-map for Cout

Cin\AB	00	01	11	10
0	0	0	1	0
1	0	1	1	1

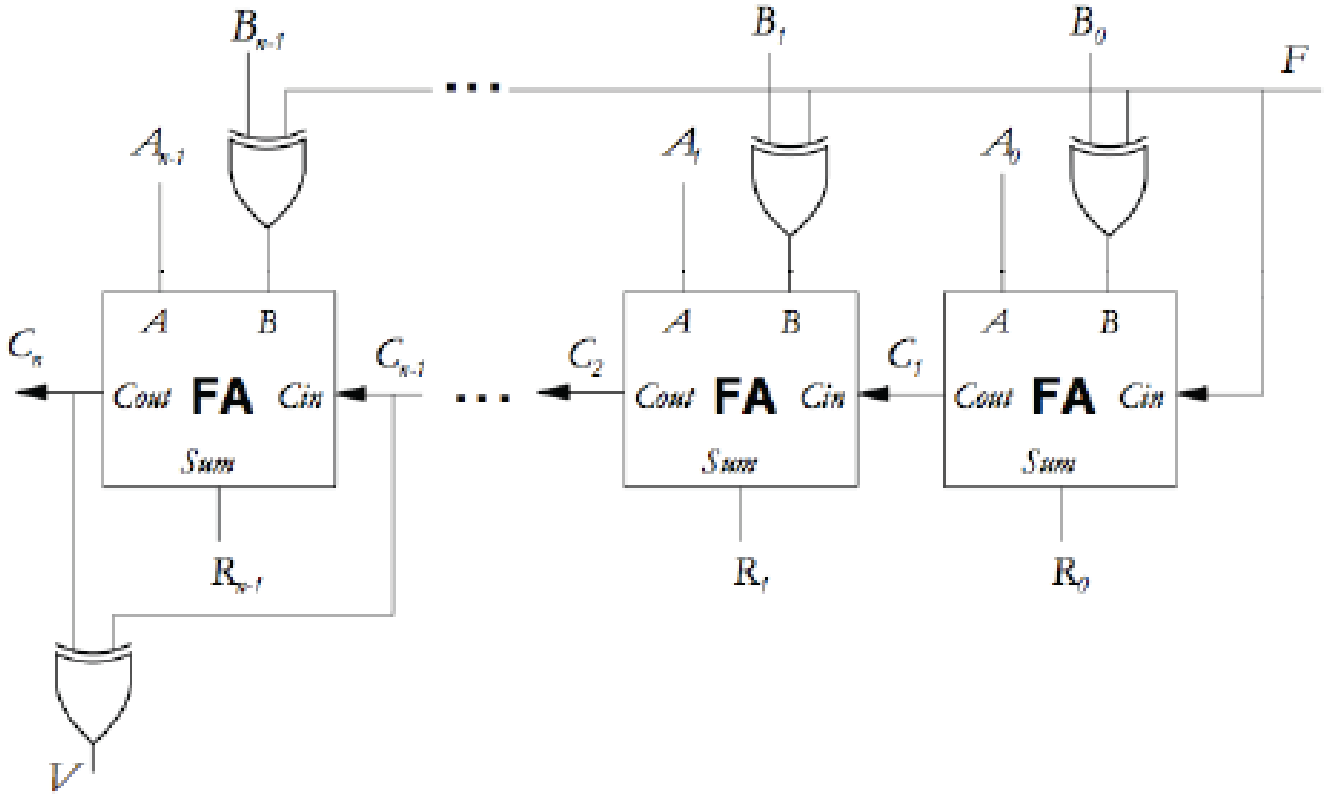
3. Combinational Logic Circuits

Binary addition: n-bit adder (Ripple-carry Adder)



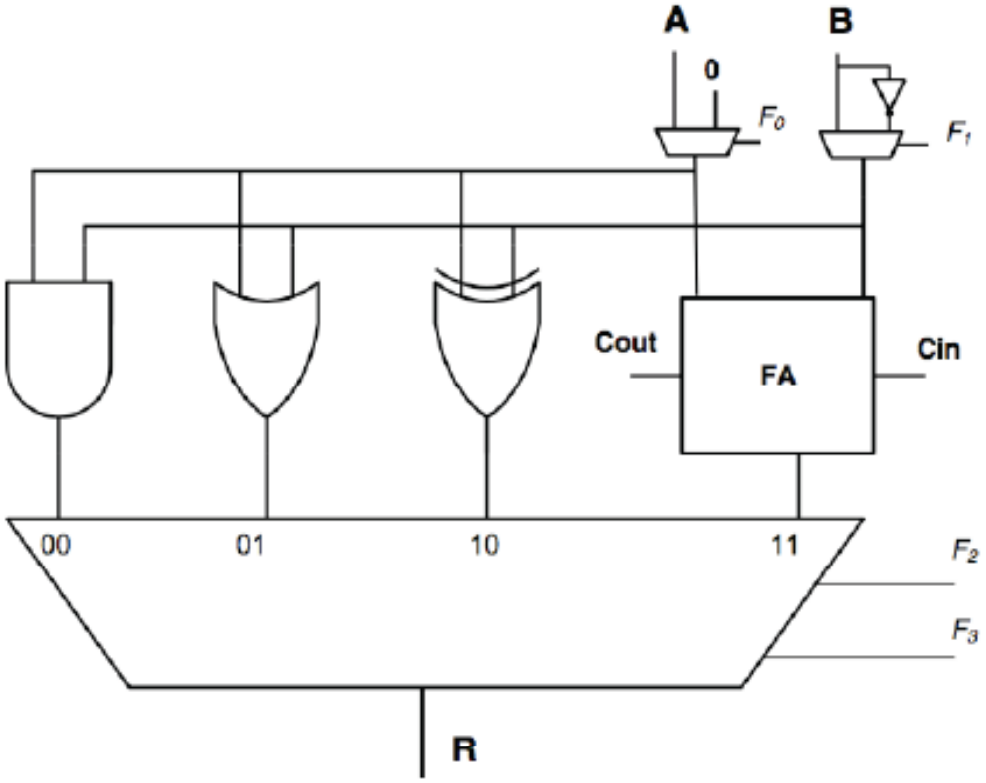
3. Combinational Logic Circuits

Binary addition: n-bit adder/subtractor



3. Combinational Logic Circuits

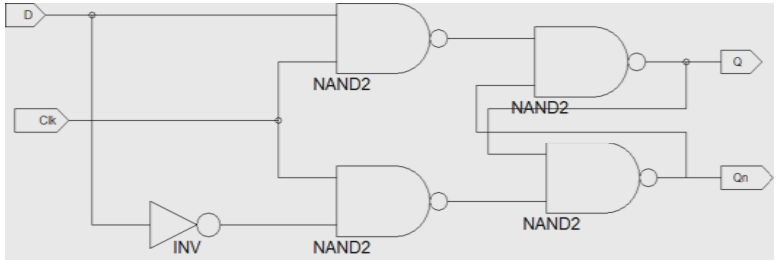
Arithmetic and logic unit (ALU)

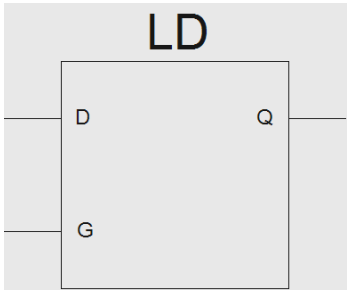


ALU Function	F_3	F_2	F_1	F_0
A AND B	0	0	0	0
A OR B	0	1	0	0
A XOR B	1	0	0	0
NOT B	1	0	1	1
A + B	1	1	0	0
A - B	1	1	1	0
-B	1	1	1	1

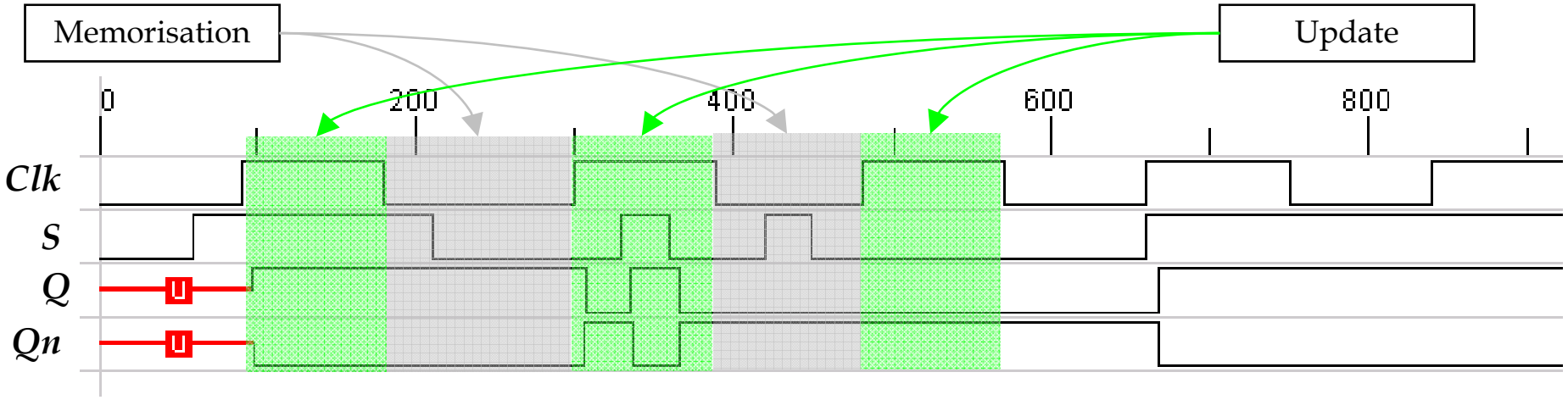
4. Sequential Circuits

Latch



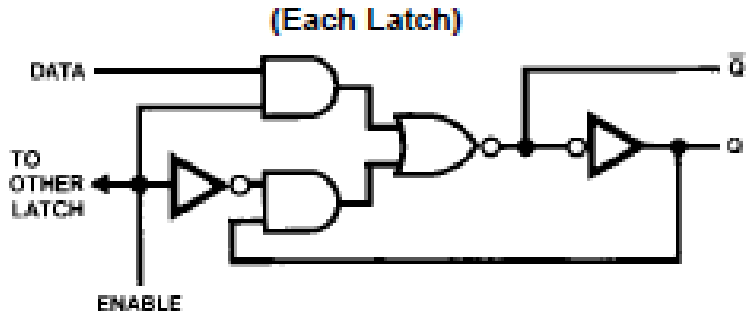


D	Clk	Q+	Qn+
X	0	Q	Qn
0	1	0	1
1	1	1	0



4. Sequential Circuits

Logic Diagram

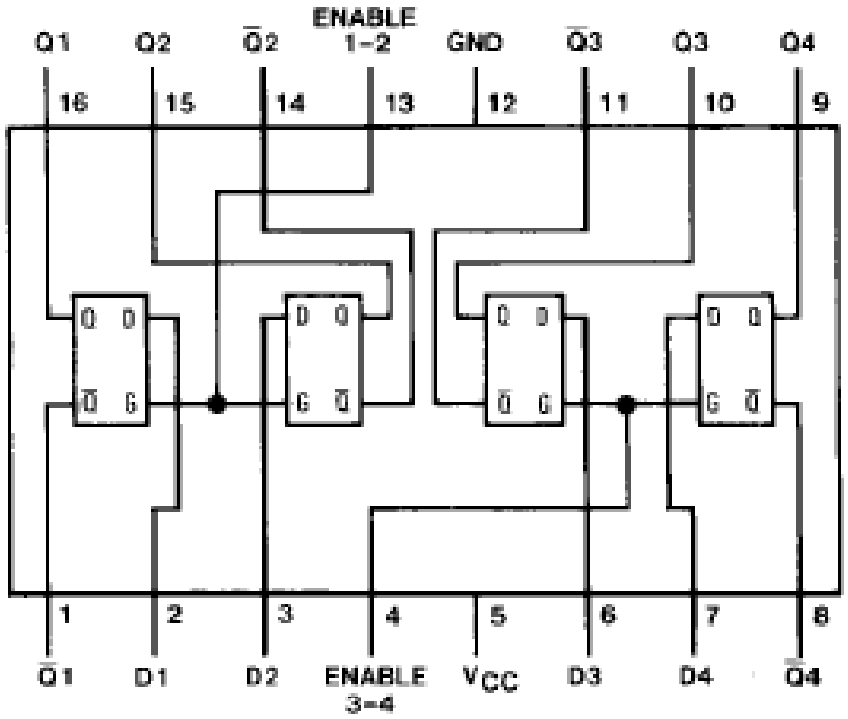


Function Table (Each Latch)

Inputs		Outputs	
D	Enable	Q	$\overline{Q}$
L	H	L	H
H	H	H	L
X	L	Q_0	$\overline{Q}_0$

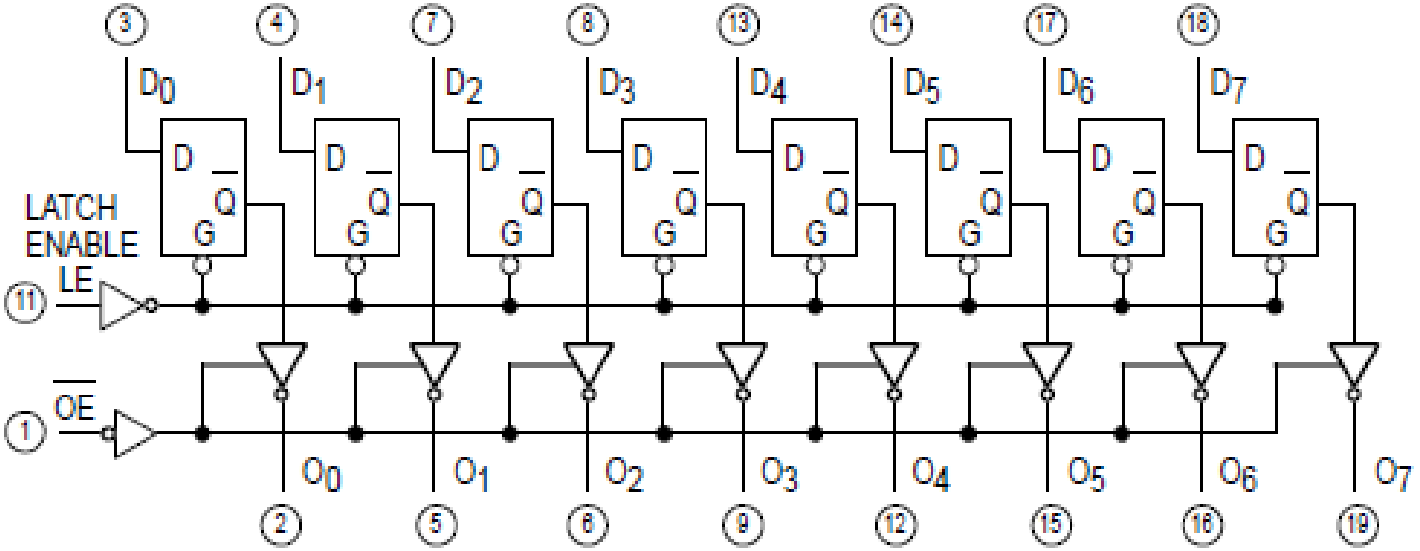
H = HIGH Level
L = LOW Level
X = Don't Care
Q_0 = The Level of Q Before the HIGH-to-LOW Transition of ENABLE

Connection Diagram



Logic and pin diagram and function table of the quad latch (74LS75).

4. Sequential Circuits



LS373

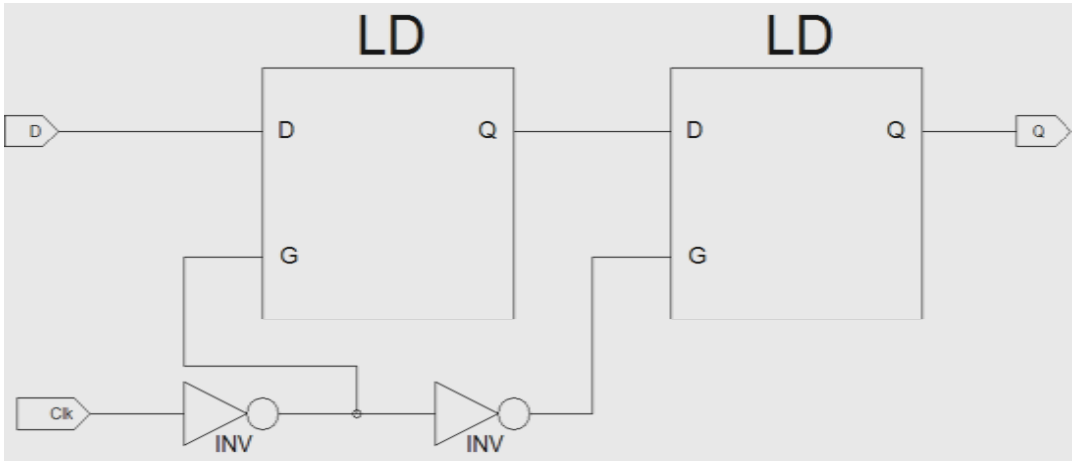
D_n	LE	$\overline{OE}$	O_n
H	H	L	H
L	H	L	L
X	L	L	Q_0
X	X	H	Z*

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance

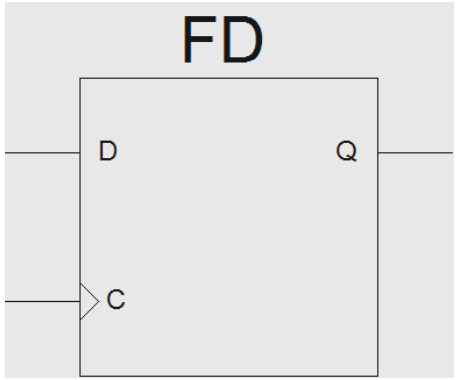
Logic diagram and function table of the octal transparent latch with 3-state outputs (74LS373).

4. Sequential Circuits

D-type Flip-Flop (DFF)



Master-slave positive-edge triggered DFF



Symbol

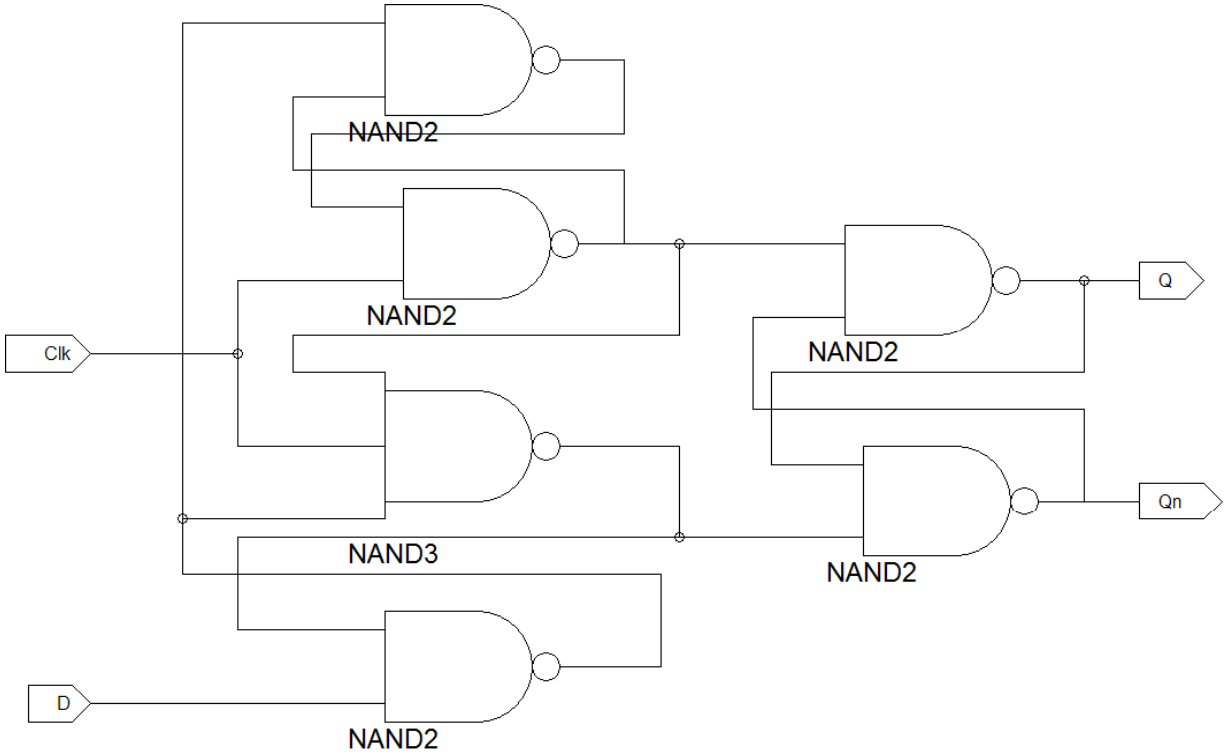
D	Clk	Q+
X	0	Q
X	1	Q
0	↑	0
1	↑	1

Positive edge

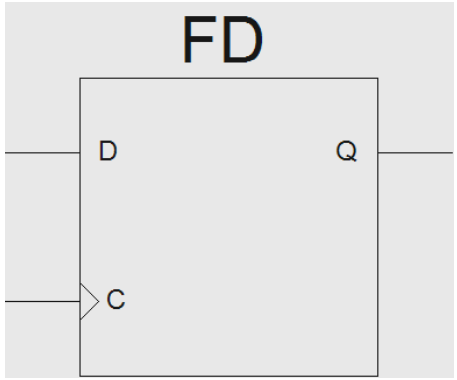
Function table

4. Sequential Circuits

D-type Flip-Flop (DFF)



Positive-edge triggered DFF designed using NAND gates

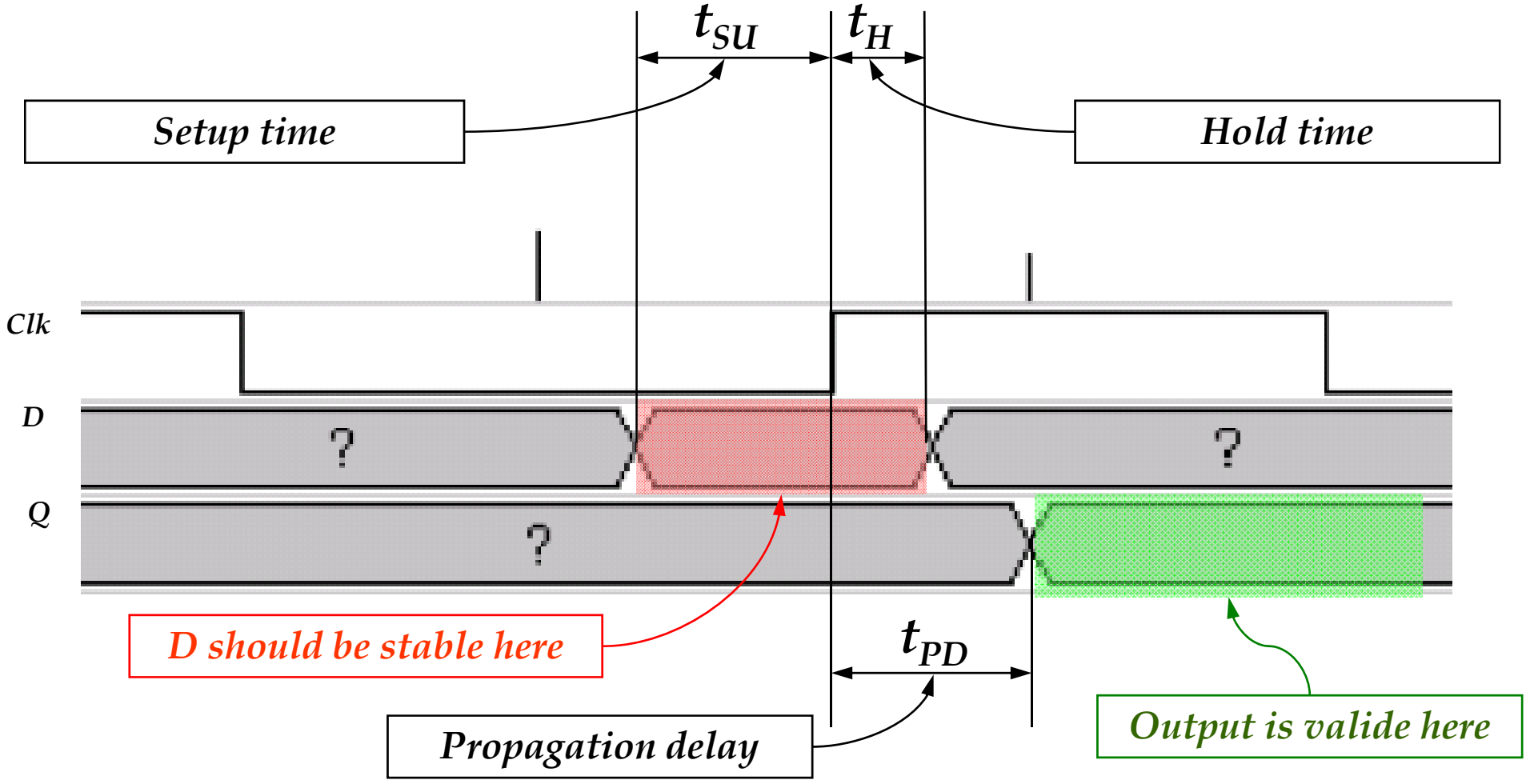


Symbol

D	Clk	Q+
X	0	Q
X	1	Q
0	↑	0
1	↑	1

Function table

4. Sequential Circuits



Setup time, hold time, and propagation delay for positive edge triggered DFF

4. Sequential Circuits

D-type Flip-Flop (DFF)

- Equation: Q = D
- Table
- Timing diagram?

D	CLK	Q_{n+1}	$\overline{Q}_{n+1}$	
X	0	Q_n	$\overline{Q}_n$	Latch
X	1	Q_n	$\overline{Q}_n$	Latch
0	↑	0	1	Reset
1	↑	1	0	Set

DFF with enable signal

- Q = D when En=1
- Table?
- Timing diagram?

4. Sequential Circuits

DFF with enable and asynchronous reset signals

- Equation?
- Table?
- Timing diagram?

DFF with enable and synchronous set signals

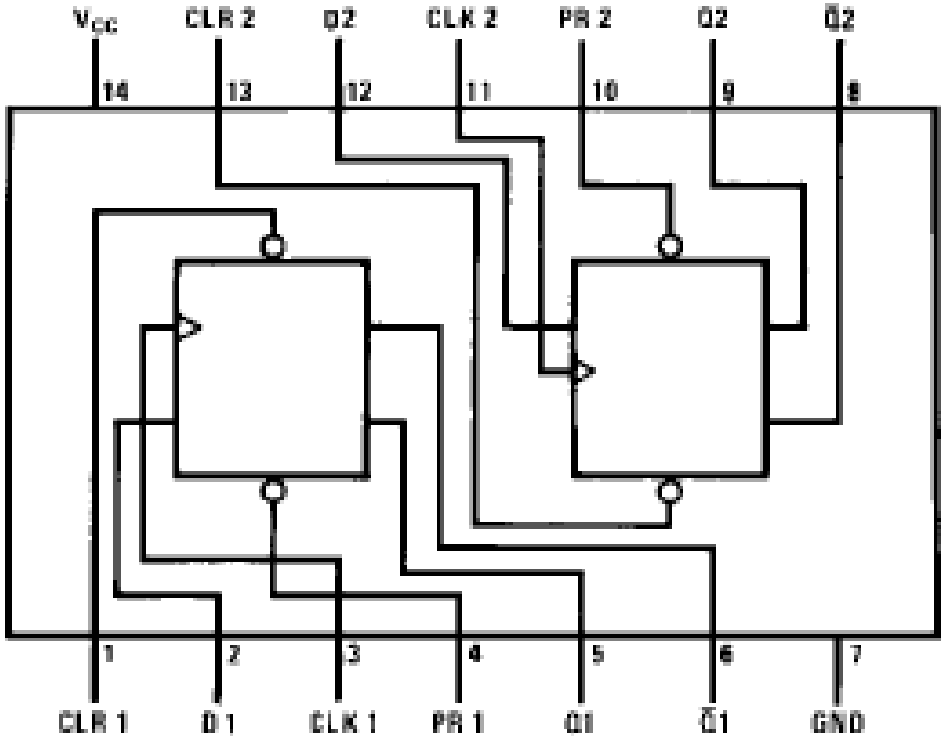
- Equation?
- Table?
- Timing diagram?

4. Sequential Circuits

Inputs				Outputs	
PR	CLR	CLK	D	Q	$\overline{Q}$
L	H	X	X	H	L
H	L	X	X	L	H
L	L	X	X	H (Note 1)	H (Note 1)
H	H	↑	H	H	L
H	H	↑	L	L	H
H	H	L	X	Q_0	$\overline{Q}_0$

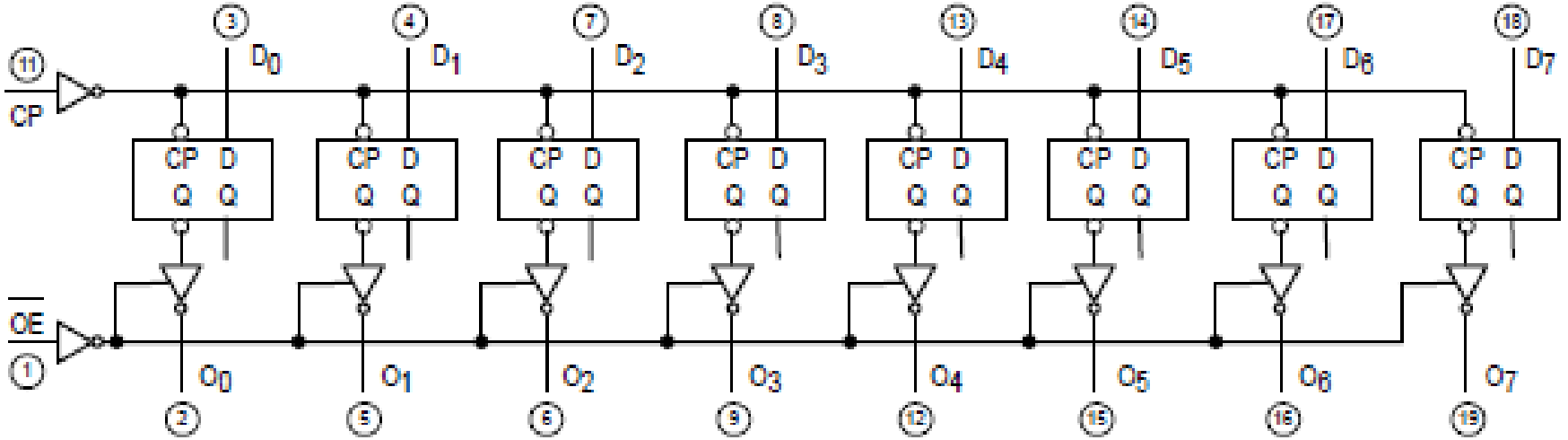
H = HIGH Logic Level
X = Either LOW or HIGH Logic Level
L = LOW Logic Level
↑ = Positive-going Transition
Q_0 = The output logic level of Q before the indicated input conditions were established.

Note 1: This configuration is nonstable; that is, it will not persist when either the preset and/or clear inputs return to their inactive (HIGH) level.



Pin diagram and function table of the Dual Positive-Edge-Triggered DFF with Preset, Clear and Complementary Outputs (74LS74).

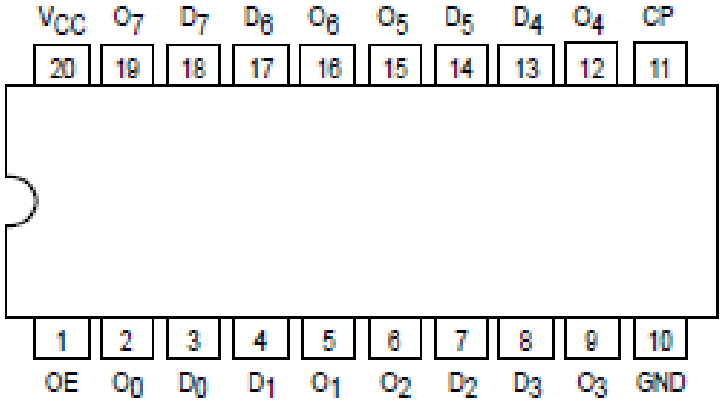
4. Sequential Circuits



LS374

D_n	LE	OE	O_n
H	⎍	L	H
L	⎍	L	L
X	X	H	Z*

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = High Impedance



Logic and pin diagram and function table of the octal transparent DFF with 3-state outputs (74LS374).

4. Sequential Circuits

N-bit Register

- Inputs : D(n-1:0)
- Output: Q[n-1:0]
- Symbol?
- Timing Diagram?

$$Equations:$$

$$Q_0(t+) = D_0(t)$$

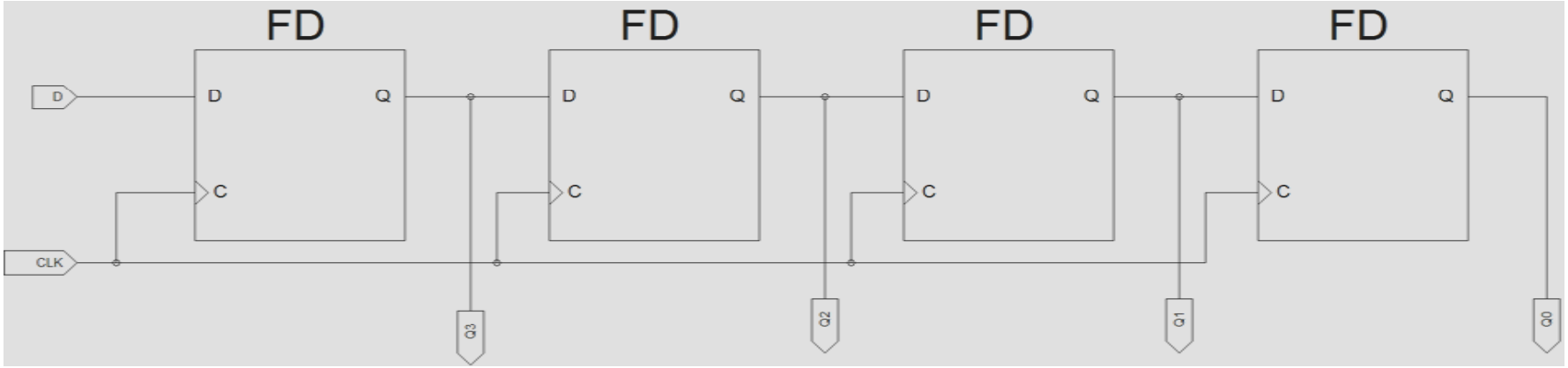
$$Q_1(t+) = D_1(t)$$

$$Q_2(t+) = D_2(t)$$

$$Q_3(t+) = D_3(t)$$

4. Sequential Circuits

Shift Register



$$Equations:$$

$$Q_0(t+1) = Q_1(t)$$

$$Q_1(t+1) = Q_2(t)$$

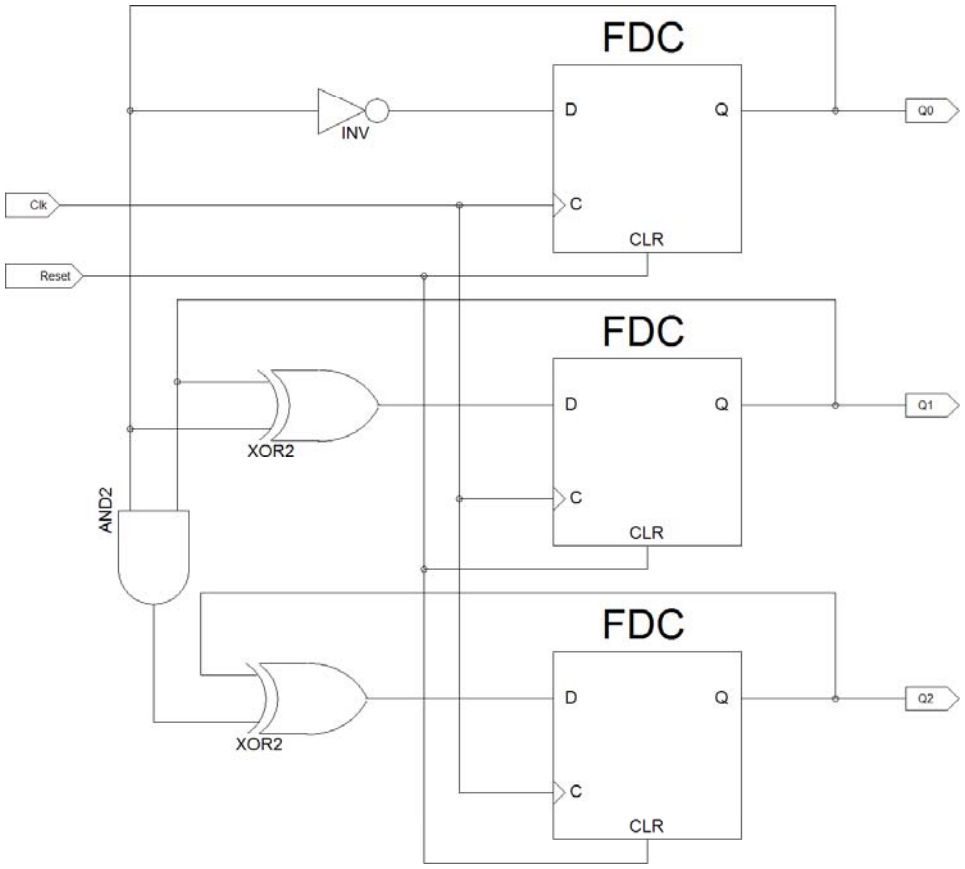
$$Q_2(t+1) = Q_3(t)$$

$$Q_3(t+1) = D(t)$$

CLK	D	Q3	Q2	Q1	Q0
0	1	U	U	U	U
1	0	1	U	U	U
2	0	0	1	U	U
3	1	0	0	1	U
4	0	1	0	0	1
5	0	0	1	0	0

4. Sequential Circuits

Counter



Q2	Q1	Q0	Q2+	Q1+	Q0+
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

Equations :

$$Q_0(t+1) = \overline{Q_0(t)}$$

$$Q_1(t+1) = Q_0(t) \oplus Q_1(t)$$

$$Q_2(t+1) = Q_0(t).Q_1(t) \oplus Q_2(t)$$

5. Logic Families

TTL (Transistor-Transistor Logic) family

- Based on Bipolar transistor
- Replaced by CMOS family in 1990

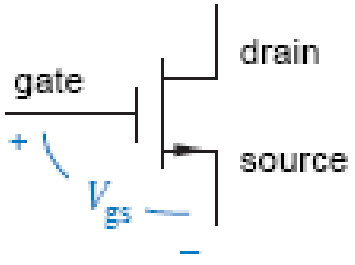
CMOS

- Based on MOSFET (Metal-Oxide Semiconductor Field Effect Transistor)

5. Logic Families

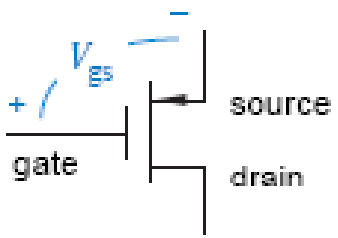
N-Channel MOS Transistor

- If V_{GS}=0, NMOS is OFF
- If V_{GS}=VDD, NMOS is ON



P-Channel MOS Transistor

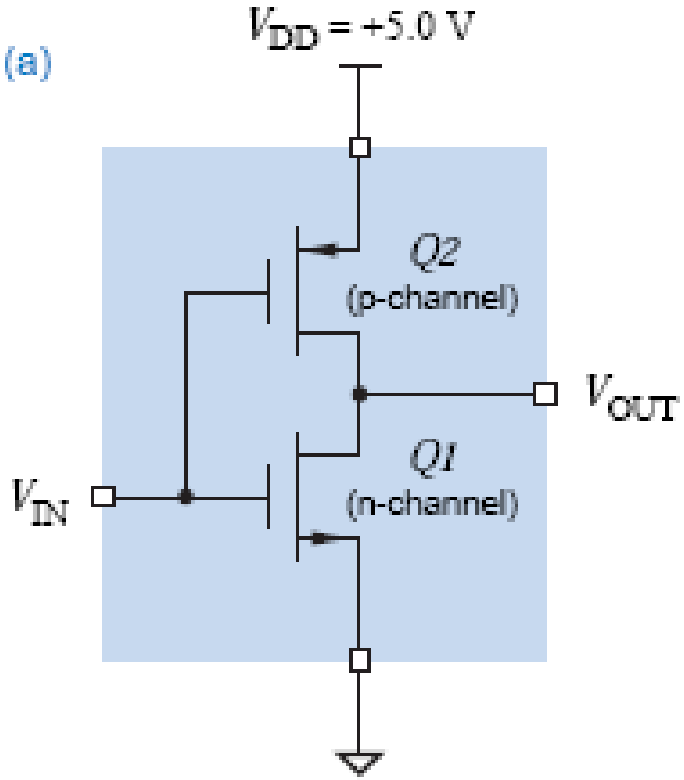
- If V_{SG}=0, PMOS is OFF
- If V_{SG}=VDD, PMOS is ON



5. Logic Families

CMOS Inverter

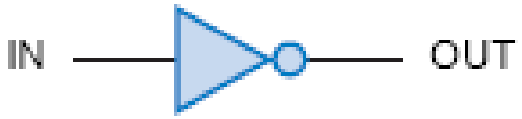
(a)



(b)

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

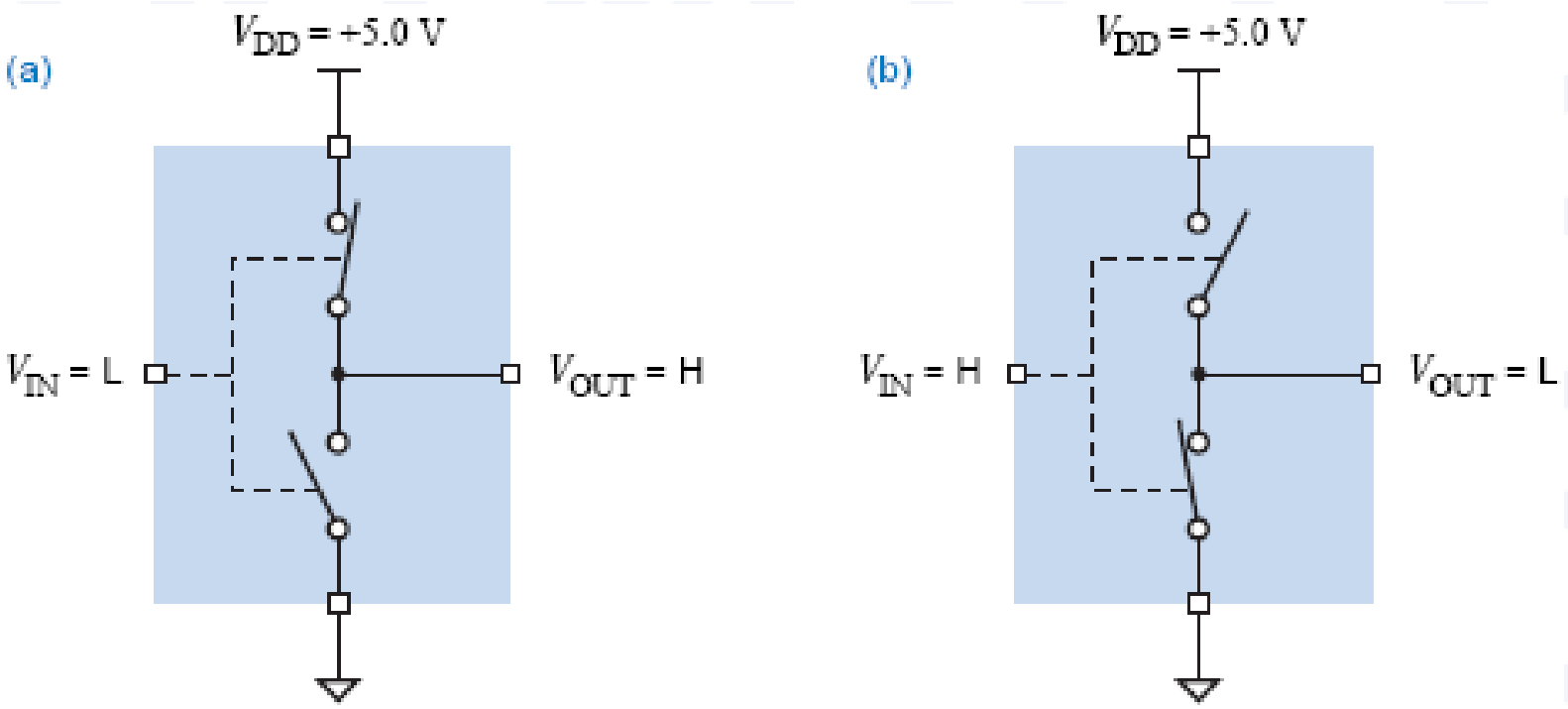
(c)



Source: Digital Design Principles and practices, by J. F. Wakerly

5. Logic Families

CMOS Inverter

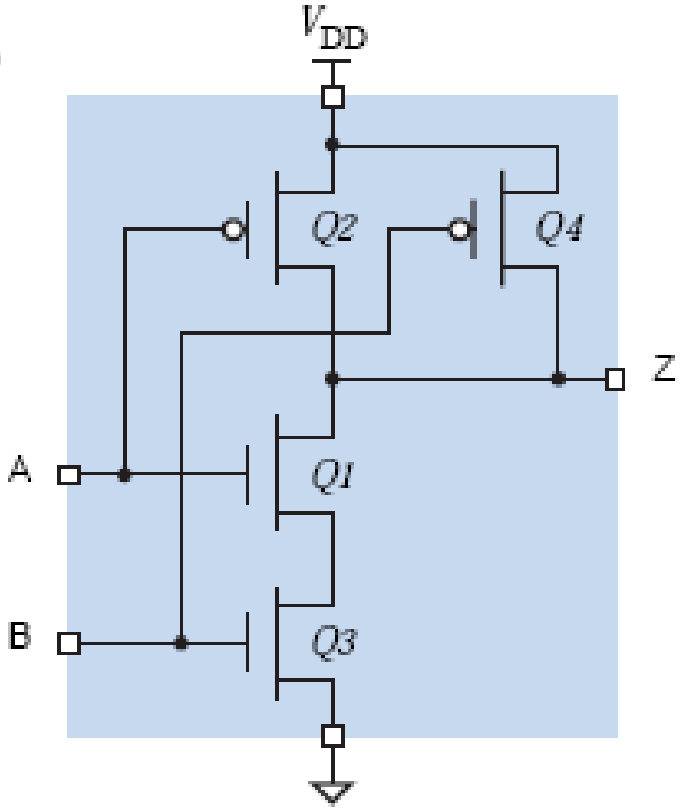


Source: Digital Design Principles and practices, by J. F. Wakerly

5. Logic Families

CMOS 2-input NAND gate

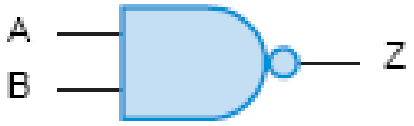
(a)



(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)

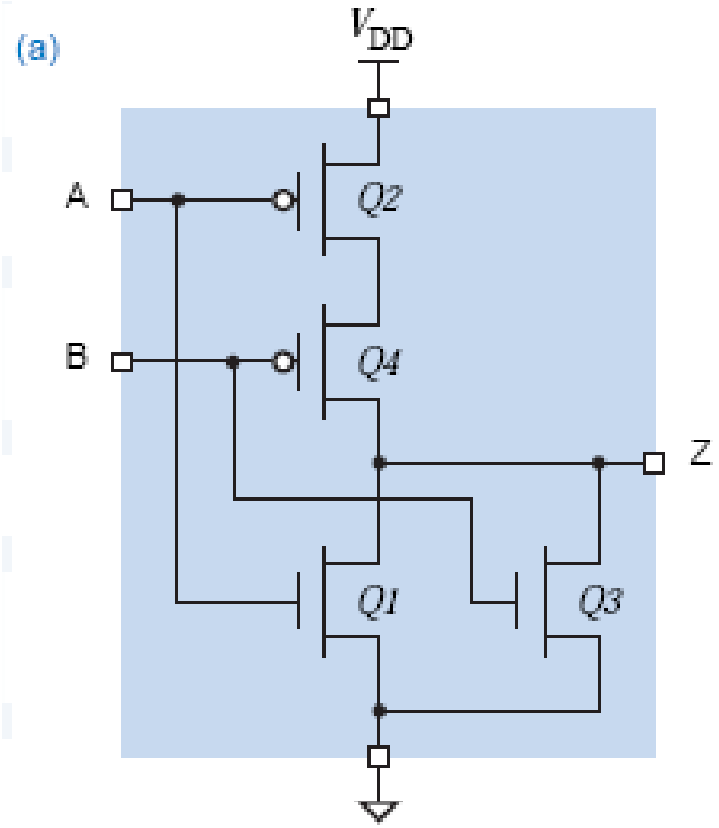


Source: Digital Design Principles and practices, by J. F. Wakerly

5. Logic Families

CMOS 2-input NOR gate

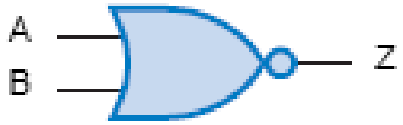
(a)



(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)



Source: Digital Design Principles and practices, by J. F. Wakerly

5. Logic Families

CMOS Gates

- What about AND and OR gates?

5. Logic Families

Other CMOS Gates

- Y = not(AB+C) ?

- Y = A(B+C) ?

Summary

- Digital systems
  - Analog vs digital systems
  - Hardware vs. software design
- Binary representations
  - Two’s complement representation
- Combinational logic circuits
- Sequential circuits
- Logic families
  - NMOS and PMOS transistors
  - CMOS Family