Design of Digital Systems II

Digital Circuits

Moslem Amiri, Václav Přenosil

Embedded Systems Laboratory
Faculty of Informatics, Masaryk University
Brno, Czech Republic

amiri@mail.muni.cz
prenosil@fi.muni.cz

October, 2012

Introduction

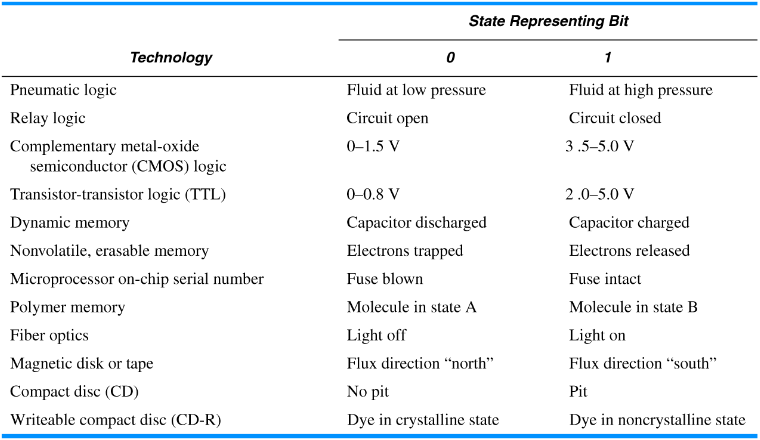
- Voltages, currents, and other physical quantities in real circuits take on values that are infinitely variable
    - Stability and accuracy in physical quantities are difficult to obtain, hence they cannot be used to represent real numbers
    - Also, many mathematical and logical operations can be difficult or impossible to perform with analog quantities

Logic Signals and Gates

- **Digital logic** hides pitfalls of analog world by mapping infinite set of real values for a physical quantity into two subsets corresponding to two **logic values**—0 and 1
  - A logic value, 0 or 1, is called a **binary digit**, or **bit**

Logic Signals and Gates

Table 1: Physical states representing bits in different logic and memory techs.



	State Representing Bit	
Technology	***0***	***1***
Pneumatic logic	Fluid at low pressure	Fluid at high pressure
Relay logic	Circuit open	Circuit closed
Complementary metal-oxide semiconductor (CMOS) logic	0–1.5 V	3 .5–5.0 V
Transistor-transistor logic (TTL)	0–0.8 V	2 .0–5.0 V
Dynamic memory	Capacitor discharged	Capacitor charged
Nonvolatile, erasable memory	Electrons trapped	Electrons released
Microprocessor on-chip serial number	Fuse blown	Fuse intact
Polymer memory	Molecule in state A	Molecule in state B
Fiber optics	Light off	Light on
Magnetic disk or tape	Flux direction “north”	Flux direction “south”
Compact disc (CD)	No pit	Pit
Writeable compact disc (CD-R)	Dye in crystalline state	Dye in noncrystalline state

- In Tab. 1, with most phenomena, there is an undefined region between 0 and 1 states, so that 0 and 1 states can be unambiguously defined

Logic Signals and Gates

- When discussing electronic logic circuits such as CMOS and TTL
    - **LOW**: A signal in the range of lower voltages, which is interpreted as a logic 0
    - **HIGH**: A signal in the range of higher voltages, which is interpreted as a logic 1
- **Positive logic**
    - Assignment of 0 to LOW and 1 to HIGH
- **Negative logic**
    - Assignment of 1 to LOW and 0 to HIGH
- A wide range of physical values represent the same binary value
    - Hence, digital logic is immune to component and power-supply variations and noise
    - **Buffer** circuits can also be used to regenerate or amplify weak values into strong ones
        - E.g., a buffer for CMOS logic converts any HIGH (LOW) input voltage into an output very close to 5.0 V (0.0 V)

Logic Signals and Gates

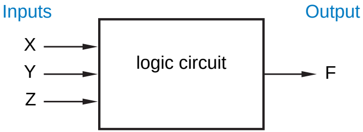


Figure 1: "Black-box" representation of a 3-input, 1-output logic circuit.

- **Black-box** representation of a logic circuit does not describe how the circuit responds to input signals
  - It takes a lot of information to describe electrical behavior of a circuit
  - But, a circuit's logical operation can be described with a table

Logic Signals and Gates

- **Combinational circuit**
  - A logic circuit whose outputs depend only on its current inputs
  - Its operation is fully described by a **truth table**
    - Truth table lists all combinations of input values and the output value(s) produced by each one

Table 2: Truth table for a combinational logic circuit with three inputs X, Y, and Z and a single output F.

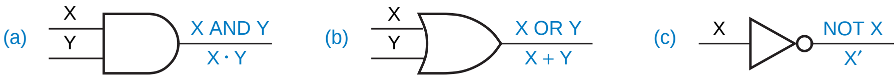
X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Logic Signals and Gates

- **Sequential circuit**
  - A circuit with memory, whose outputs depend on current input *and* sequence of past inputs
  - Its behavior may be described by a **state table**
    - State table specifies its output and next state as functions of its current state and input

Logic Signals and Gates

- Three basic logic functions, AND, OR, and NOT can be used to build any combinational digital logic circuit



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

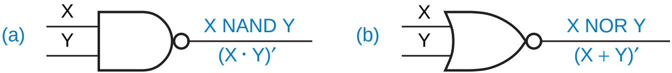
X	NOT X
0	1
1	0

Figure 2: Basic logic elements: (a) AND; (b) OR; (c) NOT (inverter).

- The circle on inverter symbol's output is called an **inversion bubble**
  - Used to denote "inverting" behavior

Logic Signals and Gates

- Two more logic functions are obtained by combining inversion with an AND or OR function in a single gate



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

Figure 3: Inverting gates: (a) NAND; (b) NOR.

- Symbols and truth tables for AND, OR, NAND, and NOR may be extended to gates with any number of inputs

Logic Signals and Gates

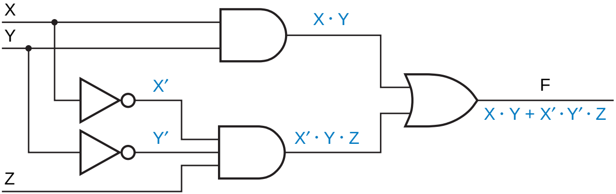


Figure 4: Logic circuit with the truth table of Tab. 2.

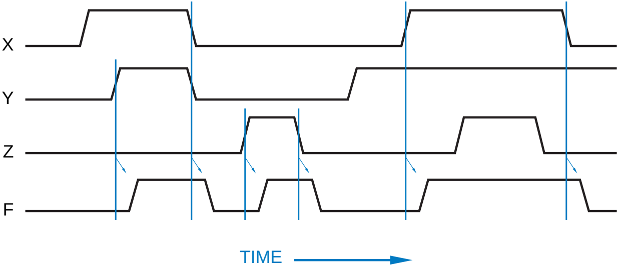


Figure 5: Timing diagram for the logic circuit shown in Fig. 4.

Logic Families

- Electronic logic circuit development
  - In 1930s, the first electronically controlled logic circuits, developed at Bell Laboratories, were based on relays
  - In mid-1940s, the first electronic digital computer, Eniac, used logic circuits based on vacuum tubes
  - In late 1950s, **semiconductor diode** and **bipolar junction transistor** were invented
    - Allowed development of smaller, faster, and more capable computers
  - In 1960s, **integrated circuit (IC)** was invented
    - Allowed multiple diodes, transistors, and other components to be fabricated on a single chip
  - In 1960s, the first IC logic families were also introduced
    - A **logic family** is a collection of different IC chips that have similar input, output, and internal circuit characteristics, but that perform different logic functions
    - Chips from same family can be interconnected
    - Chips from different families may not be compatible

Logic Families

- **Transistor-transistor logic (TTL)**
  - The most successful **bipolar logic family**
    - Based on bipolar junction transistors
  - First introduced in 1960s
  - TTL evolved into a family of logic families that were compatible with each other but differed in speed, power consumption, and cost
- **Metal-oxide semiconductor field-effect transistor (MOSFET)** or simply **MOS transistor**
  - Its principles were introduced ten years *before* bipolar junction transistor
  - Difficult to fabricate in early days until 1960s
  - Even in 1960s, MOS circuits were slower than bipolar ones, but attractive in a few applications because of their lower power consumption and higher levels of integration
  - Beginning in mid-1980s, advances in design of MOS circuits, in particular **complementary MOS (CMOS)** circuits, tremendously increased their performance and popularity
  - Almost all new SSI, MSI, and LSI ICs use CMOS with equivalent functionality or better than TTL, higher speed and lower power consumption

CMOS Logic: CMOS Logic Levels

- A typical CMOS logic circuit operates from a 5-volt power supply
  - Any voltage in range 0–1.5 V $\longrightarrow$ logic 0
  - Any voltage in range 3.5–5.0 V $\longrightarrow$ logic 1
  - Voltages in intermediate range (1.5–3.5 V) are not expected to occur except during signal transitions
    - They yield undefined logic values
    - A circuit may interpret them as either 0 or 1

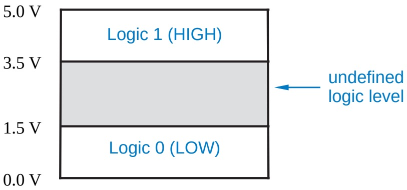


Figure 6: Logic levels for typical CMOS logic circuits.

CMOS Logic: MOS Transistors

- A MOS transistor can be modeled as a 3-terminal device that acts like a voltage-controlled resistance
    - An input voltage applied to one terminal controls resistance between remaining two terminals
- **Off transistor**
    - Its resistance is very high
- **On transistor**
    - Its resistance is very low

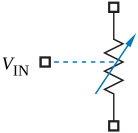
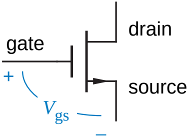


Figure 7: The MOS transistor as a voltage-controlled resistance.

CMOS Logic: MOS Transistors

- Two types of MOS transistors
  - *n*-channel
  - *p*-channel
- **n-channel MOS (NMOS) transistor**



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

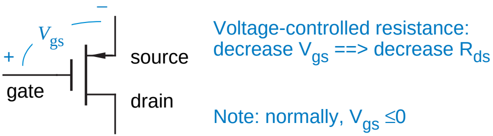
Note: normally, $V_{gs} \geq 0$

Figure 8: Circuit symbol for an *n*-channel MOS (NMOS) transistor.

- In Fig. 8
  - Orientation shows that drain is normally at a higher voltage than source
  - $V_{gs} = 0 \longrightarrow R_{ds}$ is very high
  - As we increase V_{gs}, R_{ds} decreases to a very low value

CMOS Logic: MOS Transistors

- **p-channel MOS (PMOS) transistor**



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

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

Figure 9: Circuit symbol for a *p*-channel MOS (PMOS) transistor.

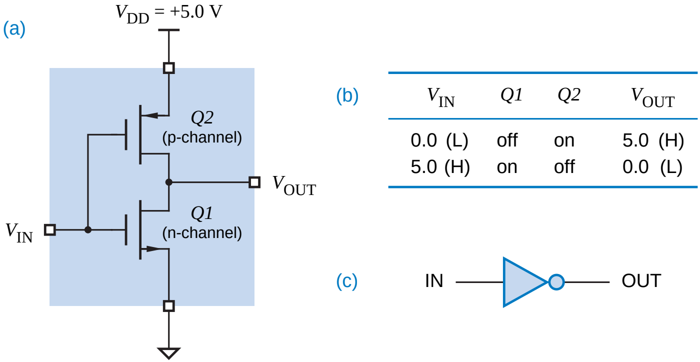
- In Fig. 9
  - Orientation shows that source is normally at a higher voltage than drain
  - $V_{gs} = 0 \longrightarrow R_{ds}$ is very high
  - As we decrease V_{gs}, R_{ds} decreases to a very low value

CMOS Logic: MOS Transistors

- Gate of a MOS transistor
  - Gate is capacitively coupled to source and drain
    - Power needed to charge and discharge this capacitance on each input-signal transition accounts for a nontrivial portion of a circuit's power consumption
  - Gate is separated from source and drain by an insulating material with a very high resistance
    - Almost no current flows from gate to source, or from gate to drain
    - Small amount of current that flows across this resistance is very small, less than one μA, and is called a **leakage current**
  - Gate voltage creates an electric field that enhances or retards flow of current between source and drain
    - This is "field effect" in "MOSFET" name

CMOS Logic: Basic CMOS Inverter Circuit

- NMOS and PMOS transistors are used together in a complementary way to form **CMOS logic**
- Logic inverter
  - The simplest CMOS circuit
  - Requires only one of each type of transistor



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

Figure 10: CMOS inverter: (a) circuit diagram; (b) functional behavior; (c) logic symbol.

CMOS Logic: Basic CMOS Inverter Circuit

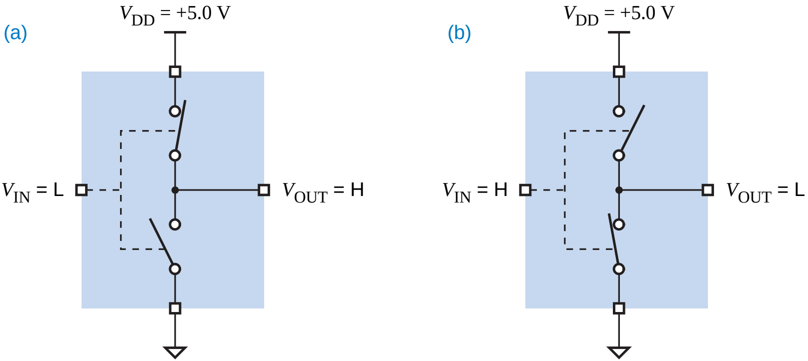


Figure 11: Switch model for CMOS inverter: (a) LOW input; (b) HIGH input.

CMOS Logic: Basic CMOS Inverter Circuit

- Another way of drawing CMOS circuits is shown in Fig. 12
  - Different symbols are used for *p*- and *n*-channel transistors to reflect their logical behavior
  - Inversion bubble on *p*-channel indicates its inverting behavior (compared to *n*-channel)

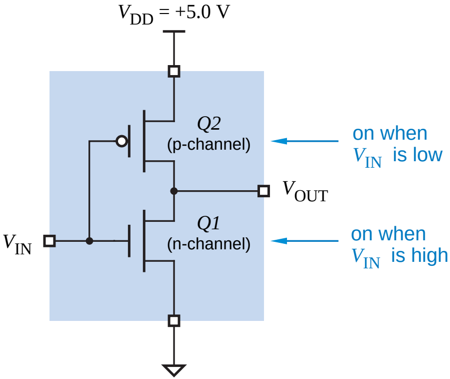
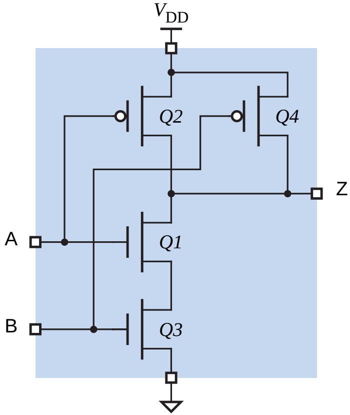


Figure 12: CMOS inverter logical operation.

CMOS Logic: CMOS NAND and NOR Gates

- A *k*-input NAND or NOR gate uses *k* *p*-channel and *k* *n*-channel transistors

(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)

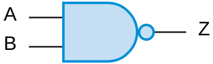


Figure 13: CMOS 2-input NAND gate: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: CMOS NAND and NOR Gates

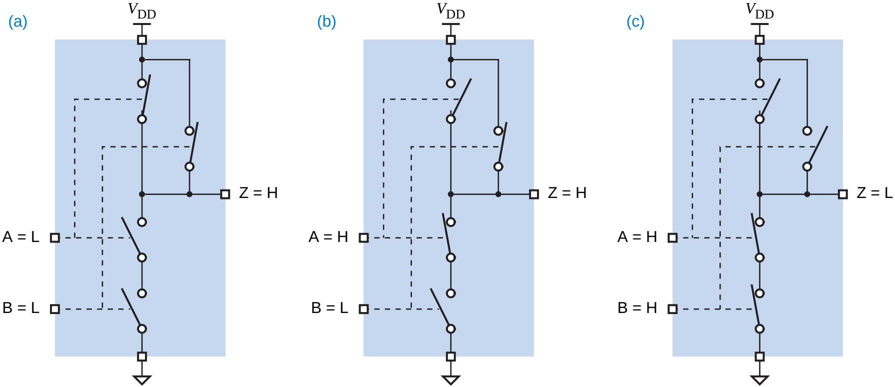
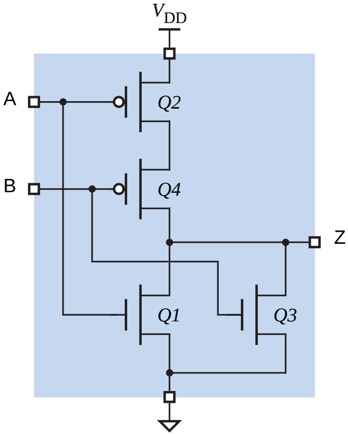


Figure 14: Switch model for CMOS 2-input NAND gate: (a) both inputs LOW; (b) one input HIGH; (c) both inputs HIGH.

CMOS Logic: CMOS NAND and NOR Gates

(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)

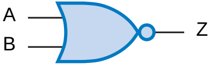


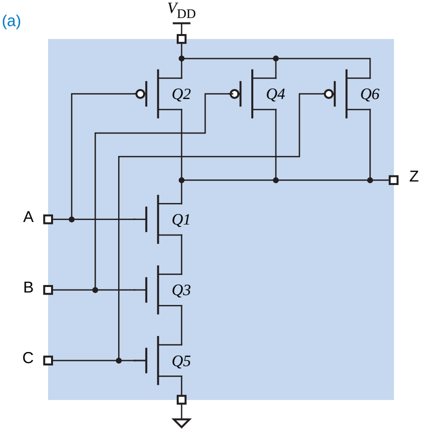
Figure 15: CMOS 2-input NOR gate: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: CMOS NAND and NOR Gates

- NAND vs. NOR
  - An *n*-channel transistor has lower "on" resistance than a *p*-channel
  - When transistors are put in series, *k* *n*-channel transistors have lower "on" resistance than do *k* *p*-channel ones
  - As a result, a *k*-input NAND gate is faster than a *k*-input NOR gate

CMOS Logic: Fan-In

- Logic family's **fan-in**
  - Number of inputs that a gate can have in a particular logic family
- An *n*-input gate has *n* series and *n* parallel transistors



(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

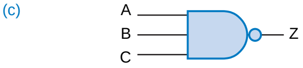


Figure 16: CMOS 3-input NAND gate: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: Fan-In

- Additive ”on” resistance of series transistors limits fan-in of CMOS gates, typically to 4 for NOR gates and 6 for NAND gates
  - As number of inputs is increased, designers may compensate by increasing the size of series transistors to reduce their resistance and corresponding switching delay
    - At some point, this becomes inefficient
  - Gates with a large number of inputs can be made faster and smaller by cascading gates with fewer inputs

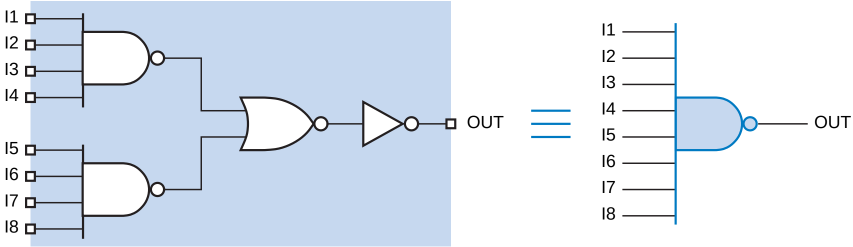
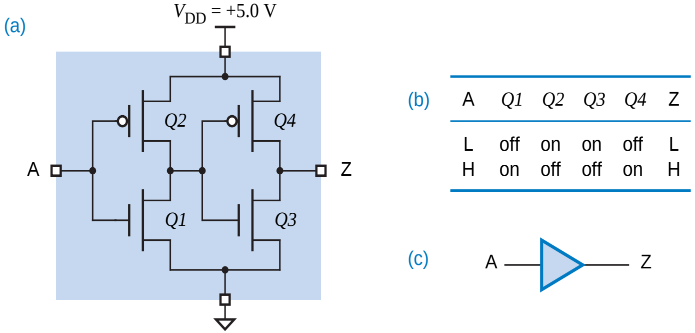


Figure 17: Logic diagram equivalent to the internal structure of an 8-input CMOS NAND gate; (total delay through a 4-input NAND, a 2-input NOR, an inverter) $<$ (delay of a one-level 8-input NAND).

CMOS Logic: Noninverting Gates

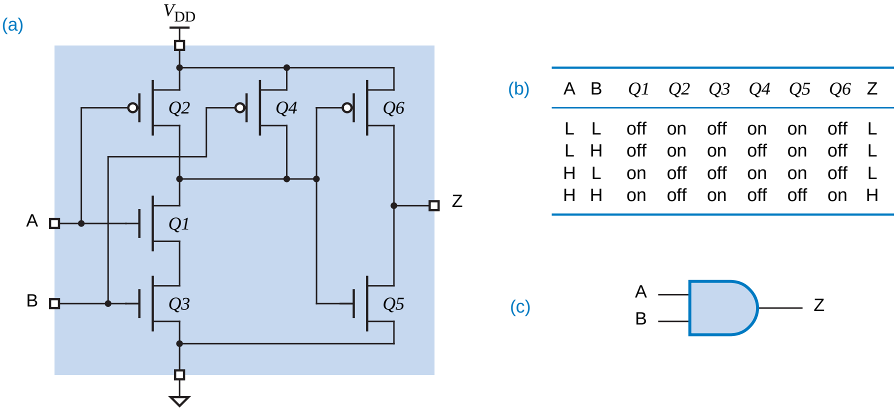
- In CMOS, the simplest gates are inverters, and then NAND and NOR
  - Inversion comes for free
  - Not possible to design a noninverting gate with a smaller number of transistors than an inverting one
- CMOS noninverting buffer, AND, and OR gates are obtained by connecting an inverter to output of corresponding inverting gate



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

Figure 18: CMOS noninverting buffer: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: Noninverting Gates



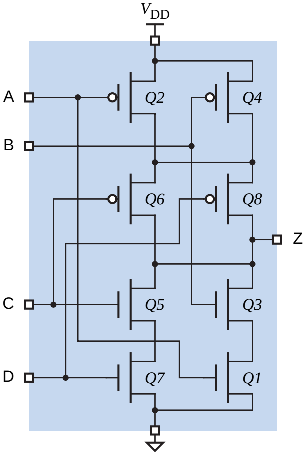
(b)

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

Figure 19: CMOS 2-input AND gate: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: AND-OR-INVERT, OR-AND-INVERT Gates

(a)



(b)

A	B	C	D	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	L
L	H	L	L	off	on	on	off	off	on	off	on	H
L	H	L	H	off	on	on	off	off	on	on	off	H
L	H	H	L	off	on	on	off	on	off	off	on	H
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	off	on	on	off	H
H	L	H	L	on	off	off	on	on	off	off	on	H
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	off	on	L
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L

Figure 20: CMOS AND-OR-INVERT (AOI) gate: (a) circuit diagram; (b) function table.

CMOS Logic: AND-OR-INVERT, OR-AND-INVERT Gates

- CMOS circuits can perform two levels of logic with just a single "level" of transistors
- Fig. 20
  - A 2-wide, 2-input CMOS **AND-OR-INVERT (AOI) gate**
  - Transistors can be added to or removed from this circuit to obtain an AOI function with a different number of ANDs or inputs per AND
  - Q1–Q8 depend only on input signal connected to the corresponding transistor's gate
  - Z never connected to both V_{DD} and ground for any input combination
    - Otherwise output would be a nonlogic value somewhere between LOW and HIGH, and output structure would consume excessive power due to low-impedance connection between V_{DD} and ground

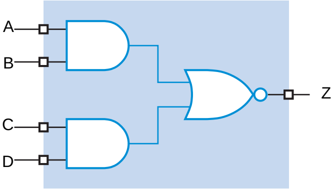
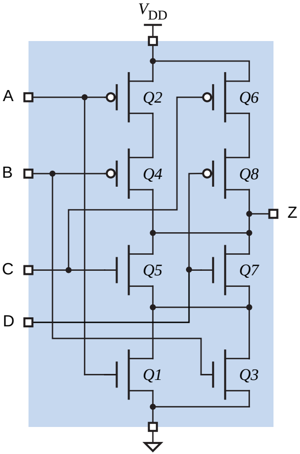


Figure 21: Logic diagram for CMOS AOI gate shown in Fig. 20.

CMOS Logic: AND-OR-INVERT, OR-AND-INVERT Gates

(a)



(b)

A	B	C	D	*Q1*	*Q2*	*Q3*	*Q4*	*Q5*	*Q6*	*Q7*	*Q8*	Z
L	L	L	L	off	on	off	on	off	on	off	on	H
L	L	L	H	off	on	off	on	off	on	on	off	H
L	L	H	L	off	on	off	on	on	off	off	on	H
L	L	H	H	off	on	off	on	on	off	on	off	H
L	H	L	L	off	on	on	off	off	on	off	on	H
L	H	L	H	off	on	on	off	off	on	on	off	L
L	H	H	L	off	on	on	off	on	off	off	on	L
L	H	H	H	off	on	on	off	on	off	on	off	L
H	L	L	L	on	off	off	on	off	on	off	on	H
H	L	L	H	on	off	off	on	off	on	on	off	L
H	L	H	L	on	off	off	on	on	off	off	on	L
H	L	H	H	on	off	off	on	on	off	on	off	L
H	H	L	L	on	off	on	off	off	on	off	on	H
H	H	L	H	on	off	on	off	off	on	on	off	L
H	H	H	L	on	off	on	off	on	off	off	on	L
H	H	H	H	on	off	on	off	on	off	on	off	L

Figure 22: CMOS OR-AND-INVERT (OAI) gate: (a) circuit diagram; (b) function table.

CMOS Logic: AND-OR-INVERT, OR-AND-INVERT Gates

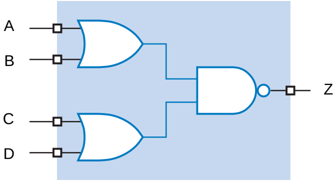


Figure 23: Logic diagram for CMOS OAI gate shown in Fig. 22.

- CMOS AOI and OAI gates are very appealing
    - They perform two levels of logic with one level of delay
    - HDL synthesis tools can automatically convert AND/OR logic into AOI gates when appropriate

CMOS Logic: Transmission Gates

- CMOS **transmission gate**
  - A *p*-channel and *n*-channel transistor pair connected together to form a logic-controlled switch

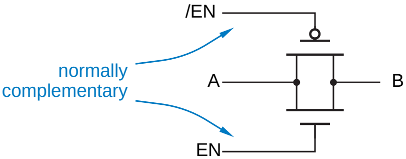


Figure 24: CMOS transmission gate.

- In Fig. 24
  - Input signals EN and EN_L are always at opposite levels
  - Once the gate is enabled, propagation delay from A to B (or vice versa) is very short
  - Because of their short delays and simplicity, these gates are often used internally in larger-scale CMOS devices such as multiplexers and flip-flops

CMOS Logic: Transmission Gates

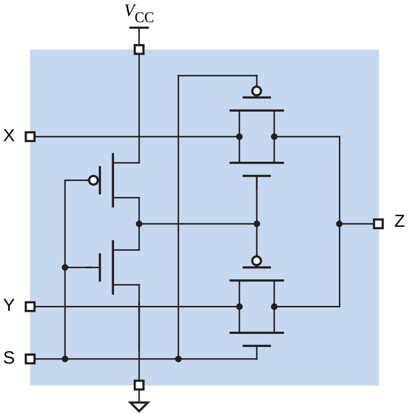
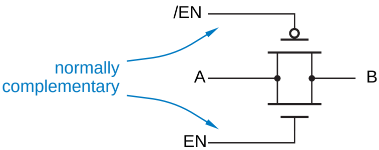


Figure 25: Two-input multiplexer using CMOS transmission gates.

- In Fig. 25
  - When S is LOW, X input is connected to Z output
  - When S is HIGH, Y is connected to Z

CMOS Logic: Transmission Gates

- In transmission gate
    - An "on" *p*-channel cannot conduct a LOW voltage between A and B very well
    - An "on" *n*-channel cannot conduct a HIGH voltage between A and B very well
    - But parallel transistors cover entire voltage range fine
    - Hence two transistors are used



CMOS Logic: Schmitt-Trigger Inputs

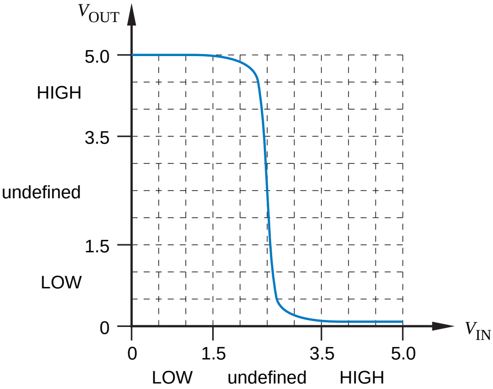


Figure 26: Typical input-output transfer characteristic of a CMOS inverter.

CMOS Logic: Schmitt-Trigger Inputs

- A **Schmitt trigger** is a circuit that uses feedback internally to shift switching threshold depending on whether input is changing from LOW to HIGH or from HIGH to LOW

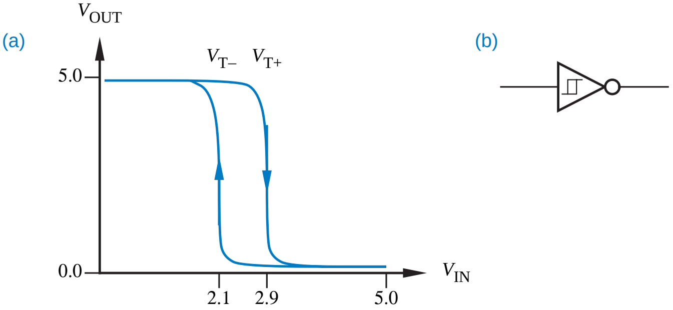


Figure 27: A Schmitt-trigger inverter: (a) input-output transfer characteristic; (b) logic symbol.

CMOS Logic: Schmitt-Trigger Inputs

- In Fig. 27
  - Switching threshold for positive-going input changes (V_{T+}) is 2.9 V, and for negative-going input changes (V_{T-}) is 2.1 V
  - Difference between the two thresholds is called **hysteresis**
  - Schmitt-trigger inverter provides about 0.8 V of hysteresis

CMOS Logic: Schmitt-Trigger Inputs

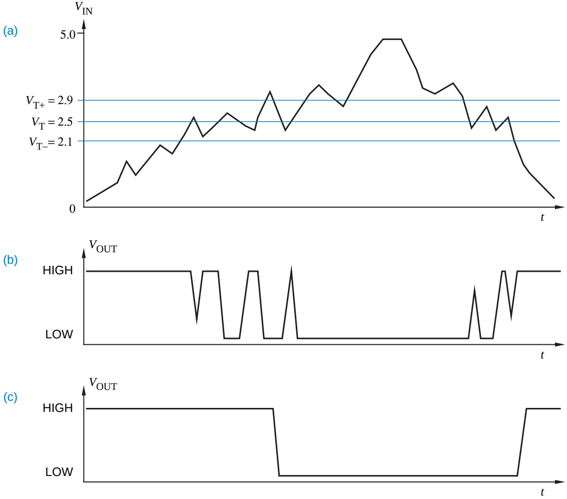


Figure 28: Device operation with slowly changing inputs: (a) a noisy, slowly changing input; (b) output produced by an ordinary inverter; (c) output produced by an inverter with 0.8 V of hysteresis.

CMOS Logic: Schmitt-Trigger Inputs

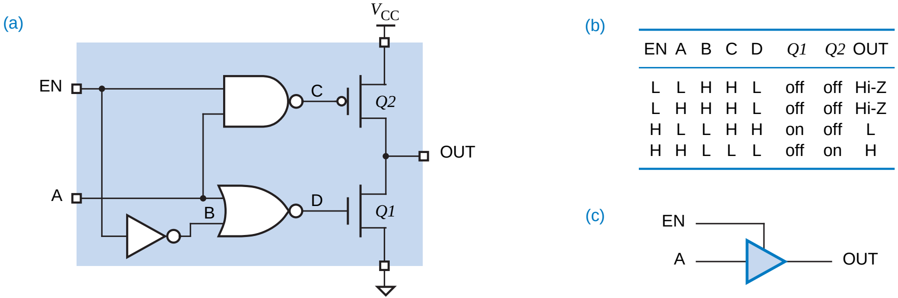
- In Fig. 28
  - (a) shows an input signal with long rise and fall times and about 0.5 V of noise on it
  - An ordinary inverter, without hysteresis, has same switching threshold for both positive-going and negative-going transitions, $V_T \approx 2.5$ V
  - Ordinary inverter responds to noise, producing multiple output changes each time noisy input voltage crosses switching threshold
  - A Schmitt-trigger inverter does not respond to noise, because its hysteresis is greater than noise amplitude
- Schmitt-trigger inputs have better noise immunity than ordinary gate inputs for signals with transmission-line reflections or long rise and fall times
  - Such signals occur in physically long connections, such as input-output buses and computer interface cables

CMOS Logic: Three-State Outputs

- Logic outputs have two normal states, LOW (0) and HIGH (1)
- Some outputs have a third electrical state that is not a logic state, called **high-impedance**, **Hi-Z**, or **floating state**
  - In this state, output behaves as if it is not connected to circuit
  - An output with three possible states is called a **three-state output** or a **tri-state output**
  - Three-state devices have an extra input, called "output enable" or "output disable," for placing device's output(s) in high-impedance state
  - A **three-state bus** is created by wiring several three-state outputs together
    - At most one output should be enabled at any time

CMOS Logic: Three-State Outputs

- The most basic three-state device is **three-state buffer** (= three-state driver)



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

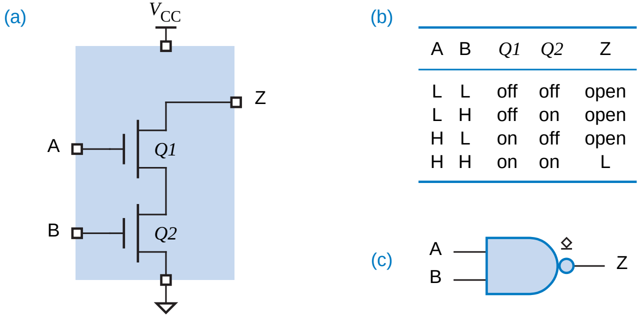
Figure 29: CMOS three-state buffer: (a) circuit diagram; (b) function table; (c) logic symbol.

CMOS Logic: Three-State Outputs

- Devices with three-state outputs are designed so that output-enable delay (Hi-Z to LOW or HIGH) is somewhat longer than output-disable delay (LOW or HIGH to Hi-Z)
    - Thus, if a control circuit activates one device's output-enable input and simultaneously deactivates a second's, the second device is guaranteed to enter Hi-Z state before the first places a HIGH or LOW level on bus
- If two three-state outputs on the same bus are enabled at the same time and try to maintain opposite states, a nonlogic voltage is produced on bus
    - If fighting is only momentary, devices probably will not be damaged
    - But large current drain through tied outputs can produce noise pulses that affect circuit behavior elsewhere in system

CMOS Logic: Open-Drain Outputs

- *p*-channel transistors in CMOS output structures provide **active pull-up**
  - They actively pull up output voltage on a LOW-to-HIGH transition
  - These transistors are omitted in gates with **open-drain outputs**



A	B	*Q1*	*Q2*	Z
L	L	off	off	open
L	H	off	on	open
H	L	on	off	open
H	H	on	on	L

Figure 30: Open-drain CMOS NAND gate: (a) circuit diagram; (b) function table; (c) logic symbol.

- In Fig. 30, drain of topmost *n*-channel is left unconnected internally
  - If output is not LOW, it is "open"

CMOS Logic: Open-Drain Outputs

- An open-drain output requires an external **pull-up resistor** to provide **passive pull-up** to HIGH level

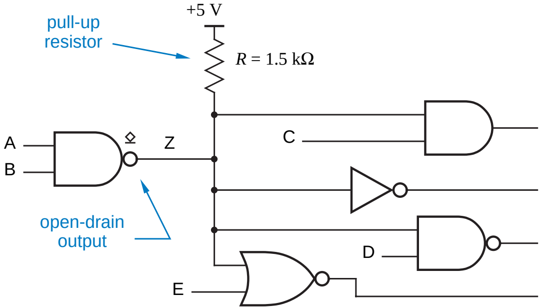


Figure 31: Open-drain CMOS NAND gate driving a load.

CMOS Logic: Open-Drain Outputs

- For the highest possible speed, an open-drain output's pull-up resistor should be as small as possible
  - This minimizes RC time constant for LOW-to-HIGH transitions (rise time)
  - The minimum resistance is determined by open-drain output's maximum sink current, I_{OLmax}
  - E.g., in HC- and HCT-series CMOS

$$I_{OLmax} = 4\ mA \longrightarrow \text{pull-up resistor}_{\text{min}} = \frac{5.0\ V}{4\ mA} = 1.25\ k\Omega$$

    - Since this is an order of magnitude greater than "on" resistance of p-channel transistors, LOW-to-HIGH output transitions are much slower for an open-drain gate than for standard gate with active pull-up

CMOS Logic: Open-Drain Outputs

- Example: in Fig. 31
  - Open-drain gate is HC-series CMOS ⟶ "on" resistance of output in LOW state = 80 Ω
  - Pull-up resistance = 1.5 $k\Omega$
  - Load capacitance = 100 pF

$$\begin{aligned} RC \text{ time constant for a HIGH-to-LOW transition} &= \text{output's fall time} \\ &= 80\ \Omega \times 100\ pF = 8\ ns \end{aligned}$$

$$\begin{aligned} RC \text{ time constant for a LOW-to-HIGH transition} &= \text{output's rise time} \\ &= 1.5\ k\Omega \times 100\ pF = 150\ ns \end{aligned}$$

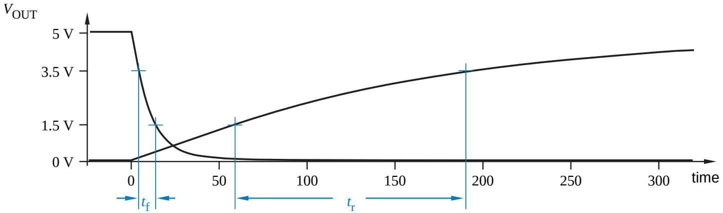


Figure 32: Rising and falling transitions of an open-drain CMOS output.

CMOS Logic: Multisource Buses

- Open-drain outputs can be tied together to allow several devices, one at a time, to put information on a common bus
  - At any time all but one of outputs are in their HIGH (open) state
  - Control circuitry selects particular device that is allowed to drive the bus at any time
- In Fig. 33
  - At most one control bit is HIGH at any time, enabling complement of corresponding data bit to be passed through bus

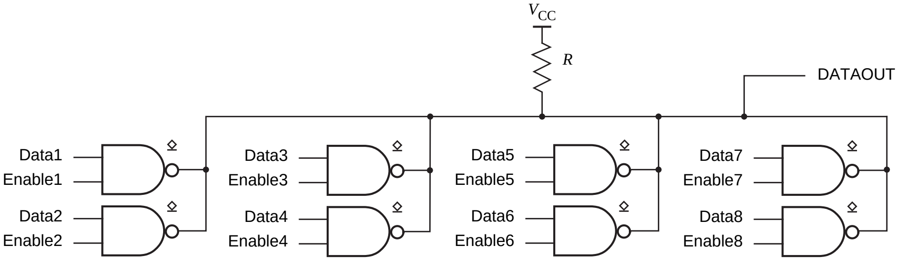


Figure 33: Eight open-drain outputs driving a bus.

CMOS Logic: Wired Logic

- If outputs of several open-drain gates are tied together with a single pull-up resistor, **wired logic** is performed
- An AND function is obtained, since wired output is HIGH iff all of individual gate outputs are HIGH (open)
  - Any output going LOW is sufficient to pull wired output LOW

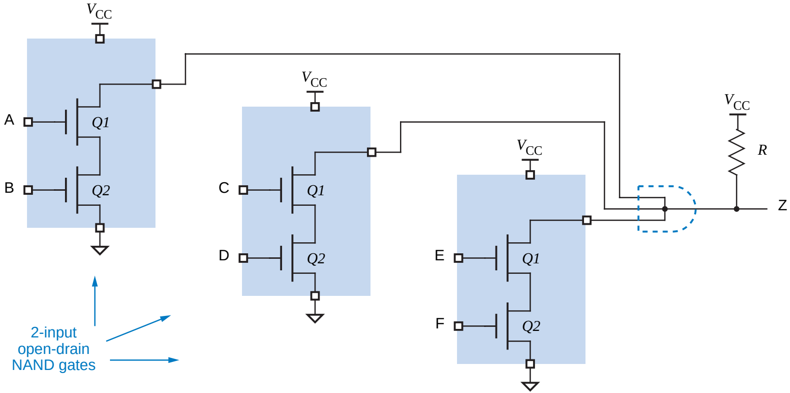


Figure 34: Wired-AND function on three open-drain NAND-gate outputs.

CMOS Logic: Wired Logic

- In Fig. 34
  - If any of individual 2-input NAND gates has both inputs HIGH, it pulls wired output LOW
  - Otherwise, pull-up resistor *R* pulls wired output HIGH
- Wired logic cannot be performed using gates with active pull-up
  - **Fighting**: Two such outputs wired together and trying to maintain opposite logic values result in a very high current flow and an abnormal output voltage
  - Exact output voltage depends on relative strengths of fighting transistors
  - If outputs fight continuously for more than a few seconds, chips can get hot enough to sustain internal damage

CMOS Logic: Wired Logic

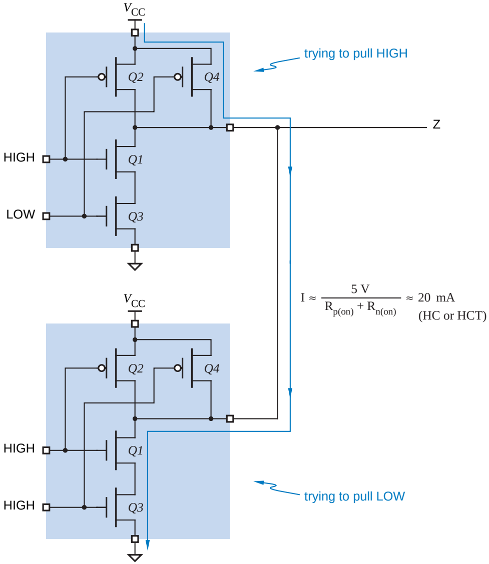


Figure 35: Two outputs trying to maintain opposite logic values on the same line

References

John F. Wakerly, *Digital Design: Principles and Practices (4th Edition)*, Prentice Hall, 2005.