CS3120
Introduction of Integrated Circuit Design

Chapter 2
Logic Design With MOSFET

Andy, Yu-Guang Chen
Associate Professor, Department of EE, National Central University
Adjunct Assistant Professor, Department of CS, National Tsing Hua University
andyygchen@ee.ncu.edu.tw

Outline

- Ideal Switches and Boolean Operations
- MOSFETs as Switches
- Basic Logic Gates in CMOS
- Complex Logic Gates in CMOS
- Transmission Gate Circuits
- Clocking and Dataflow Control

2.1 Ideal Switches and Boolean Operation

- CMOS integrated circuits use bi-directional devices called MOSFETs as logic switches
- This chapter examines the logical characteristics of MOSFETs and develops techniques for building digital networks
- All digital designs are based on primitive logic operations
- The first task in our study of VLSI will be to create electronic logic gates that can be used as building blocks in complex switching networks

2.1 Ideal Switches and Boolean Operation

- Logic gates are created by sets of controlled switches
- In this idealized situation, (open or close) of the switch is controlled by the value of A
- **Assert-high** controlled switch
  - $y = x \cdot A \ \ iff \ \ A = 1$
  - The relationship between x and y is undefined if A=0

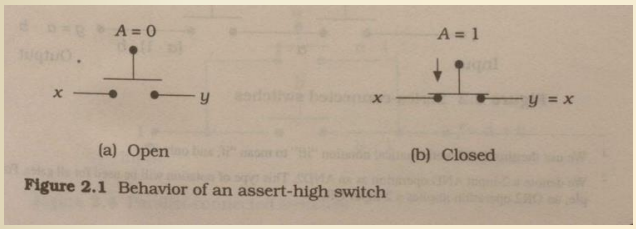


Figure 2.1 Behavior of an assert-high switch

2.1 Ideal Switches and Boolean Operation

- We can create a logic network by combining the concept of an ideal switch with a voltage source
- The two switches are in series with each other
  - $g = (a \cdot 1) \cdot b = a \cdot b$
  - a = 1 AND b = 1 ➔ output = 1
- AND2 operation (2-input AND operation)

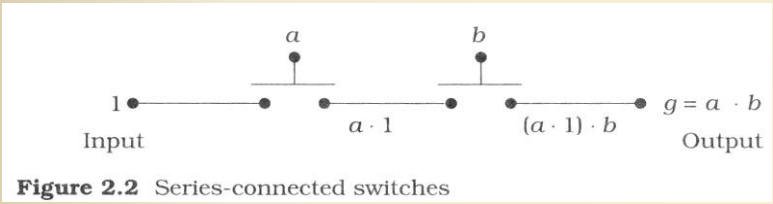


Figure 2.2 Series-connected switches

2.1 Ideal Switches and Boolean Operation

- We can create a logic network by combining the concept of an ideal switch with a voltage source
- The two switches are in parallel with each other
  - $g = (a \cdot 1) + (b \cdot 1) = a + b$
  - a = 1 OR b = 1 ➔ output = 1
- OR2 operation (2-input OR operation)

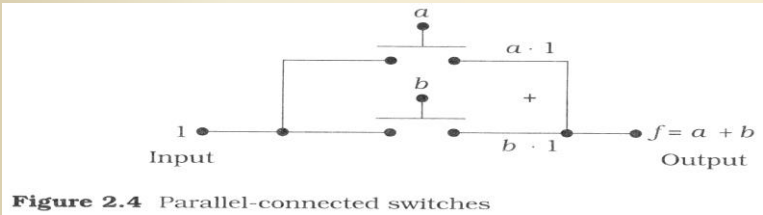


Figure 2.4 Parallel-connected switches

2.1 Ideal Switches and Boolean Operation

- The switch drawings will be called schematic diagram
- Wire 1 and Wire 2 are assumed to be totally separate
  - The signal a on Wire 1 has no relationship to the signal b on Wire 2
- If we wish to create a connection, we will use a "dot"
  - The two wires are connected so that placing a signal a on one of the lines results in the same value on all points of both lines

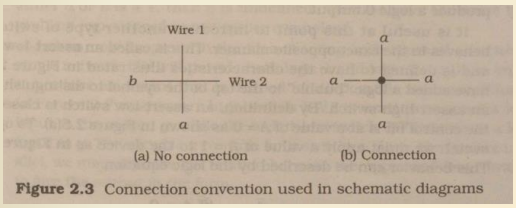


Figure 2.3 Connection convention used in schematic diagrams

2.1 Ideal Switches and Boolean Operation

- **Assert-low** controlled switch
  - $y = x \cdot \bar{A}$ iff $A = 0$
  - The relationship between x and y is undefined if A=1
- The assert-high and assert-low switch behave in a complementary manner

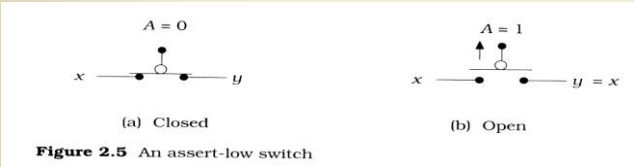


Figure 2.5 An assert-low switch

2.1 Ideal Switches and Boolean Operation

- The two switches are in series with each other
  - $g = (\bar{a} \cdot 1) \cdot \bar{b} = \bar{a} \cdot \bar{b} = \overline{a + b}$
  - a = 0 AND b = 0 ➔ output = 1
  - If either a or b is a 1, then g is undefined
- NOR2 operation

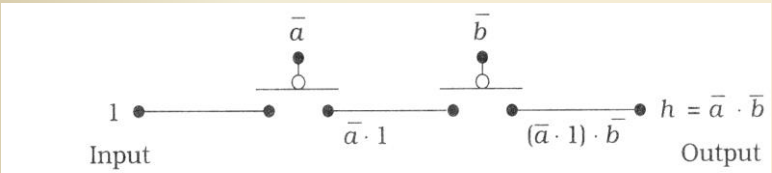


Figure 2.6 Series-connected complementary switches

2.1 Ideal Switches and Boolean Operation

- Progress to the idea of using both types of switches in a single network
- Provide both logic 1 and logic 0 inputs in an effort to produce an output that is defined for all possible input combinations
- $y = \bar{a}$ ➔ NOT operation

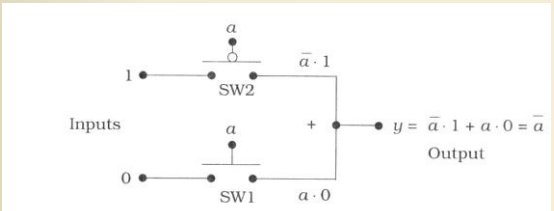


Figure 2.7 A switch-based NOT gate

2.2 MOSFET as Switches

- MOSFET stands for Metal Oxide-Semiconductor Field Effect Transistor
  - Behave like the idealized switches
  - Obey circuit equations
  - Performance limited by the law of physics
- We concentrate on creating switching models for the devices in this chapter
- Complementary MOS
  - N channel MOSFET (nFET) behaves as an assert-high switch
  - P channel MOSFET (pFET) behaves as an assert-low switch

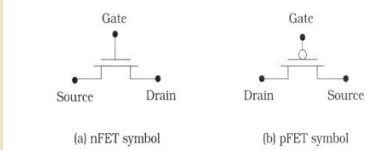


Figure 2.9 Symbols used for nFETs and pFETs

2.2 MOSFET as Switches

- The gate terminal acts as the control electrode for the device
- Applying a voltage on the gate electrode determines the current flow between drain and source terminals
- MOSFETs are intrinsically electronic devices
- To use them, we need to translate between Boolean values and electrical parameters

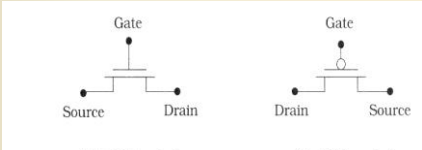


Figure 2.9 Symbols used for nFETs and pFETs

2.2 MOSFET as Switches

- Two power supply voltages V_{DD} and V_{ss} are defined
- Early generations of silicon MOS logic circuits used both positive and negative supply voltages
- Modern designs require only a single positive voltage V_{DD} and the ground connection
  - V_{DD} = 5V or 3.3V or lower
  - V_{SS} = 0V

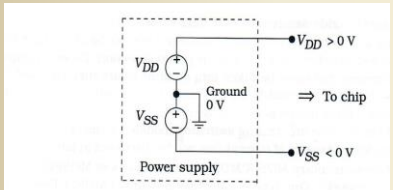


Figure 2.10 Dual power supply voltages

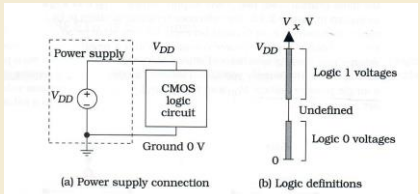


Figure 2.11 Single voltage power supply

2.2 MOSFET as Switches

- We can now define the relationship between logic variables and voltages
  - Boolean variables ➔ $x = 0\ or\ x = 1$
  - Represent the variable x using a voltage V_x such that $0 \le V_x \le V_{DD}$
  - x = 0 means that V_x = 0 V
  - x = 1 means that $V_x = V_{DD}$
- Realistic circuits are more lenient
  - Low voltages correspond to logic 0 values
  - High voltages correspond to logic 1 values

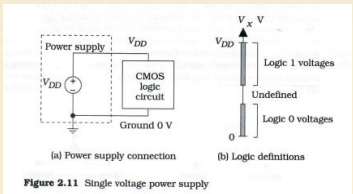
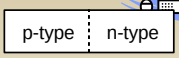
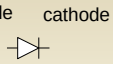


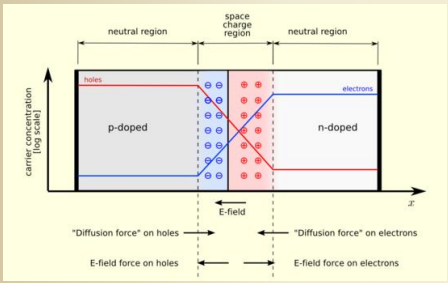
Figure 2.11 Single voltage power supply

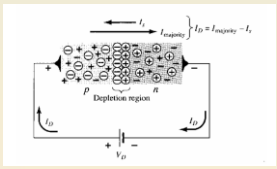
2.2 MOSFET as Switches



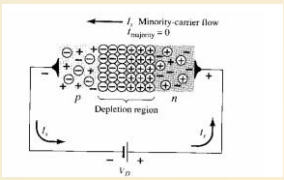


- P-N Junctions
  - A junction between p-type and n-type semiconductor forms a diode
  - Current flows only in one direction
  - Potential barrier





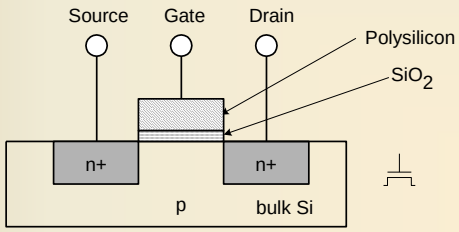
Forward bias



Reverse bias

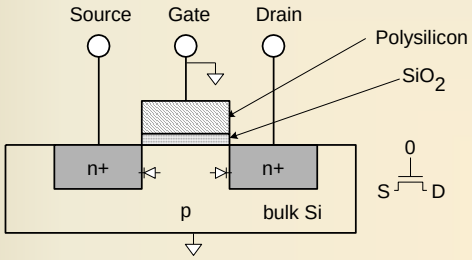
2.2 MOSFET as Switches

- nMOS Transistor
- Gate–oxide–body stack looks like a capacitor
  - Gate and body are conductors
  - SiO_2 (oxide) is a very good insulator
  - Called metal–oxide–semiconductor (MOS) capacitor
  - Even though gate is no longer made of metal



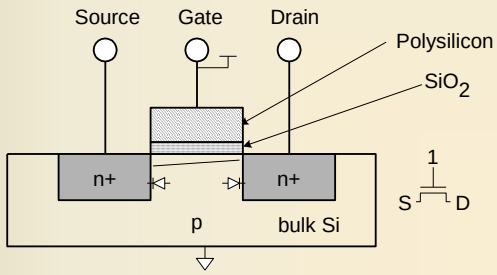
2.2 MOSFET as Switches

- Body is commonly tied to ground (0 V)
- When the gate is at a low voltage:
  - P-type body is at low voltage
  - Source-body and drain-body diodes are OFF
  - No current flows, transistor is OFF



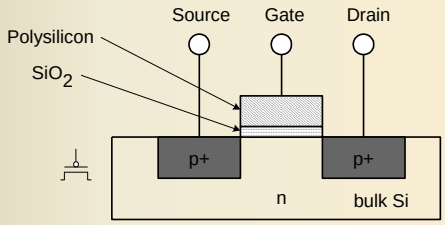
2.2 MOSFET as Switches

- When the gate is at a high voltage:
  - Positive charge on gate of MOS capacitor
  - Negative charge attracted to body
  - Inverts a channel under gate to n-type
  - Now current can flow through n-type silicon from source through channel to drain, transistor is ON



2.2 MOSFET as Switches

- pMOS Transistor
- Similar, but doping and voltages reversed
  - Body tied to high voltage (V_{DD})
  - Gate low: transistor ON
  - Gate high: transistor OFF
  - Bubble indicates inverted behavior



2.2 MOSFET as Switches

- Ideally, an nFET behaves like an assert-high switch
- The pFET is exactly opposite in that it behaves like an assert-low switch
- MOSFETs allow us to design logic circuits using the technique assert-high and assert-low switching networks

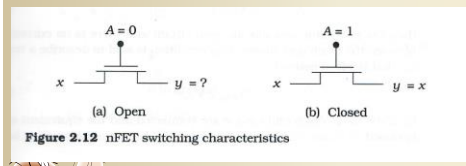


Figure 2.12 nFET switching characteristics

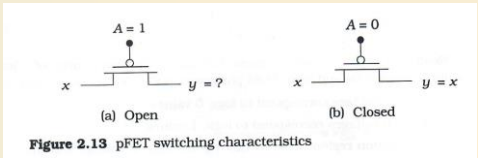


Figure 2.13 pFET switching characteristics

2.2 MOSFET as Switches

- We want to define a range of voltages for both cases of A = 0 and A=1
- **Threshold voltage (V_T)**
  - Established during the manufacturing process
  - Is assumed to be a given value to the designer
  - nFETs and pFET have different threshold voltages

2.2 MOSFET as Switches

- nFET has a threshold voltage V_{Tn}, a positive number (0.5V~0.7V)
  - $V_{GS} \leq V_{Tn}$ ➔ open circuit ➔ transistor off
  - $V_{GS} \geq V_{Tn}$ ➔ close circuit ➔ transistor on
  - $V_A = V_{GSn}$

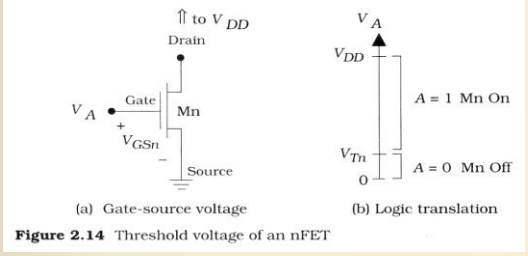


Figure 2.14 Threshold voltage of an nFET

2.2 MOSFET as Switches

- A pFET behaves in a complementary manner
- V_{Tp} is a negative number (-0.5V ~ -0.8V)
- We use $V_{SGp} = -V_{GSp}$
  - $V_{SGp} \leq |V_{Tp}|$ ➔ open circuit ➔ transistor off
  - $V_{SGp} \geq |V_{Tp}|$ ➔ close circuit ➔ transistor on

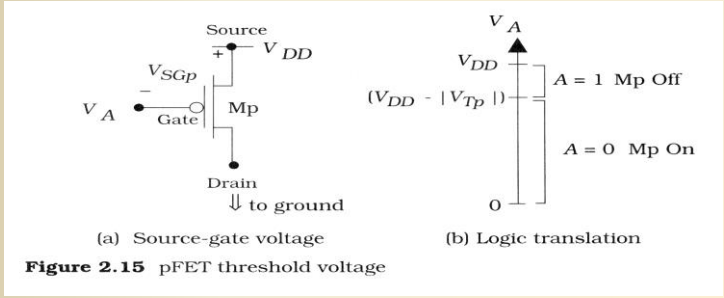


Figure 2.15 pFET threshold voltage

2.2 MOSFET as Switches

- It is important to note that the logic 0 and logic 1 voltage ranges of V_A are different for the two types of FETs
- There are regions of overlap for both A = 0 and A = 1 values
  - It can be used if a uniform definition is needed
- $V_A = 0$ and $V_A = V_{DD}$ valid for both devices

2.2 MOSFET as Switches

- Pass Characteristics
  - Ideal electrical switch can pass any voltage
  - MOSFETs are more limited in their capabilities and are not able to pass arbitrary voltages from source to drain or vice versa
  - Complementary MOS (CMOS)
    - Use pFET to pass logic 1 voltages of V_{DD}
    - Use nFETs to pass logic 0 voltages of V_{SS} = 0 V

2.2 MOSFET as Switches

- The pass characteristics of nFET
  - If a logic 0 is connected from left to the right
  - This results in an output voltage of $V_y = 0\ V$ as desired
  - If a V_{DD} is applied in the left, the output voltage is reduced to a value $V_y = V_{DD} - V_{Tn}$ ➔ Threshold voltage loss
  - nFET can only pass a weak logic 1 but strong logic 0
  - minimum value of the gate-source voltage need to maintain an on state is $V_{GSn} - V_{Tn}$

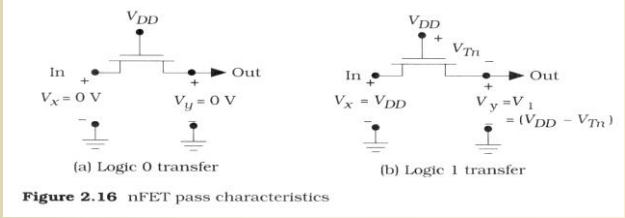


Figure 2.16 nFET pass characteristics

2.2 MOSFET as Switches

- The pass characteristics of pFET
  - If a logic 1 is connected from left to the right
  - This results in an output voltage of $V_y = V_{DD}$ as desired
  - If a V_{ss} is applied in the left, the output voltage can only drop to a value $|V_{Tp}|$ ➔ Threshold effect
  - pFET can pass a strong logic 1 but only weak logic 0
  - In order to keep the pFET on requires a minimum source-gate voltage of $V_{SGp} = |V_{Tp}|$

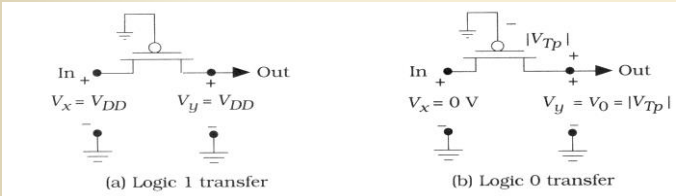


Figure 2.17 pFET pass characteristics

2.3 Basic Logic Gates in CMOS

- The concept of a general CMOS digital logic gate is in Fig. 2.18
- Use transistors to divert one of the supply voltages V_{dd} or 0V to output
- The upper and lower switch will not close/open at the same time
- They are **complementary pairs**
- The important of this behavior is that the nFET and pFET are electrical opposite ➔ translates directly into a coherent switching scheme

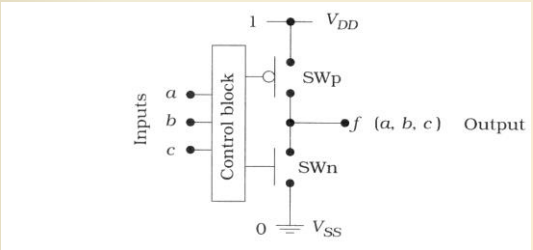


Figure 2.18 General CMOS logic gate

2.3 Basic Logic Gates in CMOS

- The operation of the general logic gate is shown in Figure 2.19
- The only missing feature in this model is the method used to control the output switches

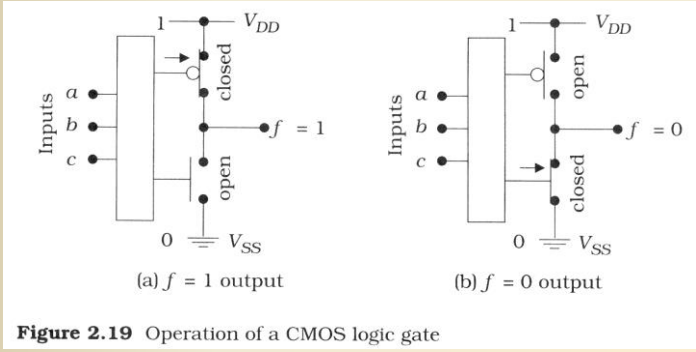
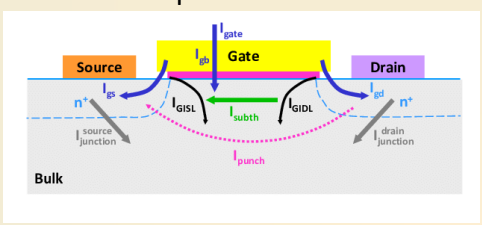


Figure 2.19 Operation of a CMOS logic gate

CMOS Power Consumption Problem

- Ideal CMOS gate does not consume power in the steady state (No current flow)
- But there is power consumption in the transition
  - The transitions are caused by changing values of input and output
  - If there is no change of input values, CMOS does not consume power (except leakage power)
- In reality, leakage current becomes an important issue
- When V_T goes down
  - Speed goes up
  - Leakage current goes up



2.3 Basic Logic Gates in CMOS

- The NOT gate
  - The NOT or INVERT function: $f(x) = NOT(x) = \bar{x}$
  - Only one transistor is on
  - Avoid both transistors are on or off ➔ ill-defined output

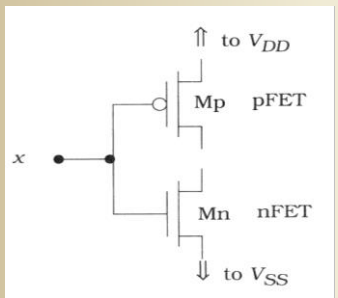


Figure 2.20 A CMOS complementary pair

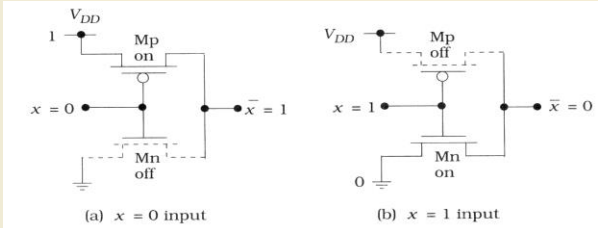
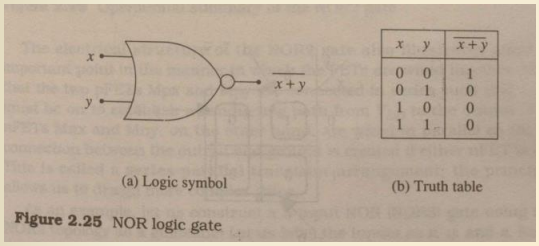


Figure 2.24 Operation of the CMOS NOT gate

2.3 Basic Logic Gates in CMOS

- The CMOS NOR Gate
  - Create 2-input NOR gate using the same principles
  - Use a complementary nFET/pFET pair for each input
  - Connect the output node to the power supply V_{DD} through pFETs
  - Connect the output node to ground through nFETs
  - Insure that the output is always a well-defined high or low voltage

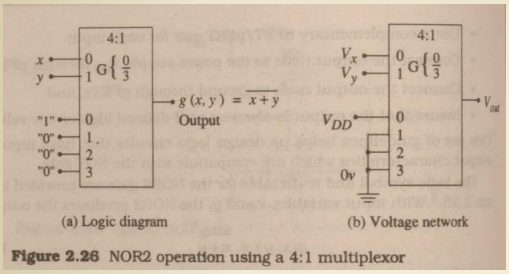


x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

Figure 2.25 NOR logic gate

2.3 Basic Logic Gates in CMOS

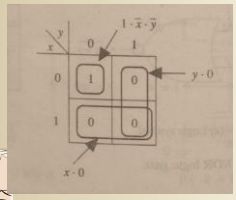
- NOR2 operation using a 4:1 multiplexor
  - $g(x,y) = \bar{x} \cdot \bar{y} \cdot 1 + \bar{x} \cdot y \cdot 0 + x \cdot \bar{y} \cdot 0 + x \cdot y \cdot 0 = \overline{x + y}$



x	y	$\overline{x+y}$
0	0	1
0	1	0
1	0	0
1	1	0

(b) Truth table

Figure 2.26 NOR2 operation using a 4:1 multiplexor



2.3 Basic Logic Gates in CMOS

- CMOS NOR2 circuit
  - The one-to-one correspondence between each line in the equation and the circuit is obvious
  - **Series-parallel** transistor arrangement

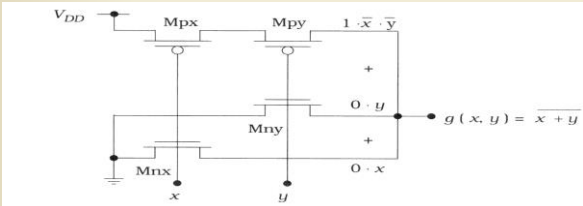


Figure 2.28 CMOS NOR2 gate

x	y	Mpx	Mpy	Mnx	Mny	g
0	0	on	on	off	off	1
0	1	on	off	off	on	0
1	0	off	on	on	off	0
1	1	off	off	on	on	0

Figure 2.29 Operational summary of the NOR2 gate

2.3 Basic Logic Gates in CMOS

- CMOS NOR3 circuit

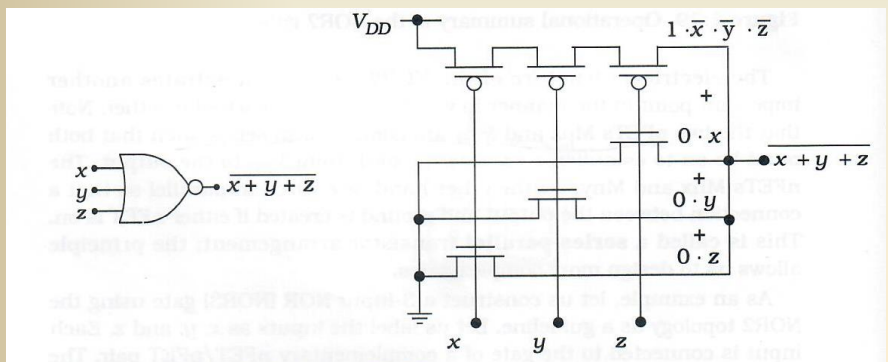


Figure 2.30 A NOR3 gate in CMOS

2.3 Basic Logic Gates in CMOS

- CMOS NAND Gate
  - NAND2 operation using a 4:1 multiplexor
  - $h(x,y) = \bar{x} \cdot \bar{y} \cdot 1 + \bar{x} \cdot y \cdot 1 + x \cdot \bar{y} \cdot 1 + x \cdot y \cdot 0 = \overline{x \cdot y}$

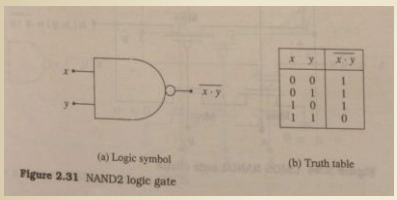


Figure 2.31 NAND2 logic gate

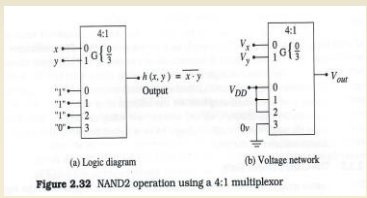


Figure 2.32 NAND2 operation using a 4:1 multiplexor

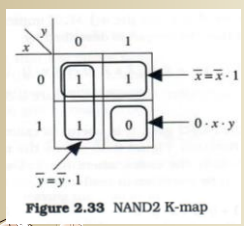


Figure 2.33 NAND2 K-map

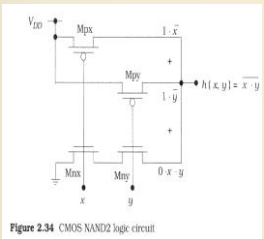
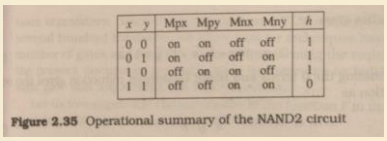


Figure 2.34 CMOS NAND2 logic circuit



x	y	Mpx	Mpy	Mnx	Mny	h
0	0	on	on	off	off	1
0	1	on	off	off	on	1
1	0	off	on	on	off	1
1	1	off	off	on	on	0

Figure 2.35 Operational summary of the NAND2 circuit

2.3 Basic Logic Gates in CMOS

- CMOS NAND3 circuit

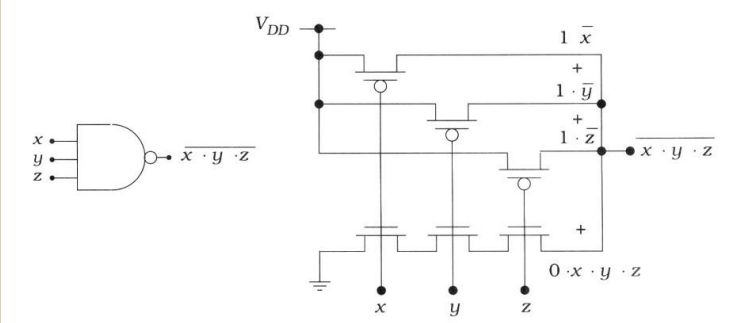


Figure 2.36 A NAND3 logic gate in CMOS

2.4 Complex Logic Gates in CMOS

- CMOS is able to create a single circuit that provide several primitive operations (NOT, AND, OR) in an integrated manner
- **Complex** or **combinational** logic gates
- Consider Boolean expression $F(a,b,c) = \overline{a \cdot (b+c)}$

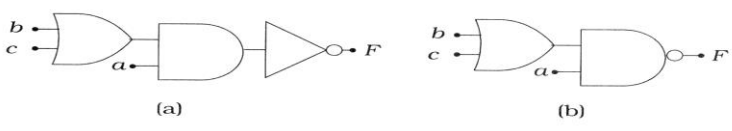


Figure 2.37 Logic function example

2.4 Complex Logic Gates in CMOS

- For pFET design

$$F(a,b,c) = \overline{a \cdot (b+c)} = \bar{a} + \overline{(b+c)} = \left[\bar{a} + \left(\bar{b} \cdot \bar{c}\right)\right] \cdot 1$$
$$= \bar{a} \cdot 1 + \left(\bar{b} \cdot \bar{c}\right) \cdot 1$$

- For nFET design

$F(a,b,c) = 0$ ➔ $a = 1\ AND\ (b+c) = 1$ ➔ $0 \cdot [a \cdot (b+c)]$

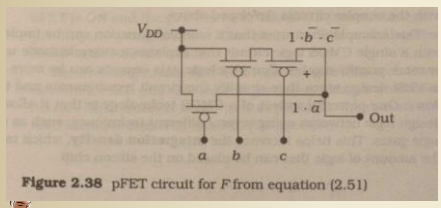


Figure 2.38 pFET circuit for F from equation (2.51)

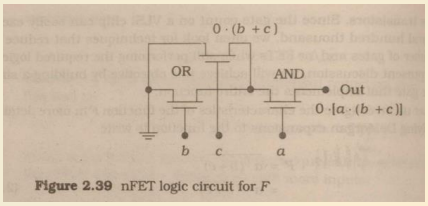


Figure 2.39 nFET logic circuit for F

2.4 Complex Logic Gates in CMOS

- Put nFET and pFET together

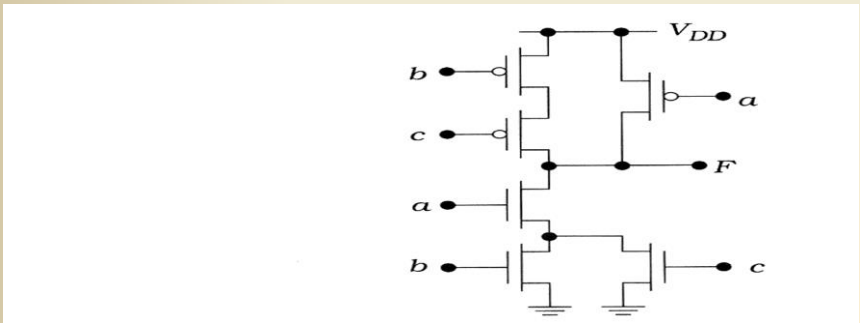


Figure 2.41 Finished complex CMOS logic gate circuit

2.4 Complex Logic Gates in CMOS

- Structured Logic Design
  - CMOS logic gates are intrinsic called inverting
  - Outputs always produces a NOT operation

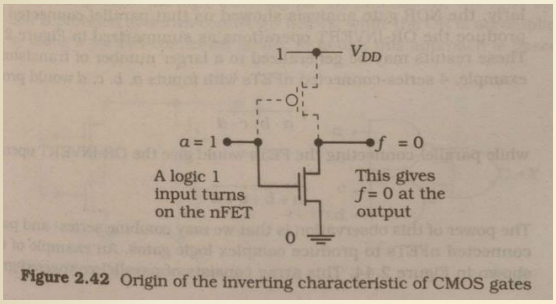


Figure 2.42 Origin of the inverting characteristic of CMOS gates

2.4 Complex Logic Gates in CMOS

- CMOS switching characteristics provide a natural means for implementing inverting logic forms such as AOI and OAI
  - AOI: and, or, inverter ➔ $X = \overline{(a \cdot b) + (c \cdot d)}$
  - OAI: or, and, inverter ➔ $Y = \overline{(a + e)(b + f)}$

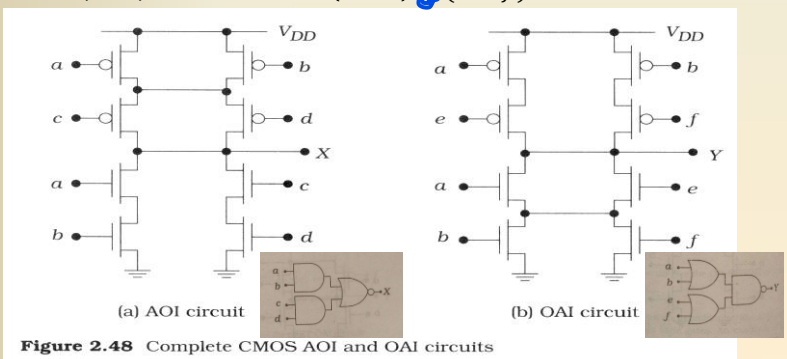


Figure 2.48 Complete CMOS AOI and OAI circuits

2.4 Complex Logic Gates in CMOS

- Consider the function $X = \overline{a + b \cdot (c + d)}$
- The nFET circuit can be constructed by using the following arrangements
  - Group 1: nFETs with inputs *c* and *d* are in parallel
  - Group 2: an nFET with input *b* is in series with Group 1
  - Group 3: an nFET with input *a* is in parallel with the Group 1- Group 2 circuit
- Each group of pFETs can be associated with the nFET group that has the same inputs
  - Group 1: pFETs with inputs *c* and *d* are in series
  - Group 2: a pFET with input *b* is in parallel with Group 1 pFETs
  - Group 3: a pFET with input a is in series with the Group 1-Group 2 pFETs

2.4 Complex Logic Gates in CMOS

- Consider the function $X = \overline{a + b \cdot (c + d)}$

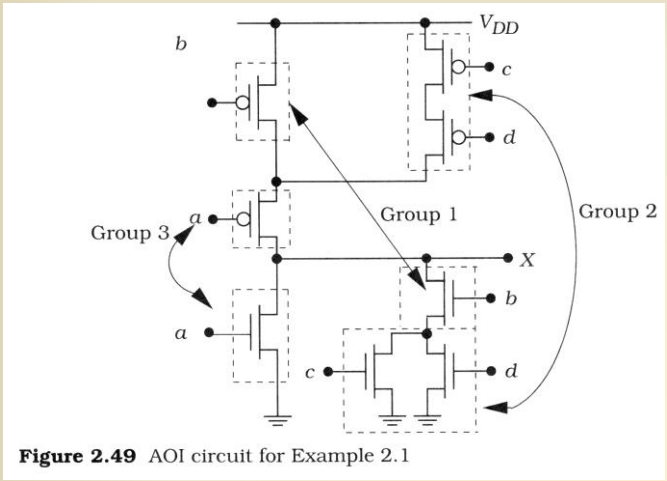


Figure 2.49 AOI circuit for Example 2.1

2.4 Complex Logic Gates in CMOS

- Bubble pushing
  - Both the nFETs and the pFETs are wired such that parallel-connected transistors give the OR operation, while series-connected FETs provide the AND operation

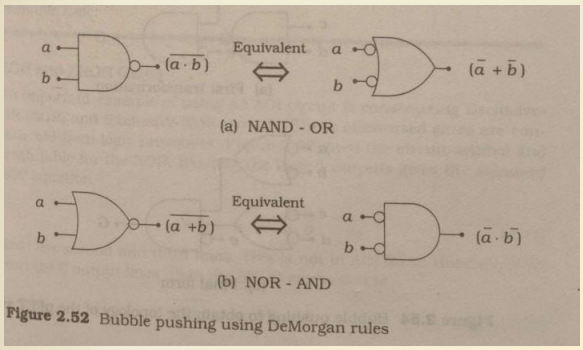


Figure 2.52 Bubble pushing using DeMorgan rules

2.4 Complex Logic Gates in CMOS

- XOR Gates
  - $a \oplus b = \bar{a} \cdot b + a \cdot \bar{b}$
  - $\overline{a \oplus b} = a \cdot b + \bar{a} \cdot \bar{b}$
  - $a \oplus b = \overline{\overline{a \oplus b}} = \overline{a \cdot b + \bar{a} \cdot \bar{b}}$ ➔AOI structure

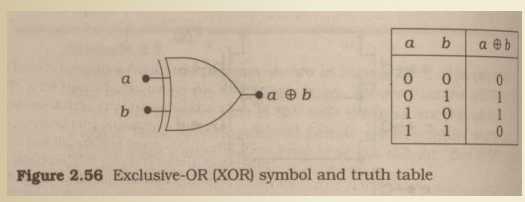
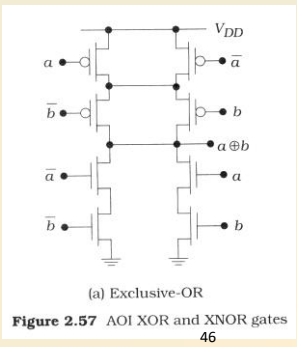


Figure 2.56 Exclusive-OR (XOR) symbol and truth table

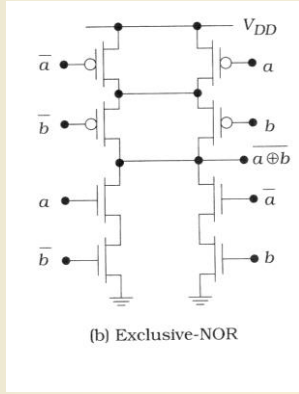


(a) Exclusive-OR

Figure 2.57 AOI XOR and XNOR gates

2.4 Complex Logic Gates in CMOS

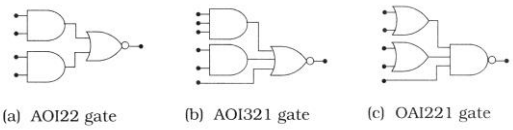
- XNOR Gates
  - $a \oplus b = \bar{a} \cdot b + a \cdot \bar{b}$
  - $\overline{a \oplus b} = \overline{\bar{a} \cdot b + a \cdot \bar{b}}$ ➔ AOI structure



(b) Exclusive-NOR

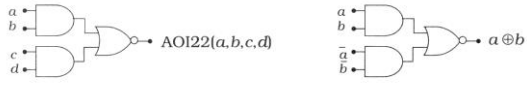
2.4 Complex Logic Gates in CMOS

- Generalized AOI and OAI Logic Gates
  - $AOI22(a,b,c,d) = \overline{a \cdot b + c \cdot d}$
  - $a \oplus b = AOI22(a, b, \bar{a}, \bar{b})$
  - $\overline{a \oplus b} = AOI22(a, \bar{b}, \bar{a}, b)$



(a) AOI22 gate (b) AOI321 gate (c) OAI221 gate

Figure 2.58 General naming convention



(a) General AOI22 gate (b) XOR

Figure 2.59 Application of an AOI22 gate

2.5 Transmission Gate Circuits

- A CMOS transmission gate is created by connecting an nFET and pFET in parallel
  - $y = x \cdot s \ \ iff \ s = 1$
- The pair acts as a good electrical switch
- TG is bi-directional switch

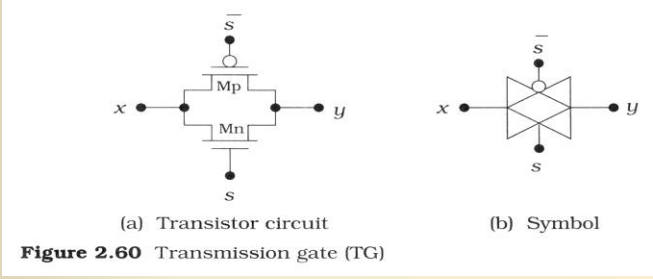


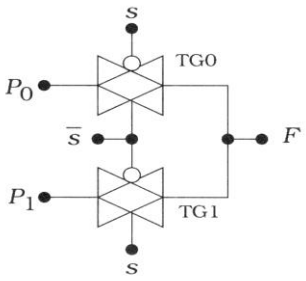
Figure 2.60 Transmission gate (TG)

2.5 Transmission Gate Circuits

- Transmission gates are useful because they can transmit the entire voltage range [0, V_{DD}] from left to right (or vice versa)
- The main drawback of using TGs in modern VLSI is they require two FETs and an implied inverter that takes s and produce $\bar{s}$

2.5 Transmission Gate Circuits

- 2:1 Multiplexors
  - $F = P_0 \cdot \bar{s} + P_1 \cdot s$



s	TG0	TG1	F
0	Closed	Open	P_0
1	Open	Closed	P_1

Figure 2.61 A TG-based 2-to-1 multiplexor

- 4:1 Multiplexors
  - $F = P_0 \cdot \overline{s_1} \cdot \overline{s_0} + P_1 \cdot \overline{s_1} \cdot s_0 + P_2 \cdot s_1 \cdot \overline{s_0} + P_3 \cdot s_1 \cdot s_0$

2.5 Transmission Gate Circuits

- TG-based XOR and XNOR

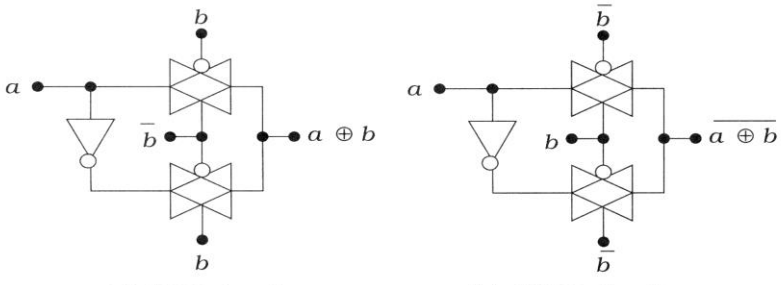


Figure 2.62 TG-based exclusive-OR and exclusive-NOR circuits

2.5 Transmission Gate Circuits

- TG-based OR
  - $f = a \cdot \bar{\bar{a}} + \bar{a} \cdot b = a + \bar{a} \cdot b = a + b$

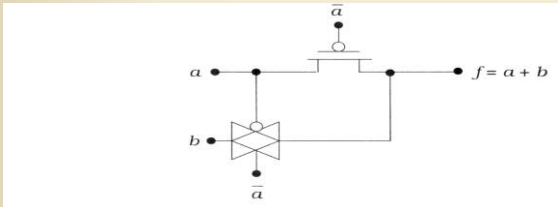


Figure 2.63 A TG-based OR gate

2.5 Transmission Gate Circuits

- Alternate XOR/XNOR circuits
  - Importance in designing error detection/correction functions

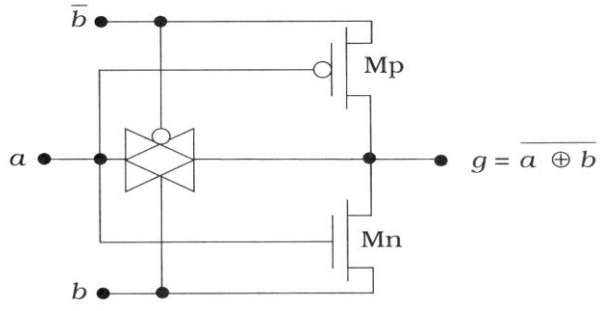


Figure 2.64 An XNOR gate that uses both TGs and FETs

2.6 Clocking and Dataflow Control

- Frequency and period
  - $f = \frac{1}{T}$
- Applying the complementary clocks to transmission gate

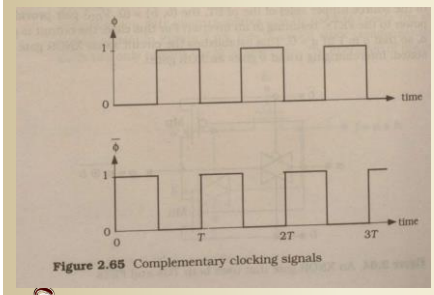


Figure 2.65 Complementary clocking signals

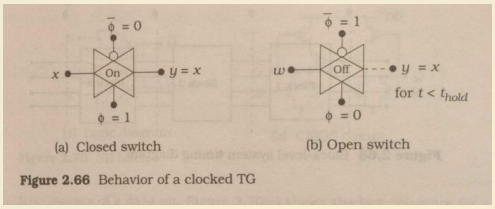


Figure 2.66 Behavior of a clocked TG

2.6 Clocking and Dataflow Control

- Data synchronization

clock長度不能大於Thold

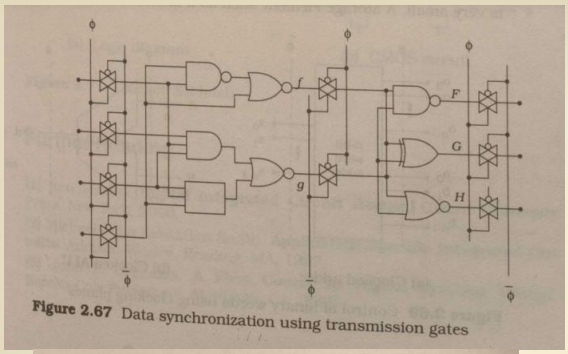


Figure 2.67 Data synchronization using transmission gates

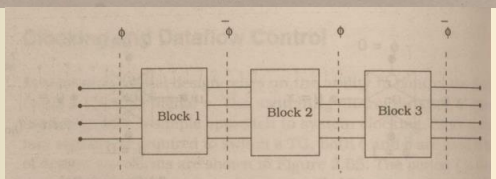
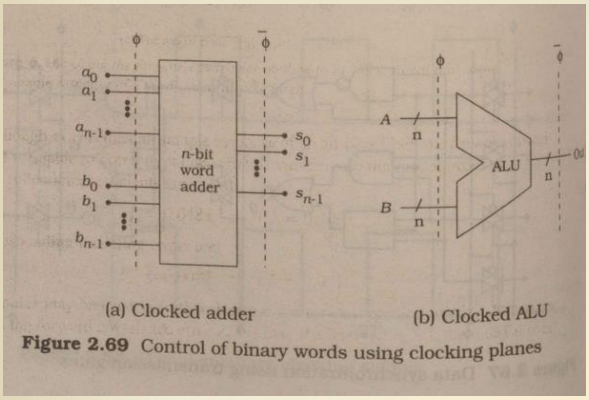


Figure 2.68 Block-level system timing diagram

2.6 Clocking and Dataflow Control

- Data synchronization

可以在positive sig送值
negative sig取值



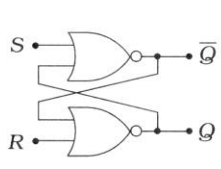
(a) Clocked adder (b) Clocked ALU

Figure 2.69 Control of binary words using clocking planes

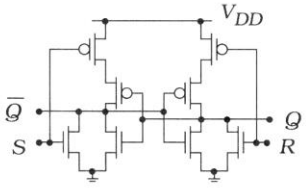
2.6 Clocking and Dataflow Control

- Clocked transmission gates synchronize the flow of signals, but the lines themselves cannot store the values for times longer than t_{hold}, which is very small
- A storage element such as a latch is needed
- SR Latch

1:1沒work



(a) Logic diagram



(b) CMOS circuit

Figure 2.70 SR latch

2.6 Clocking and Dataflow Control

- Clock control can be added to the circuit by inserting AND gates at the inputs to arrive at the modified logic diagram in Figure 2.71(a)
- This only allows changes in the inputs when ϕ = 1 (clock)
- Clocked SR Latch

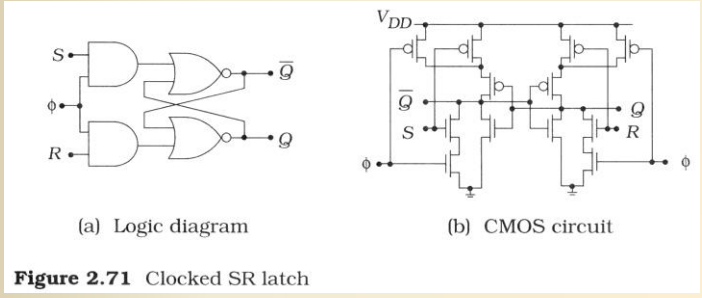
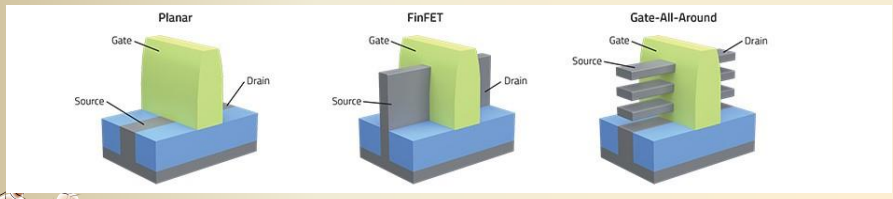


Figure 2.71 Clocked SR latch

Evolution of FET Structure

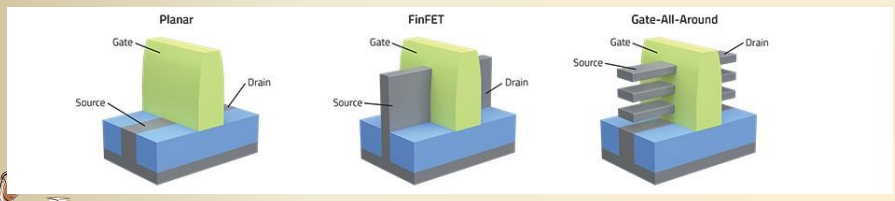
- FinFET
  - Fin field-effect transistor
  - A multigate device, a MOSFET built on a substrate where the gate is placed on two, three, or four sides of the channel or wrapped around the channel, forming a double or even multi gate structure
  - The source/drain region forms fins on the silicon surface
  - The FinFET devices have significantly faster switching times and higher current density than planar CMOS



Source: Lam Research

Evolution of FET Structure

- Gate-All-Around
  - Gate-all-around FETs (GAAFETs) are the successor to FinFETs, as they can work at sizes below 7 nm
  - Similar in concept to a FinFET except that the gate material surrounds the channel region on all sides
  - Depending on design, gate-all-around FETs can have two or four effective gates
  - GAAFETs with up to 7 nanosheets have been demonstrated which allow for improved performance and/or reduced device footprint



Source: Lam Research

Q&A

Andy, Yu-Guang Chen
Asscoiate Professor, Department of EE, NCU
Email: andyygchen@ee.ncu.edu.tw
FB: Yu-Guang Chen
IG: ncu.eda.andy
Google account: andyygchen.ncu