

CHAPTER2: Logic Elements and Electronic Signal on Silicon

2.1. Digital Logic and CMOS Circuit.

Devices Implemented in LSI

○ MOS Transistor

Broadly used owing to the properties of high-speed, low-voltage, and high-integration.



○ Bipolar Transistor

Used in RF and analog applications owing to its high-drive-ability.



○ Other devices

Diode, resistors, capacitors etc.



MOS: Metal Oxide Semiconductor

What is Semiconductor?

A semiconductor is a material that behaves between a conductor and an insulator.

At room temperature, semiconductor has higher electric conductivity than an insulator, but lower than a conductor.

At very low temperatures, pure or intrinsic semiconductors behave like insulators.

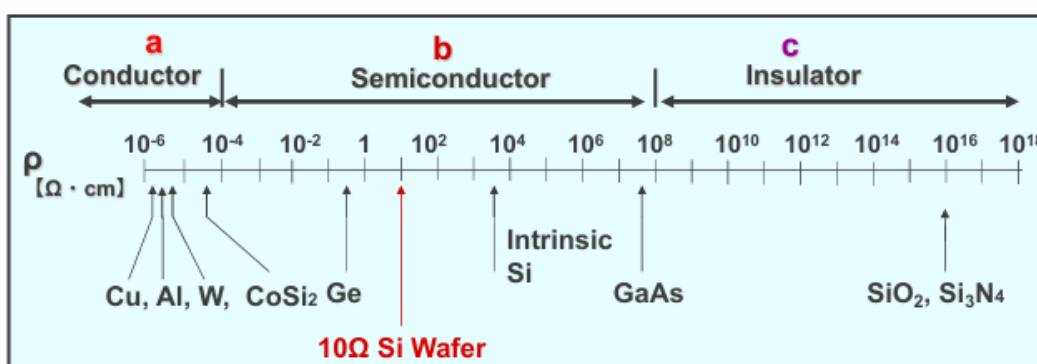
At higher temperatures or under light, pure or intrinsic semiconductors can become conductive.

The addition of impurities to a pure semiconductor can also increase its conductivity.

What is Semiconductor?

- Conductor (Cu, Al) electron free from atomic bound
- Semiconductor (Si, Ge, GaAs) electron loosely bounded to atom
- Insulator (SiO_2 , Si_3N_4) electron tightly bounded to atom

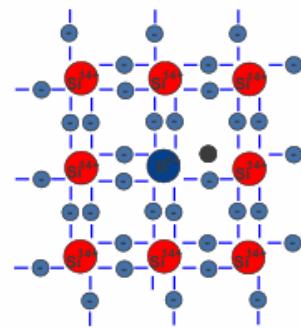
Materials are grouped by 3 types in electrical resistivity



P-type Silicon - Acceptor

It is very easy for an electron from a nearby **Silicon to Silicon** bond to fall into this hole and effectively move the hole away from the **Boron** atom

Since the **Boron** atom will accept an electron, **Boron** and the other elements of Group III (B, Ga) are referred to as **acceptors**



Silicon with acceptor is called as **P-type Silicon**, since “positive” holes are generated and contribute a current flow

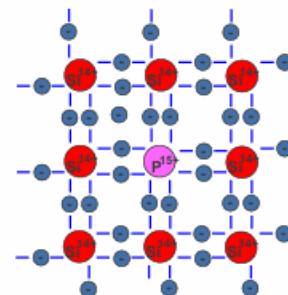
N-type Silicon - Donor

If a Group V atom, such as **Phosphorus**, is introduced into the **Silicon** lattice, it will have an extra electron which may easily break away, becoming a conduction electron

The **Phosphorus** is referred to a **donor**, since it donates an electron to the conduction band.

Other donor is **As**

Silicon with donor is called as **N-type Silicon**, since “negative” electrons are generated and contribute the current flow

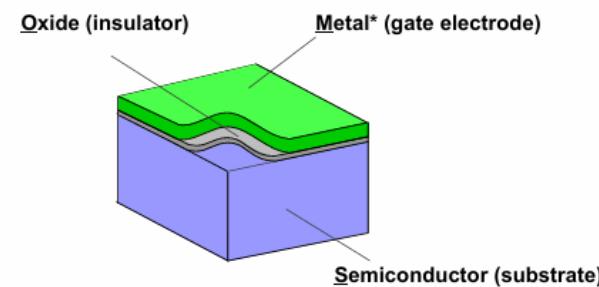


Properties of Silicon and Silicon Oxide

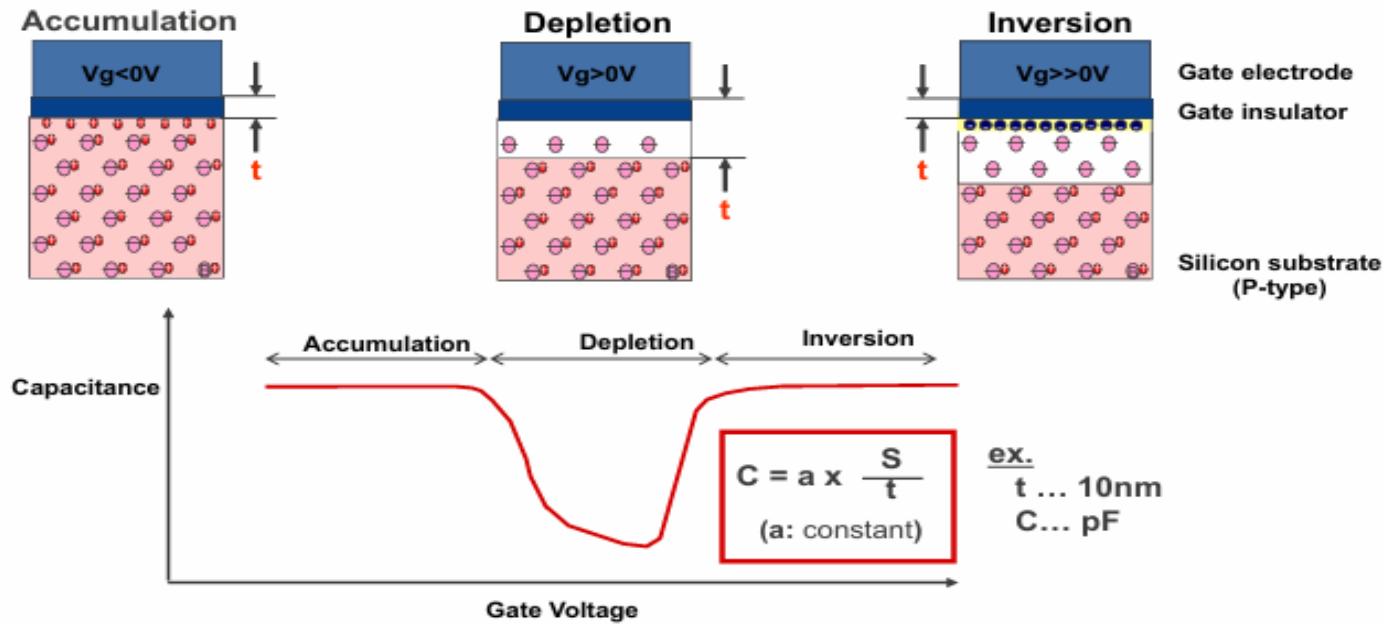
MOS Capacitor

TABLE 2.1 Physical Properties of Si and SiO₂ at Room Temperature (300 K)

Property	Si	SiO ₂
Atomic/molecular weight	28.09	60.08
Atoms or molecules/cm ³	5.0×10^{22}	2.3×10^{22}
Density (g/cm ³)	2.33	2.27
Crystal structure	Diamond	Amorphous
Lattice constant (Å)	5.43	—
Energy gap (eV)	<u>1.12</u>	8–9
Dielectric constant	<u>11.7</u>	<u>3.9</u>
Intrinsic carrier concentration (cm ⁻³)	<u>1.4×10^{10}</u>	—
Carrier mobility (cm ² /V-s)	Electron: 1430 Hole: 470	—
Effective density of states (cm ⁻³)	Conduction band, $N_c: 3.2 \times 10^{19}$ Valence band, $N_v: 1.8 \times 10^{19}$	—
Breakdown field (V/cm)	3×10^5	$>10^7$
Melting point (°C)	1415	1600–1700
Thermal conductivity (W/cm·°C)	1.5	0.014
Specific heat (J/g·°C)	0.7	1.0
Thermal diffusivity (cm ² /s)	0.9	0.006
Thermal expansion coefficient (°C ⁻¹)	2.5×10^{-6}	0.5×10^{-6}

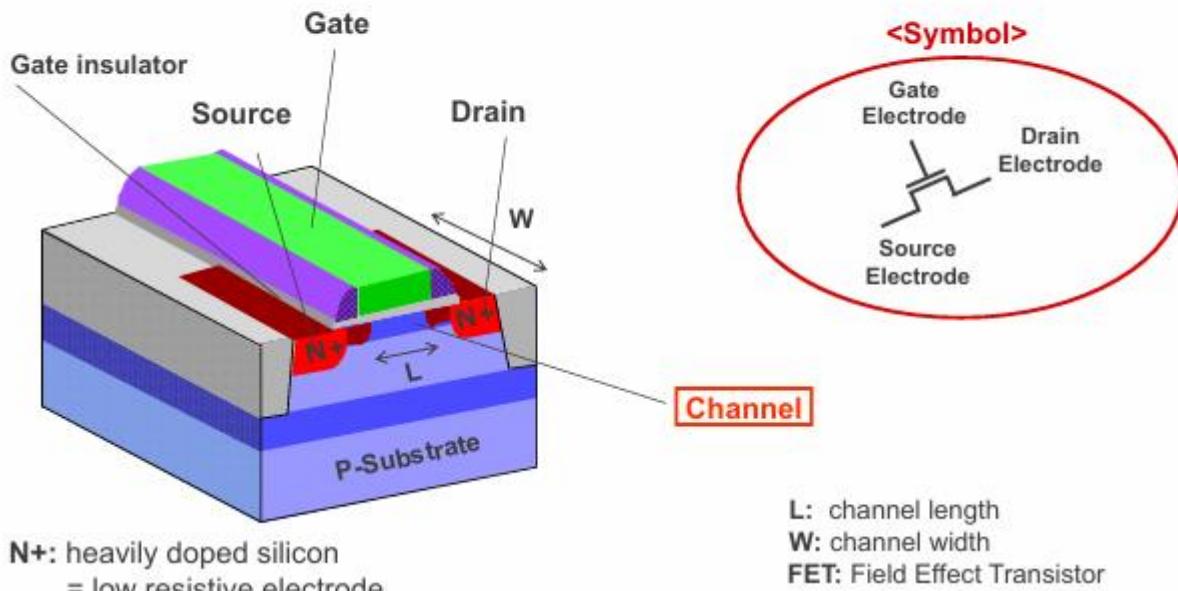


Behavior of MOS Capacitor

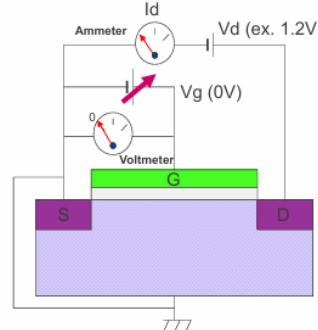
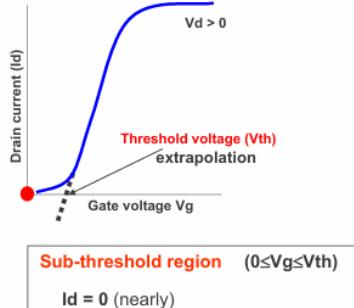


MOSFET

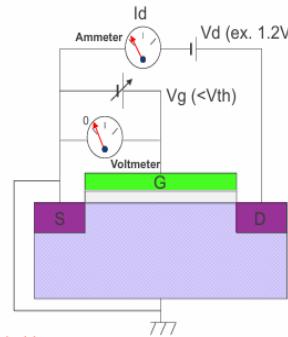
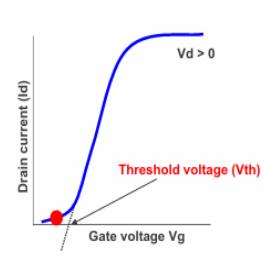
*** Focusing on n-channel transistor ***



- Gate characteristics



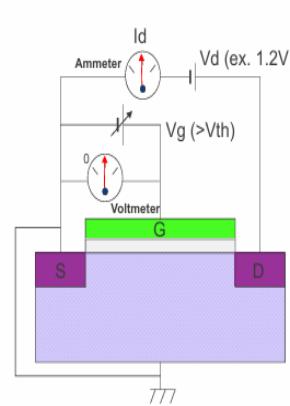
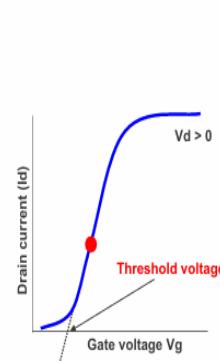
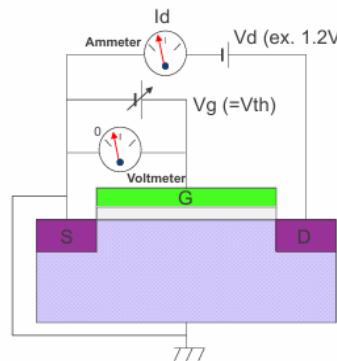
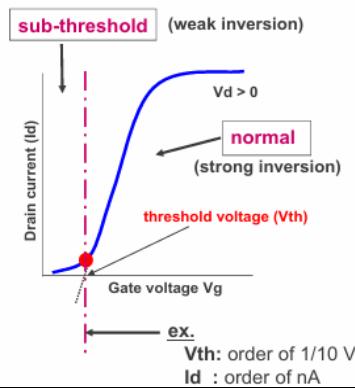
- Gate characteristics



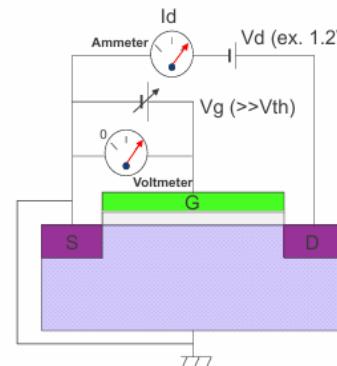
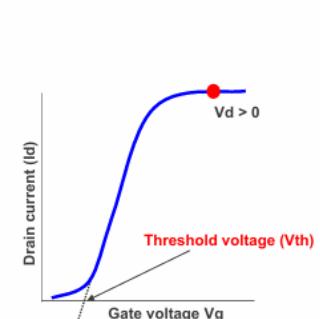
When $V_g < V_{th}$, the electric current is called **sub-threshold current**. This is one of the leakage currents.

Gate characteristics

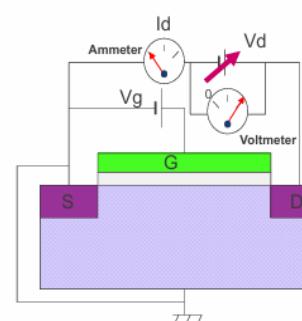
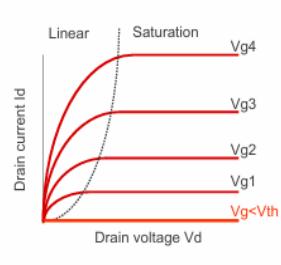
Gate characteristics



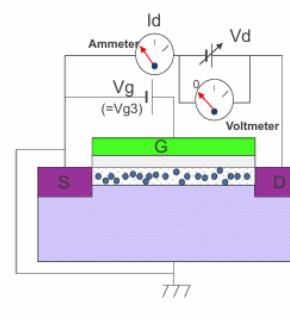
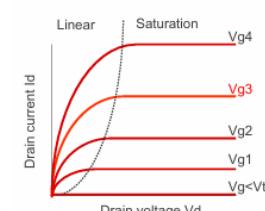
Gate characteristics



Drain characteristics



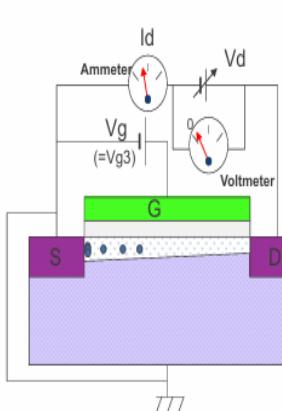
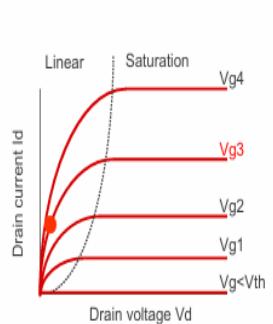
Drain characteristics



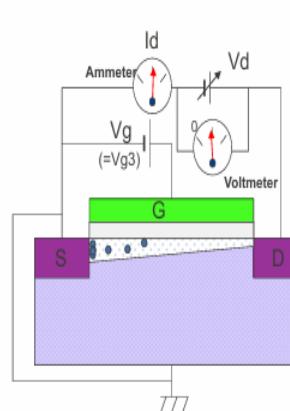
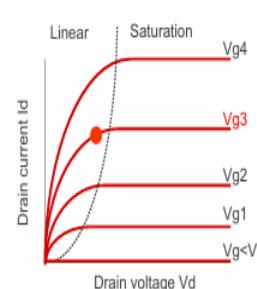
Apply $V_g = V_{g3}$, and watch what happens.

Even if $V_d > 0$, there is no current (except leakage) when $V_g < V_{th}$.

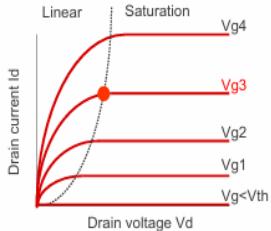
Drain characteristics



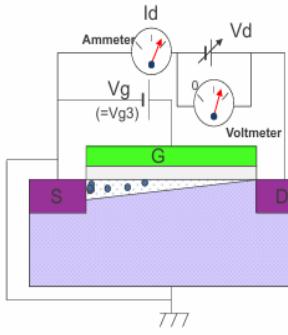
Drain characteristics



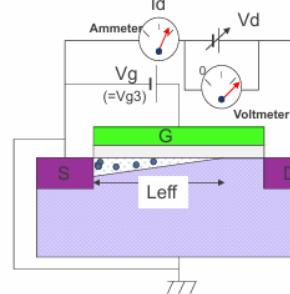
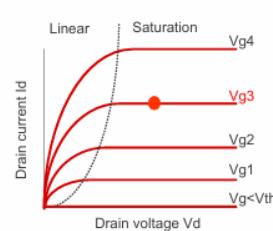
Drain characteristics



When $V_d = V_g - V_{th}$, it is called "pinch-off point".



Drain characteristics



After pinch-off, the current becomes constant.
Electrons are attracted by drain field and flow through the depletion region.

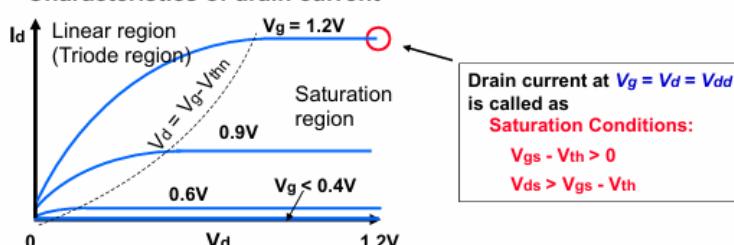
Current characteristics of MOSFET (Summary)

<Fundamental Formula of drain current>

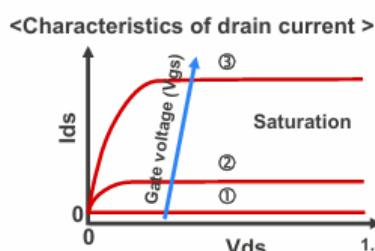
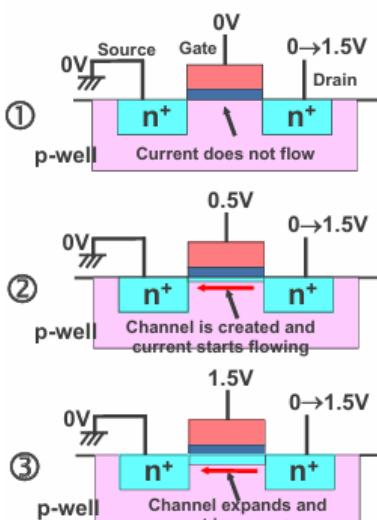
$$I_d = \begin{cases} \beta_n \{V_d(V_g - V_{thn}) - V_d^2/2\} & (\text{Linear region} \\ & 0 \leq V_d \leq V_g - V_{thn}) \\ \beta_n (V_g - V_{thn})^2 / 2 & (\text{Saturation region} \\ & V_d > V_g - V_{thn}) \end{cases}$$

$\beta_n = \mu C_{ox} W/L$
 μ : Electron Mobility
 C_{ox} : Gate capacitance per unit area
 W : Gate width
 L : Gate length
 V_{thn} : Threshold voltage of NMOS
(gate voltage required to switch ON transistor)

<Characteristics of drain current>



Operations of nMOS Transistor

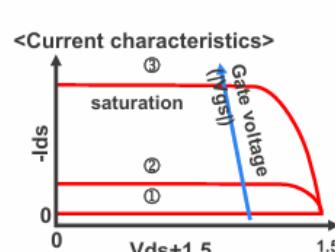
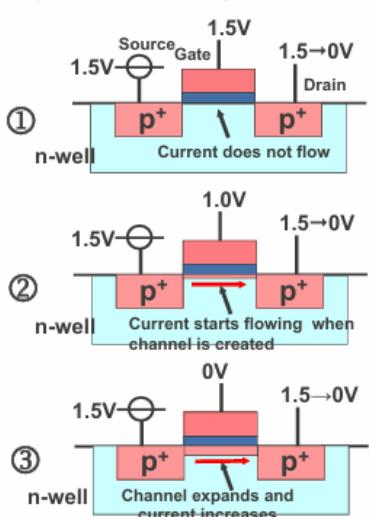


<Formula of drain current>

$$I_{ds} = \begin{cases} \beta_n \{V_d(V_g - V_{thn}) - V_d^2/2\} & (\text{when } 0 \leq V_d \leq V_g - V_{thn}) \\ \beta_n (V_{gs} - V_{thn})^2 / 2 & (\text{when } V_d > V_g - V_{thn}) \end{cases}$$

β_n : proportionality coefficient
 V_{thn} : threshold voltage (gate voltage required to switch ON transistor)

Operations of pMOS Transistor

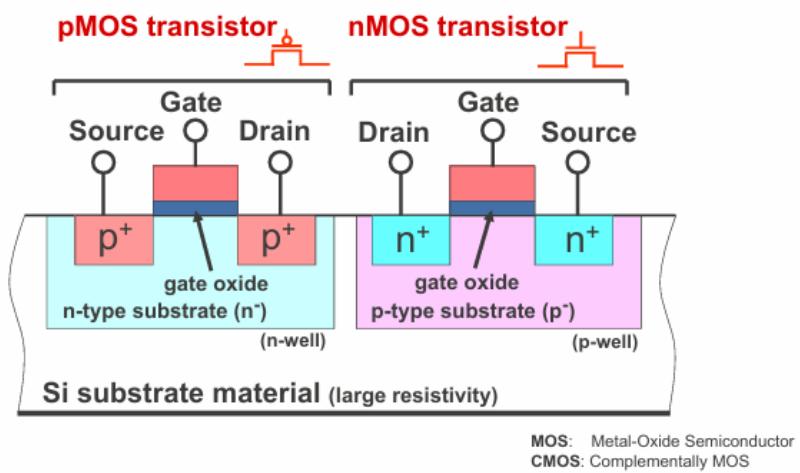


<Formula of current>

$$I_{ds} = \begin{cases} -\beta_p \{V_d(V_g - V_{thp}) - V_d^2/2\} & (\text{when } V_g - V_{thp} \leq V_d \leq 0) \\ -\beta_p (V_{gs} - V_{thp})^2 / 2 & (\text{when } V_{ds} + 1.5 > V_{gs} - V_{thp}) \end{cases}$$

β_p : proportionality coefficient
 V_{thp} : threshold voltage (gate voltage required to switch ON transistor)

Structure of CMOS Transistor

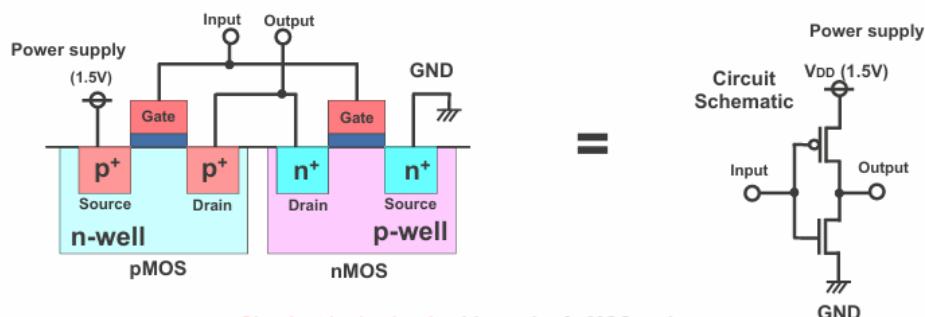


MOS: Metal-Oxide Semiconductor
CMOS: Complementary MOS

2.2. Logic Gate on Silicon.

Inverter

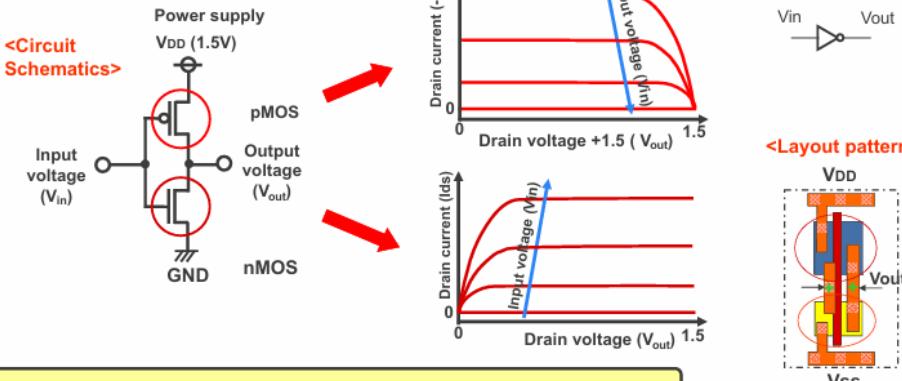
<CMOS inverter Structure >



Simplest logic circuit with a pair of pMOS and nMOS transistors

Logic Circuit on Silicon Chip

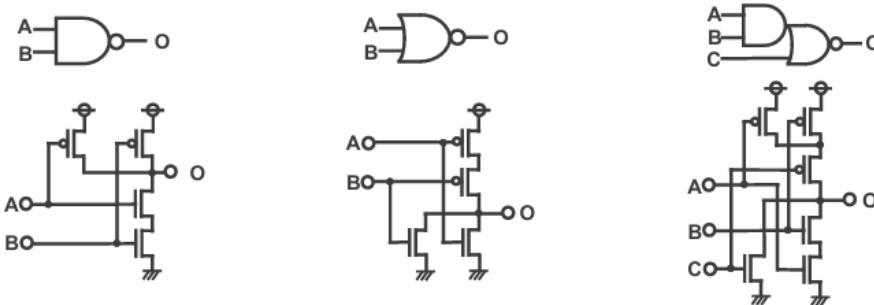
Inverter



Inverter characteristics are defined by current characteristics of both transistors.

Logic Gate

- 2 input NAND ($O = A \cdot B$)
- 2 input NOR ($O = A + B$)
- Combinational gate ($O = A \cdot B + C$)



With complementary transistor configurations, all logic circuits can be implemented.
Basically their complementary operations are similar to those of an inverter.

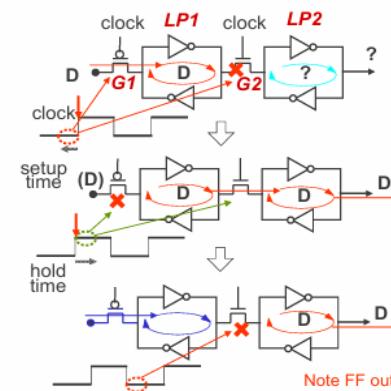
Nguyên lý hoạt động của Flip Flop

1. Cấu trúc cơ bản

- **Flip Flop bao gồm hai mạch vòng lưu trữ (Loop):**
 - LP1: lưu dữ liệu tạm thời khi clock = 0.
 - LP2: lưu dữ liệu ổn định khi clock = 1.
- **Hai cổng điều khiển G1 và G2:**
 - G1: hoạt động khi clock = 0 (cho phép nạp dữ liệu D vào LP1).
 - G2: hoạt động khi clock = 1 (chuyển dữ liệu từ LP1 sang LP2).

Flip Flops

Principle of operation



1. Gate G1 is ON while clock remains low, and data D is taken into loop LP1.

2. When clock rises, G1 is OFF and G2 is ON to pass data D into next loop LP2.

3. When clock goes low, G2 is OFF and LP2 keeps its data D.

Note FF output is not determined until any value is set from outside from logic simulation standpoint. Especially care should be taken after power on.

2. Quá trình hoạt động từng giai đoạn

Giai đoạn 1: Khi clock = 0

- G1 bật → Dữ liệu D được nạp vào vòng lặp LP1.
- G2 tắt, LP2 giữ nguyên giá trị cũ (không thay đổi).

Giai đoạn 2: Khi clock chuyển từ 0 → 1 (cạnh lên)

- G1 tắt, chặn D mới.
- G2 bật, cho phép truyền dữ liệu từ LP1 sang LP2.
- Dữ liệu D từ LP1 được cập nhật vào LP2 (ra output).

Giai đoạn 3: Khi clock = 1 → 0

- G2 tắt → giữ dữ liệu ổn định trong LP2.
- LP2 lưu giữ giá trị D đã nhận được từ lần clock lên trước đó.

Các thời điểm quan trọng

- **Setup time:** Thời gian tối thiểu dữ liệu D phải ổn định **trước khi clock chuyển từ 0 → 1** (cạnh lên).
- **Hold time:** Thời gian tối thiểu dữ liệu D cần tiếp tục giữ ổn định **sau khi clock chuyển từ 0 → 1**.
- Nếu vi phạm **setup** hoặc **hold time**, Flip Flop có thể hoạt động sai, dẫn đến **dữ liệu không xác định**.

Ghi chú quan trọng

- Ban đầu sau khi cấp nguồn, **giá trị FF output chưa xác định** cho đến khi dữ liệu D được nạp vào từ ngoài.
- Khi mô phỏng hoặc thiết kế mạch logic, cần đặc biệt chú ý quá trình **reset** hoặc **khởi tạo ban đầu**.

Metastable State (Metastability)

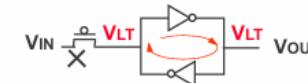
1. Vấn đề lý thuyết

- **Ngưỡng điện áp logic (V_LT):**
 - Là mức điện áp ngưỡng để phân biệt **logic 0** và **logic 1**.
 - Thường đặt $V_{LT} = \frac{1}{2} V_{DD}$ để tối ưu **biên độ nhiễu an toàn (noise margin)**.
- **Vấn đề với Flip Flop:** Khi tín hiệu đầu vào (V_{IN}) ở ngay tại mức ngưỡng V_{LT} , nếu cổng bị tắt (OFF), Flip Flop có thể giữ **trạng thái không xác định** trong một khoảng thời gian **không dự đoán trước**.

Flip Flops

Metastable State (Metastability)

- To set logical threshold voltage (V_{LT}) of each GATE at the same level is important to secure noise margin: $V_{LT} = 1/2 V_{DD}$.
- But big problem in FLIP FLOPs.



- When input V_{IN} is held at V_{LT} and then gate becomes OFF, the flip flop MIGHT keep this level for unpredictable period.
- But actually when small plus noise is applied to left node of loop, V_{OUT} accordingly becomes 0. In minus case, V_{OUT} goes to 1.
- Behavior of V_{OUT} is **not predictable** depending upon noise level: **metastable**.

< hypothetical case >

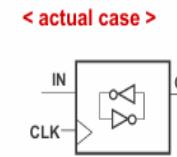
2. Cơ chế xảy ra Metastability

- Nếu $V_{IN} = V_{LT}$ ngay lúc gate tắt:
 - Flip Flop **không thể quyết định** ra logic 0 hay 1 ngay lập tức.
 - Output V_{OUT} sẽ "lơ lửng" ở mức không xác định trong khoảng thời gian bất kỳ.
- Tuy nhiên, **một nhiễu nhỏ** (noise) xuất hiện:
 - Nếu nhiễu dương (+): Flip Flop ngả về **logic 0**.
 - Nếu nhiễu âm (-): Flip Flop ngả về **logic 1**.

Nguyên nhân gây metastability

- Khi cạnh xuống của IN (tín hiệu dữ liệu) và cạnh lên của CLK (clock) xảy ra gần nhau (rất sát về thời gian).
- Khi đó:**
 - Điện áp tại vòng phản hồi (inverter loop) bên trong Flip Flop rơi vào vùng gần bằng V_{LT} (mức ngưỡng logic).
 - Hành vi kế tiếp (output ổn định về 0 hay 1) trở nên không dự đoán được.

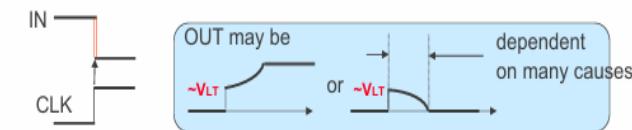
Metastable State (Metastability)



< actual case >

When falling edge of IN and rising edge of CLK are very close, voltages at inverter loop within FF become close to V_{LT} depending on timing of IN.

Succeeding behavior is unpredictable.



→ This phenomenon surely happens when IN is asynchronous with CLK. OUT must be sensed at least one clock cycle after CLK rising edge under focusing.

2. Mô tả hiện tượng Khi IN và CLK thay đổi quá gần nhau:

- Output (OUT) có thể:
 - Chậm trễ ổn định về mức thấp (~0).
 - Hoặc chậm trễ ổn định về mức cao (~1).
- Ngoài ra, thời gian giữ ở trạng thái không ổn định phụ thuộc vào nhiều yếu tố: nhiễu, sức mạnh tín hiệu, nhiệt độ, v.v.

3. Nhận xét quan trọng: Hiện tượng này chắc chắn sẽ xảy ra khi tín hiệu IN không đồng bộ với CLK. Do đó:

- Không nên đọc OUT ngay lập tức sau cạnh lên của CLK.
- Cần chờ ít nhất 1 chu kỳ clock để đảm bảo tín hiệu OUT đã ổn định hoàn toàn (hạn chế bị ảnh hưởng bởi metastability).

SoC Design and Jigsaw Puzzle

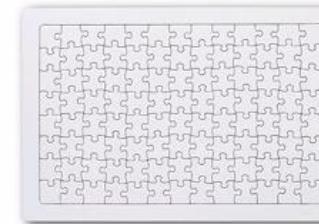
- Jigsaw Puzzle:** Là trò chơi lắp ráp nhiều mảnh nhỏ vào nhau để tạo thành một bức tranh hoàn chỉnh.
- SoC Design (Thiết kế vi mạch SoC):** Là quá trình lắp ráp toàn bộ con chip từ nhiều thành phần (cells) lấy ra từ các thư viện mạch. Các cell được đặt vị trí (placement) và kết nối dây (routing) sao cho hoạt động hài hòa và đúng chức năng yêu cầu.

→ Các loại thư viện mạch chính cần thiết cho thiết kế SoC:

- Logic Cell Library
- Memory Library
- Analog Circuit Library

SoC Design and Jigsaw Puzzle

- Jigsaw puzzle:** assemble whole picture from many of interlocking pieces.
- SoC design:** assemble whole chip from components (cells) in circuits libraries, place and route them, so that they function harmoniously.



1. Logic Cell library
2. Memory library
3. Analog Circuit library

Page 36

Circuit libraries on SoC

1. Logic cell library
 - Primitive cells (Inverter, Buffer, NAND, NOR, FF, etc.)
 - Data Path (Execution unit, selector, multiple-bit width)
 - Clock Buffer
 - Power Control Circuits (Power Switch, Substrate Bias Controller)
2. Memory library
 - Register File
 - RAM (Random Access Memory)
 - ROM (Read Only Memory)
3. Analog Circuit library
 - IO (Input and Output Buffer, Level Shifter)
 - PLL (Phased Locked Loop)
 - ADC (Analog to Digital Converter)
 - RF (Radio Frequency Circuit)
 - PA (Power Amplifier)

Circuit Libraries trên SoC (Chi tiết)

1. Logic Cell Library

- Chứa các cell cơ bản để xây dựng logic số:**
 - Primitive cells:** Inverter, Buffer, NAND, NOR, Flip-Flop (FF), ...
 - Data Path Elements:** Các đơn vị xử lý (Execution Unit), bộ chọn (Selector), xử lý dữ liệu nhiều bit (multiple-bit width).
 - Clock Buffer:** Các khối đệm xung clock nhằm đảm bảo tín hiệu clock mạnh và đồng đều.
 - Power Control Circuits:** Mạch điều khiển năng lượng như Power Switch, Substrate Bias Controller.

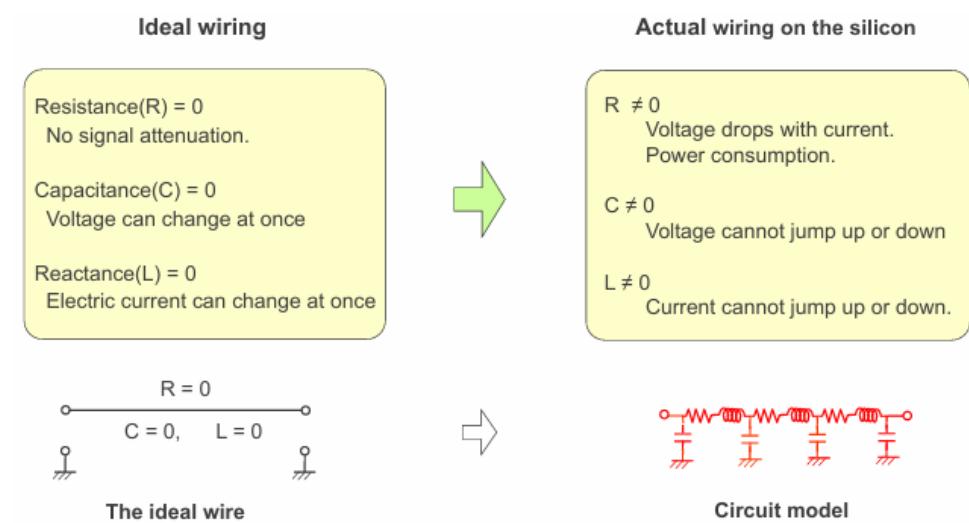
2. Memory Library: Cung cấp bộ nhớ tích hợp:

- Register File: tập hợp các thanh ghi lưu trữ tạm thời.
- RAM (Random Access Memory): bộ nhớ truy cập ngẫu nhiên, đọc/ghi nhanh.
- ROM (Read Only Memory): bộ nhớ chỉ đọc, dùng lưu trữ cố định chương trình hay dữ liệu.

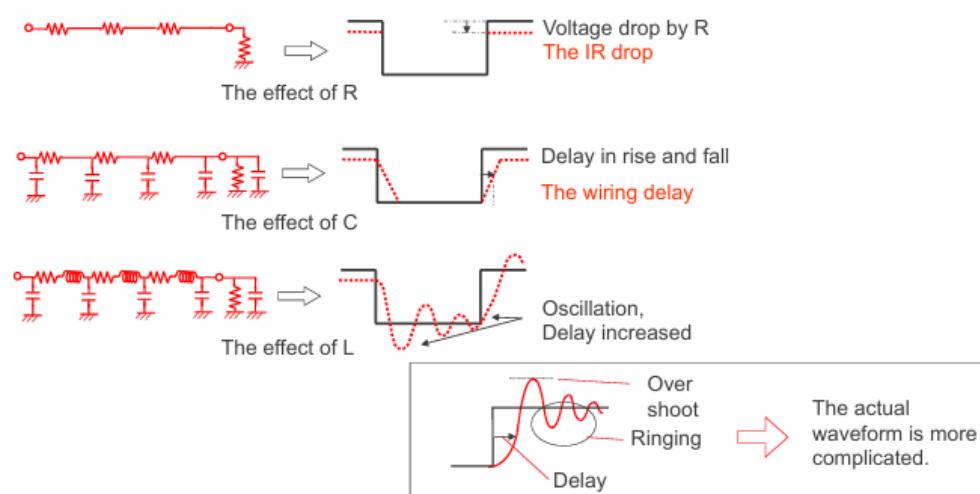
3. Analog Circuit Library: Các cell liên quan đến tín hiệu tương tự (analog):

- IO Buffers: Bộ đệm đầu vào/ra (Input and Output Buffer), bộ chuyển mức điện áp (Level Shifter).
- PLL (Phased Locked Loop): Mạch đồng bộ tần số và pha tín hiệu clock.
- ADC (Analog to Digital Converter): Bộ chuyển đổi tín hiệu analog thành digital.
- RF Circuit (Radio Frequency Circuit): Mạch RF cho giao tiếp không dây.
- PA (Power Amplifier): Bộ khuếch đại công suất, đặc biệt cho ứng dụng RF.

2.3. Electronic Signal Propagation on Silicon.



Effect of R, C and L



Delay Definitions (Định nghĩa các loại trễ trong mạch logic)

1. Logic delay (Trễ logic tổng quát)

• Định nghĩa:

- Trễ logic qua một cổng được mô tả bởi thời gian trễ lan truyền (t_p).
- Đây là thời gian trung bình cần thiết để đầu ra phản ứng với thay đổi trạng thái logic ở đầu vào.

• Công thức: $t_p = 1/2 (t_{pHL} + t_{pLH})$

- t_{pHL} : Falling propagation delay.
- t_{pLH} : Rising propagation delay.

2. Falling Propagation Delay (t_{pHL}): Thời gian đầu ra giảm từ mức cao ("1") xuống mức thấp ("0") sau khi đầu vào thay đổi 50% mức VDD.

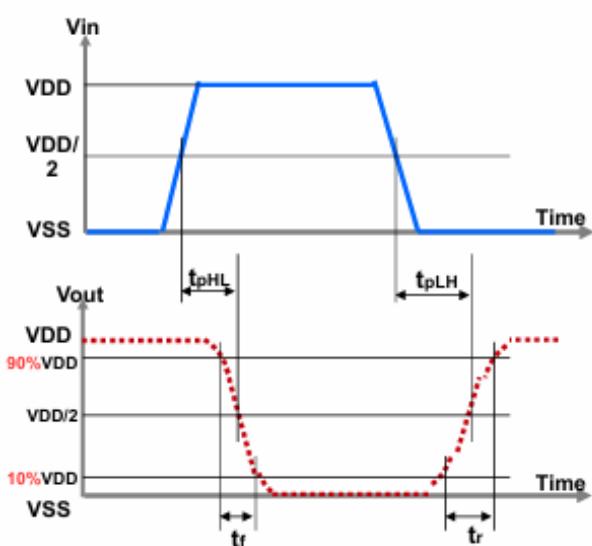
3. Rising Propagation Delay (t_{pLH}): Thời gian đầu ra tăng từ mức thấp ("0") lên mức cao ("1") sau khi đầu vào thay đổi 50% mức VDD.

4. Fall time (t_f): Thời gian đầu ra rời từ mức logic "1" về mức logic "0".

5. Rise time (t_r): Thời gian đầu ra tăng từ mức logic "0" lên mức logic "1".

6. Giải thích thêm về Logical Levels

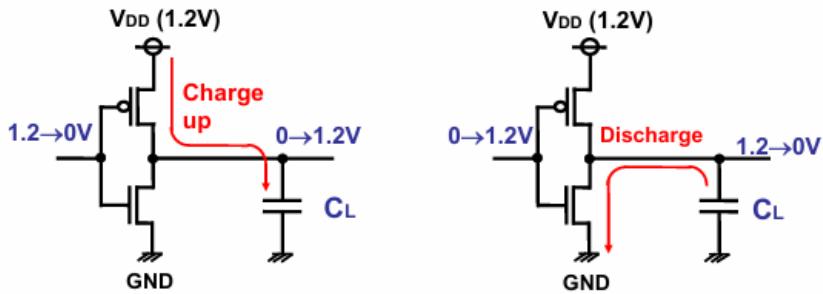
- Mức logic "0": VSS đến 10% VDD.
- Mức logic "1": Từ 90% VDD đến VDD.



Logical level:

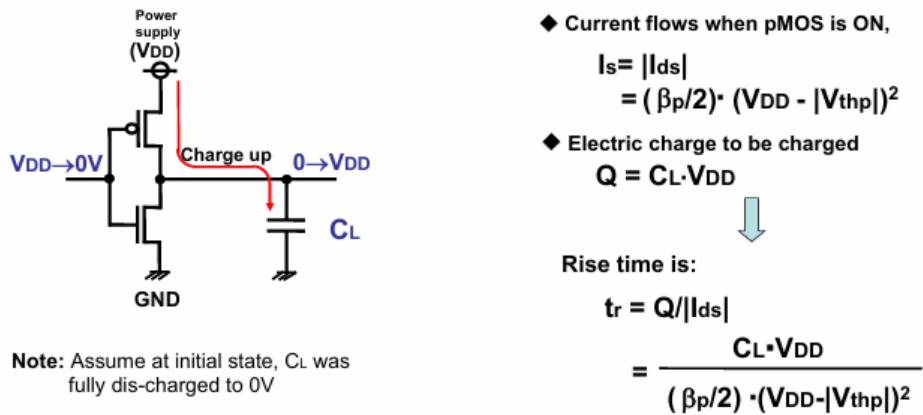
- Level "0": from VSS to 10% of VDD;
- Level "1": from 90% of VDD to VDD.

Charge up & Discharge

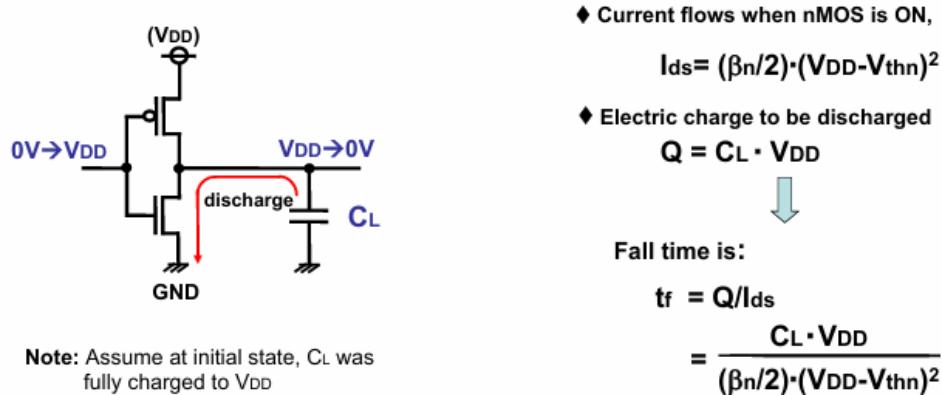


Rise time and fall time: times required to charge or discharge the load capacitor.
Large load capacitance results in large delay.

Rise time - Delay of charge up



Fall time - Delay for discharge



Summary of delay time

$$\text{Delay time} = \begin{cases} \frac{C_L \cdot V_{DD}}{(\beta_p/2) \cdot (V_{DD} - |V_{thp}|)^2} & : \text{rise time} \\ \frac{C_L \cdot V_{DD}}{(\beta_n/2) \cdot (V_{DD} - V_{thn})^2} & : \text{fall time} \end{cases}$$

$$\beta = \mu * C_{ox} * W/L$$

→

◆ Faster, if gate length L is shorter
◆ Faster, if gate width W is wider

Mobility Capacity of gate oxide per unit area

2.4. Problem with Electronic Signal

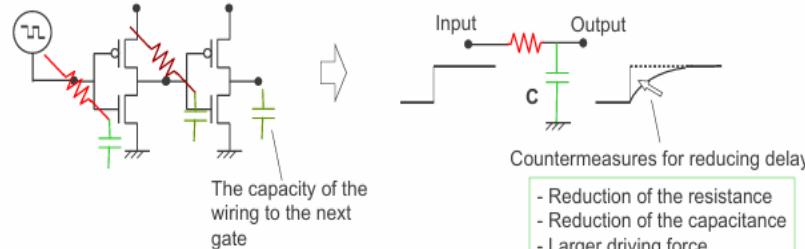
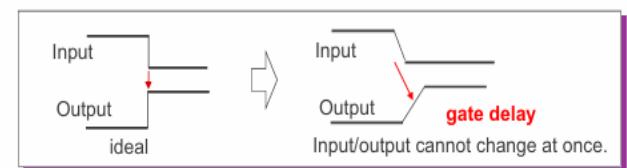
Gate Delay & Wiring Delay

Gate Delay & Wiring Delay

1. Gate Delay (Trễ cổng)

- Ideal case (trường hợp lý tưởng):** Input thay đổi → Output cũng thay đổi ngay lập tức (không có trễ).
- Thực tế:**
 - Khi input thay đổi, output không thể thay đổi ngay lập tức.
 - Xuất hiện trễ cổng (gate delay).

Gate delay = Thời gian xử lý bên trong cổng logic (ví dụ: NAND, NOR...) khi đầu vào thay đổi.



2. Wiring Delay (Trễ dây nối)

- Wiring delay đến từ:**
 - Điện trở (R) của dây nối.
 - Điện dung (C) do đường dây tạo ra (cộng với điện dung của gate tiếp theo).
- Dây nối và cổng tiếp theo tạo thành một mạch RC, làm chậm tín hiệu.
- Mô hình:** Input → R (resistor) → tụ điện C → Output.

Hiệu ứng: Làm slope (biên độ thay đổi) của tín hiệu chậm lại (tăng thời gian rise time, fall time).

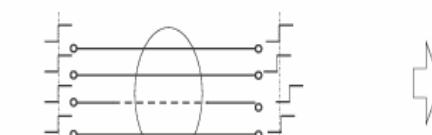
3. Biện pháp giảm delay

- Giảm điện trở của dây nối.
- Giảm điện dung của dây (ví dụ rút ngắn chiều dài dây, giảm kích thước dây).
- Tăng lực lái (driving force) của cổng logic (dùng buffer mạnh hơn hoặc transistor lớn hơn)

Skew (Độ lệch tín hiệu)

Skew

1. Skew là gì? Khi các tín hiệu cùng được phát đi cùng lúc, nhưng đến nơi không cùng lúc do: Khác nhau về điện trở (R), điện dung (C) và điện cảm (L) trên từng đường dây.

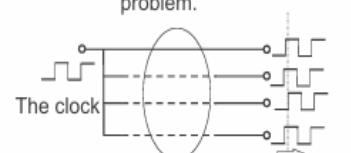


skew

For more-than-1-bit signal, skew is always a big problem.

These lines have different electrical (R, C, L) characteristics.

Clock signal will always suffer from skew problem.

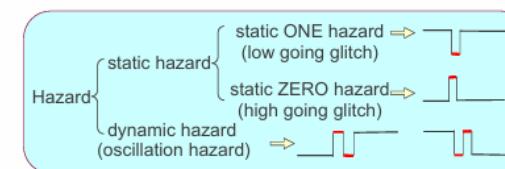
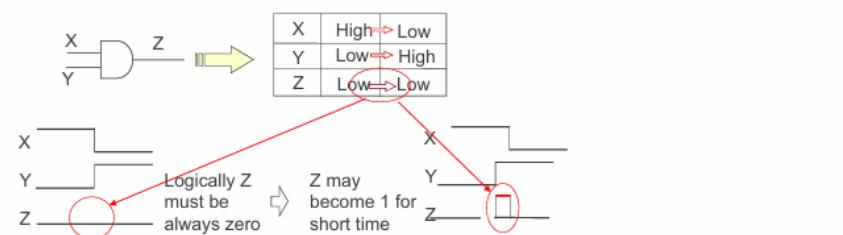


The clock skew

Skew causes difficulty in wiring clock lines, especially in higher frequency range and limits the clock speed.

Hazard & Glitch

Hazard is a problem caused by a small timing difference among several signals.



Hazard & Glitch

1. Hazard là gì?

- Là vấn đề xảy ra do **sự chênh lệch nhỏ về thời gian** giữa các tín hiệu đầu vào.
- Dẫn đến đầu ra **bị sai tạm thời**, dù về mặt logic thì không nên sai.

2. Ví dụ đơn giản

- Với 2 tín hiệu X và Y vào một cổng logic,
 - Lý thuyết: đầu ra Z **luôn phải là 0**,
 - Nhưng thực tế: Z **bị nhảy lên 1 trong chốc lát** vì tín hiệu X và Y **không thay đổi cùng lúc**.
- ➡ Gây ra **Glitch** (nhiều tạm thời).

3. Phân loại Hazard

Loại Hazard	Ý nghĩa
Static hazard	Đầu ra lẽ ra không thay đổi, nhưng bị nhấp nháy
- Static ONE hazard	Từ mức 1, glitch thành 0 ngắn
- Static ZERO hazard	Từ mức 0, glitch thành 1 ngắn
Dynamic hazard (Oscillation hazard)	Tín hiệu dao động nhiều lần trước khi ổn định

Cách tránh Hazard

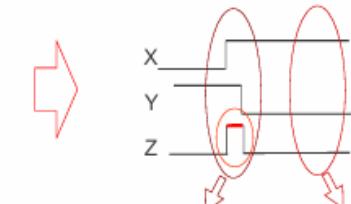
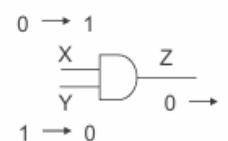
1. Nguyên lý tránh Hazard

- Hazard gây ra **output sai tạm thời** ngay sau khi input thay đổi.
- Nếu chờ đủ thời gian (cho tín hiệu ổn định), **output sẽ đúng**.
- Vì vậy, chỉ lấy tín hiệu sau một khoảng thời gian delay ổn định.

Quy tắc: Chờ tín hiệu đầu vào ổn định rồi mới lấy output từ logic tổ hợp.

How to avoid hazard?

It is difficult to prevent hazard. However, we can avoid using hazard signal by selecting a timing to use the signal.



After enough time passed, the output must be true.

2. Hazard đặc biệt nguy hiểm với tín hiệu cạnh (edge signal)

- Ví dụ: Tín hiệu **CLK** (Clock) rất nhạy cảm với glitch.
- Nếu có hazard trên đường clock → có thể **hở sớm 1 chu kỳ**.
- Hậu quả: **Hệ thống đồng bộ bị sai lệch**, lỗi rất khó sửa.

To avoid hazard, outputs of combinational logic must be used after proper time period passed since input signals become stable.

When handling edge signal, you must be very careful about hazard, because **edge signal is very sensitive to hazard**. Especially, clock signal is very sensitive to hazard.



This cause a big problem. Why??

3. Cách tránh hazard trên Clock

- X Không được chèn bất kỳ logic tổ hợp nào vào đường clock.**
- Thiết kế đồng bộ hoàn toàn theo clock chuẩn.

Quy tắc vàng: Đường clock chỉ nên đi thẳng, không có logic tổ hợp xen vào.

Hazard cause big problems resulting in malfunction of a system. Synchronous design may help us to get out of these problems.

To avoid hazard on clock signal, do not insert any combinational logic in clock line.

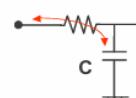
Power Consumption (Tiêu thụ điện năng)

1. Công thức Điện năng tiêu thụ động của CMOS:

$$P \sim f \times C \times V^2$$

Trong đó:

- fff = Tần số hoạt động
- CCC = Điện dung tải
- VVV = Điện áp hoạt động



$$\begin{aligned} \text{Electric power} &\sim V \cdot I \\ &\sim V \cdot Q / T \\ &\sim C \cdot V^2 / T \\ &\sim f \cdot C \cdot V^2 \end{aligned}$$

The power consumption of CMOS

$$P \sim f \cdot C \cdot V^2$$

Dynamic power

The keys to reduce power consumption.

- <1> Making operation voltage low
- <2> Lowering an operating frequency
- <3> Reducing capacitance and so on.

Power consumption limits the operating speed. Large power consumption makes it difficult to design packages.

However, the **leakage current is becoming significant**, we have to apply partial power off strategy to reduce power significantly.

2. Cách giảm tiêu thụ điện năng

Cách giảm	Ý nghĩa
1 Giảm điện áp hoạt động (V)	Rất hiệu quả do công suất tỉ lệ với V^2
2 Giảm tần số hoạt động (f)	Giảm công suất tuyến tính theo tần số
3 Giảm điện dung tải (C)	Tối ưu hóa layout, transistor nhỏ hơn

3. Vấn đề mới: Leakage Current

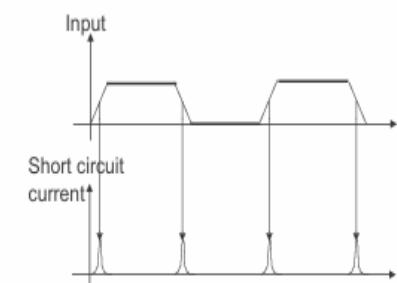
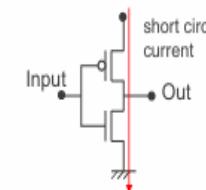
- Khi công nghệ nhỏ hơn (ví dụ 7nm, 5nm), **dòng rò (leakage current)** bắt đầu trở nên đáng kể → Dù không hoạt động, chip vẫn tiêu thụ điện!
- Giải pháp:** phải dùng các kỹ thuật **power gating** hoặc **partial power off** để cắt điện cho các khối không hoạt động.

Short Circuit Power (Công suất đoản mạch)

Short Circuit Power

1. Khái niệm

- Khi một **cổng CMOS** chuyển trạng thái (từ 0 → 1 hoặc từ 1 → 0), trong một khoảng thời gian rất ngắn, **cả transistor NMOS và PMOS đều dẫn (ON) cùng lúc**.
- Điều này tạo ra một dòng điện chạy trực tiếp từ nguồn (VDD) xuống mass (GND) → **gây ra dòng đoản mạch (short circuit current)**.



2. Mô hình mô tả

- Hình bên trái:** Input thay đổi → đồng thời NMOS và PMOS mở → xuất hiện dòng "short circuit current" từ VDD qua cả hai transistor đến GND.



Power consumption by short circuit current is about less than 10 to 15% of the total power consumption

- Hình bên phải (biểu đồ):**

- Input:** có dạng xung (vuông), biến thiên theo thời gian t.
- Short circuit current:** xuất hiện tại những thời điểm **chuyển mức** (tăng lên hoặc giảm xuống của input).
- Khi input ổn định ở mức cao hoặc thấp, dòng đoản mạch biến mất.

3. Đặc điểm dòng đoản mạch

- Chỉ tồn tại** trong **thời gian chuyển mạch** của tín hiệu đầu vào.
- Đặc biệt ngắn (vài picosecond tới nanosecond).
- Không liên tục:** chỉ xuất hiện mỗi lần tín hiệu vào thay đổi mức.

4. Ảnh hưởng đến công suất tiêu thụ

- Công suất tiêu thụ do dòng đoản mạch chiếm:
 - Khoảng 10% đến 15%** tổng công suất tiêu thụ của cổng CMOS.
- Ngoài công suất đoản mạch, tổng công suất tiêu thụ CMOS còn bao gồm:
 - Dynamic power** (công suất động do nạp/xả điện tích).
 - Leakage power** (công suất rò).

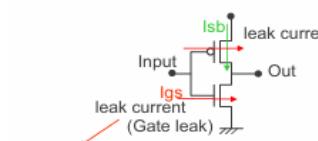
5. Kết luận

- Short circuit power** không phải thành phần chính, nhưng cũng **góp phần đáng kể** (~10-15%).
- Để **giảm tiêu thụ năng lượng**, cần:
 - Tối ưu thời gian chuyển mạch tín hiệu (giảm slope của input).
 - Thiết kế tối ưu transistor để giảm short-circuit current.

Leak Current (Dòng rò)

1. Khái niệm: Leak current là dòng điện **không mong muốn** rò rỉ qua các phần tử của transistor CMOS khi nó **không chuyển trạng thái** (tức là cả khi đang standby).

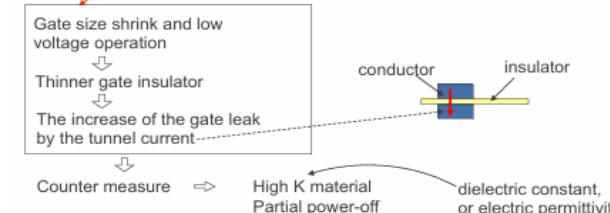
Leak Current



Devices with large leak current are not applicable to battery powered products.

2. Nguyên nhân gây ra leak current

- Thu nhỏ kích thước gate** (Gate size shrink) và **hoạt động điện áp thấp** (Low voltage operation):
 - Để đạt mật độ transistor cao hơn và giảm công suất động.
- Điều này dẫn đến:
 - Lớp cách điện gate** (Gate Insulator) phải mỏng hơn.
 - Khi lớp cách điện mỏng, **dòng tunnel** (tunnel current) qua lớp cách điện tăng lên → **gây leak current**.



3. Tác hại

- Các thiết bị có **leak current lớn** sẽ **không phù hợp** với sản phẩm dùng pin (battery-powered products).
- Vì leak current gây ra **mất năng lượng** ngay cả khi không hoạt động.

4. Hình minh họa (phía dưới slide)

- Mô phỏng dòng rò giữa hai conductor (dẫn điện) bị cách bởi một lớp insulator (cách điện).
- Nếu insulator quá mỏng hoặc không đủ tốt → dòng rò tăng.

5. Biện pháp khắc phục (Countermeasures)

- Sử dụng vật liệu High-K:**
 - Vật liệu có **hằng số điện môi (dielectric constant)** cao.
 - Cho phép **giữ điện dung cao** mà vẫn **giữ lớp cách điện dày hơn**, giảm tunnel current.
- Tắt một phần mạch** khi không dùng (Partial Power-off):
 - Cắt nguồn các phần tử không hoạt động để tránh dòng rò.

6. Kết luận

- Leak current** trở thành vấn đề nghiêm trọng ở công nghệ CMOS kích thước nhỏ (< 90 nm).
- Phải tối ưu thiết kế và vật liệu để duy trì hiệu suất năng lượng cao, đặc biệt cho các sản phẩm dùng pin như điện thoại, laptop...

Metastability of Flip-Flop

Metastability của Flip-Flop là gì?

- Khi **In** thay đổi **ngay tại thời điểm cạnh clock**, flip-flop **không kịp chốt giá trị**.
- Out** sẽ bị **không xác định (unstable)** trong một khoảng thời gian ngắn.
- Hiện tượng này gọi là **metastability**: không thể đoán trước ngay lập tức liệu output sẽ là 0 hay 1.

Ví dụ đơn giản: Nếu input **In** thay đổi đúng lúc clock trigger, **Out** có thể bị "lưỡng lự" (giữa 0 và 1) trong vài nanoseconds.

Đặc điểm metastability

- Out** sẽ:
 - Chậm ổn định về 0 hoặc 1 (delay).
 - Gây lỗi truyền dữ liệu giữa các module đồng hồ khác nhau (asynchronous clock domains).
- Thời gian metastable thường < 10 ns (nhưng có thể gây sai logic nặng trong hệ thống nhanh).

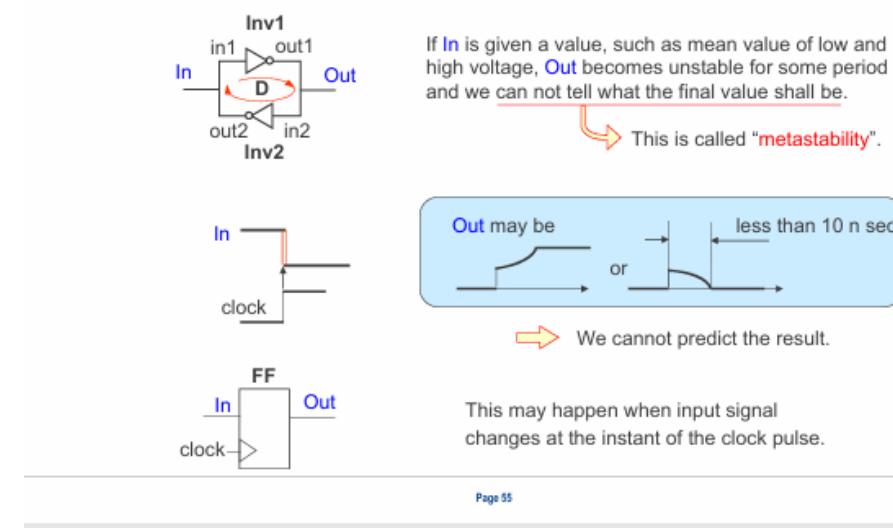
Cách tránh metastability

- Giữ dữ liệu ổn định** trước và sau cạnh clock:
 - Setup time**: Dữ liệu phải ổn định **trước** cạnh clock.
 - Hold time**: Dữ liệu phải ổn định **sau** cạnh clock.
- Sử dụng STA (Static Timing Analysis):**
 - STA kiểm tra toàn bộ các timing path để đảm bảo dữ liệu **không thay đổi** trong **setup/hold window**.
- Thiết kế đồng bộ:**
 - Dùng đồng hồ chung (single clock domain) hoặc
 - Nếu phải dùng nhiều đồng hồ (asynchronous domains) → **sử dụng synchronizer (double-flop synchronizer)**.

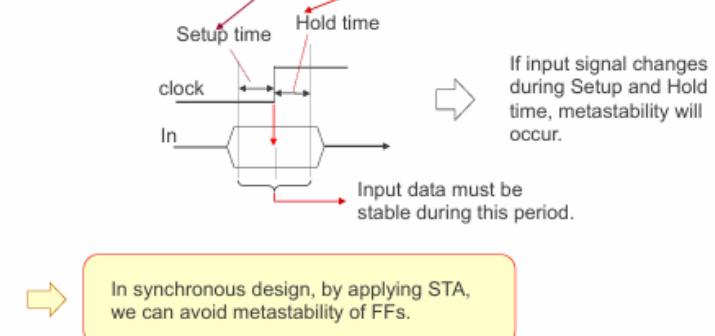
Thông số thực tế:

- Setup time** và **Hold time** thường rất nhỏ: Ví dụ: < 100 picoseconds (ps).

Hình minh họa flow logic: Nếu bạn cần minh vẽ thêm sơ đồ "Synchronizer 2-stage Flip-Flop" dùng để chống metastability khi truyền giữa 2 clock domain, mình có thể vẽ cho bạn nhé.



In many cases, metastability can be avoided by ensuring that inputs are held constant for specified periods **before** and **after** the clock pulse.



Typical Setup time or Hold time is less than 100 p sec.

Unknown Initial Value

- Ngay sau khi reset/power-up, giá trị của Flip-Flop là không xác định.
- Flip-Flop (FF) không đảm bảo sẽ là 0 hay 1 nếu chưa có khởi tạo đúng.
- (Tương tự SRAM: giá trị bit sau khi bật nguồn cũng random.)

Hậu quả

- **Nếu thiết kế logic phụ thuộc vào giá trị ban đầu (mà không reset/init rõ ràng):**
 - Có thể gây hành vi bất định (unpredictable behavior).
 - Gây ra bug logic rất khó bắt trong mô phỏng hay thực tế.
- **Nhiều lỗi hệ thống thực tế (field failures) đã xuất phát từ việc quên khởi tạo đúng FFs.**

Cách giải quyết

- Luôn reset các Flip-Flop ngay sau khi hệ thống bật nguồn:
 - Synchronous reset (reset theo clock).
 - Hoặc Asynchronous reset (reset ngay khi tín hiệu reset active).
- Đảm bảo mô phỏng và kiểm chứng kỹ tất cả các paths liên quan đến reset.
- Cẩn thận đặc biệt với khối FSM (Finite State Machine), vì trạng thái ban đầu phụ thuộc mạnh vào việc reset các FFs.
- Nếu logic của bạn dùng giá trị ban đầu chưa xác định để hoạt động → bạn đã tự tạo ra bug trong thiết kế.

Fan-out là gì?

- **Fan-out:** Số lượng cổng logic đầu vào mà một cổng đầu ra có thể điều khiển được.
- Do ảnh hưởng của **drop điện áp** (giảm áp) khi tải quá nhiều đầu vào từ một đầu ra.

Vấn đề xảy ra nếu fan-out quá lớn:

- **Output không còn đủ mức điện áp logic** (không còn rõ High/Low như yêu cầu).
- **Circuit có thể bị malfunction** (hoạt động sai).
- **Tăng thời gian trễ (delay)** do điện dung tải tăng lên.
- **Làm chậm tốc độ mạch.**

Cách khắc phục:

- **Chèn thêm Buffer:**
 - Buffer sẽ khuếch đại tín hiệu.
 - Buffer chia tải ra để giảm gánh nặng cho cổng logic ban đầu.
- Các **EDA tools** (công cụ thiết kế điện tử) sẽ tự động nhận biết và đề xuất thêm buffer khi cần.

Về tốc độ:

- **Tải lớn hơn → delay lớn hơn:** Delay đường truyền tăng vì điện trở và điện dung tổng tăng.
- Nếu **không chia tải hợp lý**, có khi: Một đường logic ít stage (ít cổng) nhưng nặng tải lại **chậm hơn** một đường nhiều stage mà chia tải tốt.

Ghi chú thêm:

- **Fan-out limit** tùy thuộc vào công nghệ (CMOS/TTL) và loại cổng.
- Trong chip thực tế, fan-out thường bị kiểm soát chặt chẽ để đảm bảo timing.

Unknown initial value of Flip-Flop

We can not tell the value of FF right after power on. (How about SRAM?)

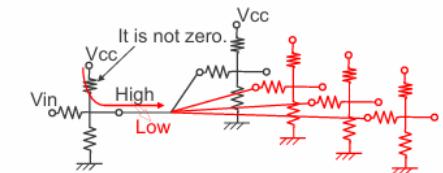
The output signal of FFs must be used after proper initialization sequence.

Many problems caused by unknown initial value of FF have been experienced. Pay special attention to this issue whenever you use FFs (or SRAM).

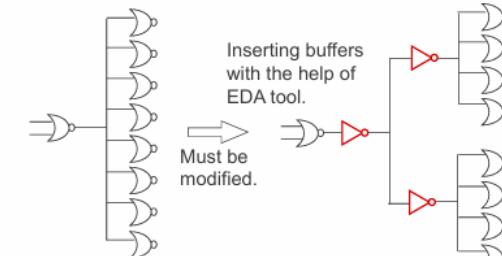
If a logic you designed uses the unknown initial values of FFs in a way that they cause unexpected operation, it means you have implemented a logic bug.

Fan-out

Because of output voltage drop, there is a limitation (fan-out) in the number of the circuits which can be connected to an output.

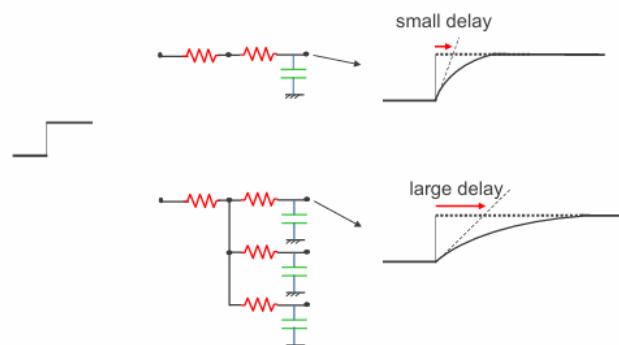


Fan-out problem causes malfunction of the circuits. EDA tools can help us to avoid this problem.



Page 58

Because larger load result in larger delay as shown below, fan out issue is also related to speed.



Sometimes delay of N stage gate logic is larger than that of having more stages gate logic if some gates have larger load.

Ground Bounce là gì?

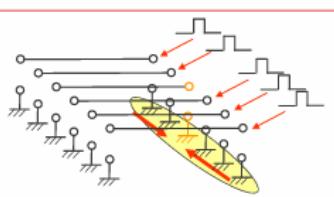
- Khi nhiều tín hiệu thay đổi trạng thái **đồng thời**, sẽ gây ra một dòng lớn chạy qua đường ground chung.
- **Điện áp ground** không còn duy trì ở mức 0V lý tưởng nữa — nó bị "nhảy lên" (bị xê dịch).

Ground bounce



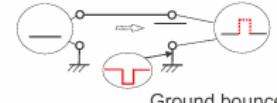
Cannot keep the potential of the ground at zero when there is a large current in the neighborhood.

Sharing the ground with other signals



Simultaneous signal change and ground bounce

A false signal can be observed when the potential of the ground is shaken by the other signal.



Ground bounce



Ground bounce creates false signal on certain signal line, thus sometimes causes a malfunction of a device.

Hậu quả của Ground Bounce:

- **Tạo tín hiệu giả (false signal):** Một số tín hiệu có thể bị "nhìn nhầm" giá trị do nhiễu nền.
- **Gây lỗi hoạt động (malfunction):** Chip hoạt động sai, nhận lệnh sai, treo hệ thống.

Nguyên nhân:

- **Nhiều đường tín hiệu chia chung một ground.**
- **Thay đổi đồng thời** gây dòng surge lớn → **shaking** của điện áp ground.

Minh họa:

- Ban đầu: Ground ổn định.
- Khi nhiều tín hiệu đổi trạng thái cùng lúc → dòng lớn → ground bị xô lệch → tín hiệu sai xuất hiện.

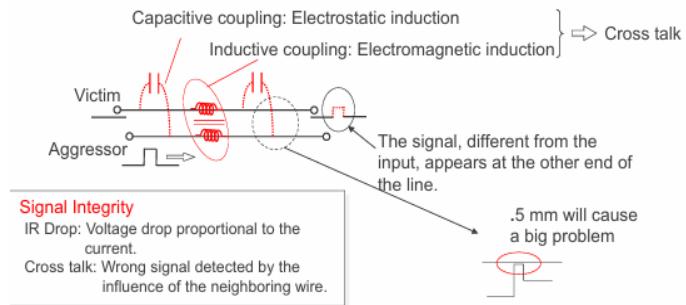
Cách phòng tránh Ground Bounce:

- **Chia nhỏ nhóm tín hiệu thay đổi cùng lúc** (phân bổ switching load).
- **Tăng số lượng pin ground** (nhiều chân ground hơn).
- **Bố trí layout cẩn thận** (tối ưu đường đi dòng điện).
- **Thêm decoupling capacitors** gần nguồn ground.

Cross Talk là gì?

- **Cross Talk** xảy ra khi một tín hiệu trên dây **Aggressor** ảnh hưởng đến tín hiệu trên dây **Victim**.
- Có hai kiểu coupling gây ra:
 - **Coupling điện dung (capacitive coupling):** Gây ra bởi điện trường.
 - **Coupling điện từ (inductive coupling):** Gây ra bởi từ trường.
- **Kết quả:** Một tín hiệu **khác** so với tín hiệu ban đầu bị xuất hiện trên dây nạn nhân.

Cross talk



Signal Integrity
IR Drop: Voltage drop proportional to the current.
Cross talk: Wrong signal detected by the influence of the neighboring wire.

The signal, different from the input, appears at the other end of the line.
.5 mm will cause a big problem

→ Cross talk creates false signal on certain signal line, thus sometimes causes a malfunction of a device.

→ IR drop creates insufficient power supply to certain circuit on silicon and limit miniaturization and operating frequency.

Hậu quả của Cross Talk:

- **Tín hiệu giả** xuất hiện trên đường dây → **gây lỗi hoạt động** cho mạch.
- **Ảnh hưởng Signal Integrity** (độ toàn vẹn tín hiệu).

Nguyên nhân:

- Các dây tín hiệu **quá gần nhau** (ví dụ chỉ cách nhau **0.5 mm** thôi cũng đủ gây lỗi).
- **Tốc độ tín hiệu cao** (switching nhanh) → coupling mạnh hơn.
- **Thiết kế layout không tối ưu**.

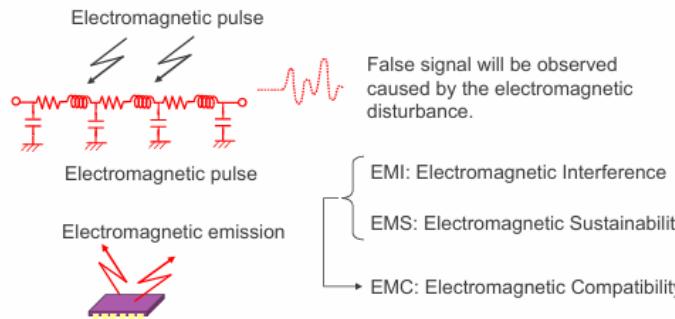
IR Drop là gì? (nhắc thêm trong hình)

- Khi dòng điện đi qua dây nguồn → tạo sụt áp (theo công thức: $V=IRV = IRV=IR$).
- **IR Drop** gây ra:
 - **Nguồn cung cấp điện không đủ** cho các block mạch.
 - **Giảm tốc độ hoạt động** hoặc **hỏng mạch** khi voltage sụt quá thấp.

EMC/EMI là gì?

- EMI (Electromagnetic Interference):**
 - Là sự **nhiều điện từ** gây ra bởi sóng điện từ hoặc xung điện từ.
 - EMI làm xuất hiện **tín hiệu giả** trên đường tín hiệu chuẩn → gây **lỗi vận hành** thiết bị.
- EMS (Electromagnetic Sustainability):**
 - Khả năng của thiết bị chịu đựng được nhiễu điện từ từ bên ngoài mà vẫn hoạt động bình thường.
- EMC (Electromagnetic Compatibility):**
 - Khả năng của thiết bị **không gây ra nhiễu** cho thiết bị khác và **chịu đựng** nhiễu từ môi trường.
 - EMC là yêu cầu **chuẩn quốc tế** khi thiết kế sản phẩm điện tử.

EMC/EMI



Electromagnetic interference may creates false signal on certain signal line, thus sometimes causes a malfunction of a device. It becomes difficult to comply with EMC guide lines as operating frequency of devices goes high.

Hậu quả của EMI:

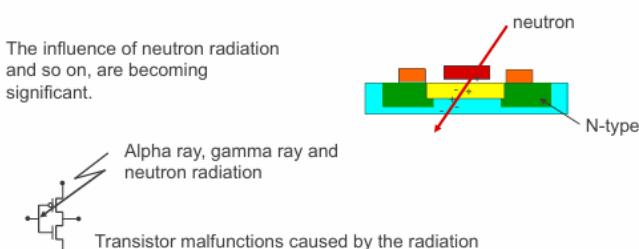
- Sai tín hiệu** trên các đường tín hiệu logic.
- Hỗn chúc năng** hoặc **reset bất ngờ**.
- Khó đạt tiêu chuẩn EMC** khi tần số hoạt động ngày càng **cao**.

Nguyên nhân:

- Xung điện từ** mạnh (ví dụ khi switching dòng lớn, switching tần số cao).
- Phát xạ điện từ** từ board mạch, chip, hoặc dây cáp.
- Tần số hoạt động cao** → phát ra sóng điện từ mạnh hơn.

Radiation from the package and cosmic ray

The influence of neutron radiation and so on, are becoming significant.



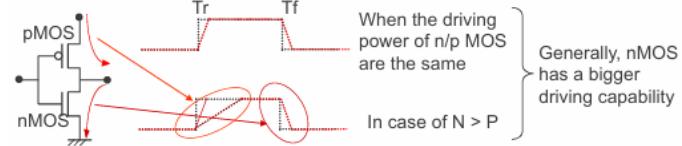
Large scale memory can not avoid this problem, therefore some error correction system is necessary for such large scale memory.

Page 63

Radiation from the package and cosmic ray

- Nguồn gốc bức xạ:**
 - Bức xạ neutron, tia alpha, tia gamma có thể **xuyên vào chip và tác động lên transistor** (nhất là vùng N-type).
 - Sự cố:**
 - Bức xạ làm transistor **bị nhiễu hoặc lỗi tạm thời** (gọi là **SEU - Single Event Upset**).
 - Đặc biệt nghiêm trọng đối với **bộ nhớ dung lượng lớn** (ví dụ: DRAM, SRAM).
- Kết luận:**
 - Không thể tránh khỏi** với bộ nhớ quy mô lớn.
 - Giải pháp:** Phải có hệ thống **sửa lỗi (Error Correction Code - ECC)** để đảm bảo dữ liệu an toàn.

The different characteristic of nMOS & pMOS



The rising becomes gentle because the driving power of pMOS is weak.

It becomes difficult to keep the duty ratio of on/off such as the clock at 50%.



Clock skew problem may become critical when inserting a buffer into the clock line.

The different characteristic of nMOS & pMOS

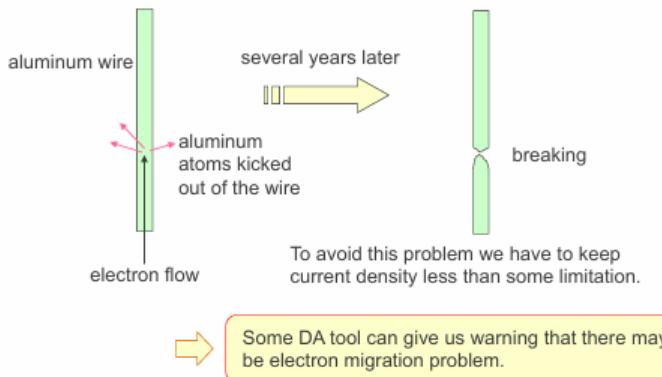
- Sự khác biệt nMOS / pMOS:**
 - nMOS mạnh hơn pMOS trong khả năng kéo dòng.
 - Khi thiết kế inverter hay buffer, **rise time (Tr)** và **fall time (Tf)** sẽ **không cân đối** → gây ra tín hiệu **không đổi xứng**.
- Hậu quả:**
 - Duty cycle** (tỷ lệ xung cao/thấp) bị lệch, **không giữ được 50%** như yêu cầu ở tín hiệu clock.
 - Khi chèn buffer vào đường clock, dễ phát sinh **clock skew** → ảnh hưởng timing toàn hệ thống.
- Lời cảnh báo:** Clock skew sẽ **cực kỳ nguy hiểm** nếu không tính toán kỹ rise/fall delay khi thêm buffer vào clock tree.

Electron migration

- Hiện tượng:** Khi dòng điện chạy lâu trong dây nhôm hoặc đồng, **các nguyên tử kim loại bị electron "đẩy" ra khỏi dây** → gây hiện tượng **đứt dây** sau nhiều năm hoạt động.
- Nguyên nhân:** **Mật độ dòng điện (current density)** quá cao.
- Giải pháp:**
 - Giới hạn mật độ dòng điện** trong thiết kế.
 - DA tool** (ví dụ: thiết kế bằng Synopsys IC Compiler hay Cadence Innovus) có thể **cảnh báo sớm** về nguy cơ **Electron Migration**.

Electron migration

Atoms of wire material, aluminum or copper, are kicked out of the wire line by electrons. This sometimes causes a breaking of wire.



Page 65

Other issues

1. Slow rising around Vth

- Slow rise** của tín hiệu đầu vào (vượt qua ngưỡng V_{th} quá chậm) → gây ra:
 - Over current** trong MOS transistor → nóng chip, giảm tuổi thọ.

2. Ringing around Vth

- Ringing** (dao động qua lại quanh V_{th}) → dễ gây:
 - False signal** (nhiều sai).

3. Runt pulse

- Một **xung quá hẹp**, không đạt được mức "High" hoặc "Low" hợp lệ.
- Gây lỗi khi nhận tín hiệu.

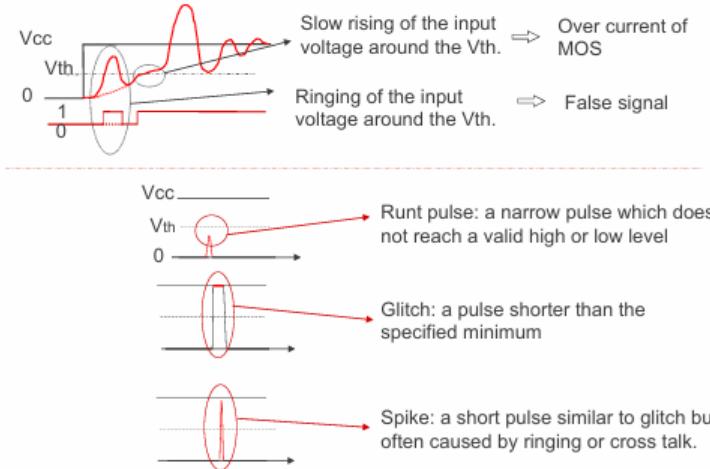
4. Glitch

- Một **xung ngắn bất thường** (nhỏ hơn thời gian xung yêu cầu).
- Dễ gây sai sót trong mạch tuần tự.

5. Spike

- Xung đột biến rất ngắn** thường do ringing hoặc cross-talk.
- Có thể làm flip-flop latch nhầm giá trị.

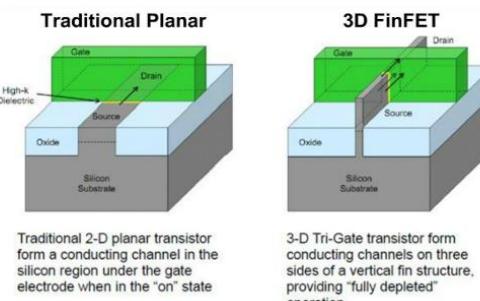
Other issues



FINFET

A FinFET is a MOSFET with the channel elevated so the gate can surround it on three sides.

FinFET technology provides numerous advantages over bulk CMOS, such as higher drive current for a given transistor footprint, hence higher speed, I-V curves get flatter, meaning lower dynamic power consumption, **lower leakage**, hence lower power consumption, no random dopant fluctuation, hence better mobility and scaling of the transistor beyond 28nm.



FinFET là gì?

- FinFET** là một loại **MOSFET** đặc biệt.
- Khác biệt:** Kênh dẫn (**channel**) được nhô lên như **một vây cá ("fin")** để **gate bao quanh 3 mặt**.
- Hệ quả:** Gate kiểm soát channel tốt hơn → giảm rò rỉ, tăng hiệu suất.

Lợi ích của FinFET so với Planar CMOS truyền thống:

Planar CMOS	3D FinFET
Gate chỉ kiểm soát 1 mặt channel.	Gate kiểm soát 3 mặt channel.
Khó giảm kích thước dưới 28nm do leakage tăng.	Tốt cho tiến trình <28nm (14nm, 7nm, 5nm...)
Leakage lớn khi thu nhỏ transistor.	Lower leakage (ít rò điện hơn).
Hiệu suất giới hạn.	Higher drive current → nhanh hơn, mạnh hơn.
Dễ bị nhiễu do dopant fluctuation.	Stable mobility → tín hiệu ổn định hơn.

Các từ khóa quan trọng:

- **Higher drive current:** Dòng dẫn lớn hơn → transistor nhanh hơn.
- **Lower leakage:** Ít rò điện hơn → tiết kiệm năng lượng.
- **No random dopant fluctuation:** Hạn chế sai lệch do tạp chất ngẫu nhiên → độ tin cậy cao hơn.
- **Fully depleted operation:** Channel hoàn toàn kiểm soát bởi gate.

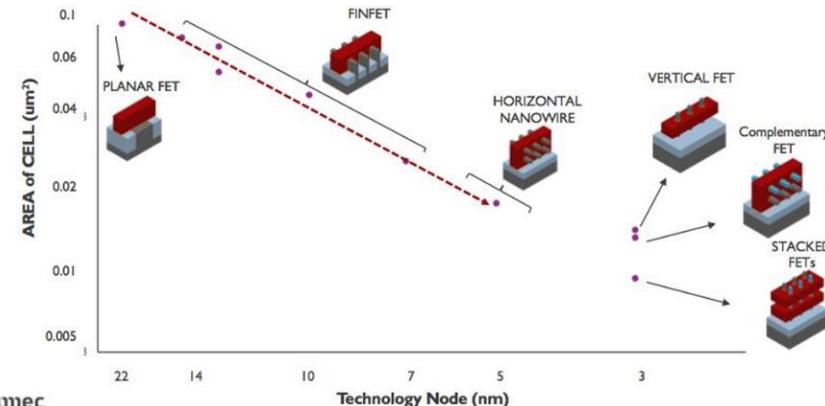
Hình ảnh phân tích:

- **Planar:** Gate đè lên kênh phẳng.
- **FinFET:** Gate ôm 3 mặt của "vây" (fin) nhô lên từ substrate → kiểm soát dòng điện cực kỳ tốt.

Advantages of FinFET

1. **Higher Transistor Density**
→ Nhiều transistor hơn trong cùng diện tích chip (so với MOSFET truyền thống).
2. **Better Short-Channel Control**
→ Fins giúp kiểm soát tốt hiện tượng short-channel khi transistor thu nhỏ.
3. **No Dopant Fluctuations**
→ Không cần doping nặng cho channel → giảm biến thiên dopant, tăng độ tin cậy.
4. **Reduced Leakage Current**
→ Gate bao quanh channel → giảm dòng rò rỉ, tiết kiệm điện năng.
5. **Lower Power Consumption**
→ Ít leakage → ít tiêu hao công suất khi hoạt động và standby.
6. **Flexible Drive Strength**
→ Tăng drive strength bằng cách thêm số lượng hoặc chiều dài các Fin.
7. **Higher Speed**
→ Dòng dẫn cao hơn → switching nhanh hơn → hiệu suất tốt hơn MOSFET.
8. **Easy Multi-Gate Fabrication**
→ Dễ dàng chế tạo transistor nhiều gate, không phức tạp như planar MOSFET.
9. **Better Subthreshold Slope and Higher Gain**
→ Độ dốc subthreshold và độ lợi điện áp cao hơn so với MOSFET phẳng.

What's after FINFET



Ghi chú nhỏ:

- **Subthreshold slope tốt** → dễ điều khiển on/off.
- **Higher voltage gain** → tín hiệu khuếch đại mạnh hơn.

What Comes After FinFET?

1. Planar FET (MOSFET)

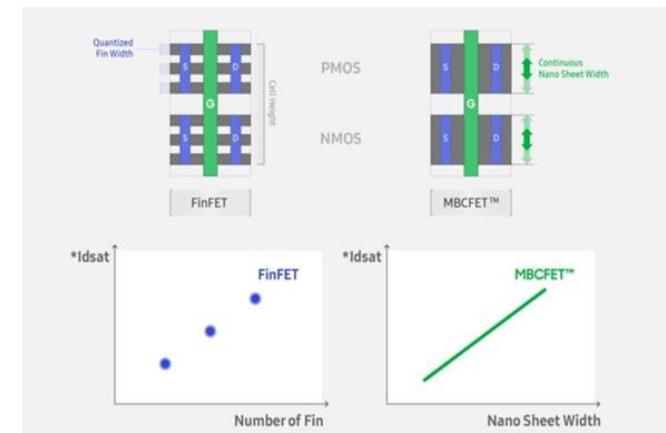
- Dùng cho các node lớn (22nm trở về trước).
- Khi xuống các node nhỏ, Planar FET không còn kiểm soát dòng rò tốt, nên buộc phải chuyển sang FinFET.

TSMC stays with FINFET for 3nm, Samsung might use MBCFET

2. FinFET

- Dùng từ ~16nm, phổ biến ở 10nm, 7nm.
- **Gate bao quanh kênh (channel)** giúp kiểm soát dòng rò và cho phép transistor hoạt động ở kích thước nhỏ hơn.

MBCFET (Samsung)



3. Horizontal Nanowire (GAAFET)

- **GAAFET** = Gate All Around FET.
- Thay vì 1 "fin", nhiều nanowire nhỏ được gate bao trọn 360°.
- Dự kiến dùng cho 5nm → 3nm → 2nm.
- **Samsung gọi nó là MBCFET (Multi-Bridge Channel FET).**

4. Vertical FET, Complementary FET, Stacked FETs (tương lai xa hơn)

- Các kiến trúc transistor dạng **dựng đứng (vertical)** để tiết kiệm diện tích hơn nữa.
 - **Stacked FETs** = nhiều tầng FET xếp chồng, tối đa hóa mật độ transistor.

Một số điểm nhấn:

- **TSMC:** Vân dùng **FinFET** cho 3nm trước khi chuyển sang GAAFET.
 - **Samsung:** Có thể sớm chuyển sang **MBCFET** ở 3nm để cạnh tranh.

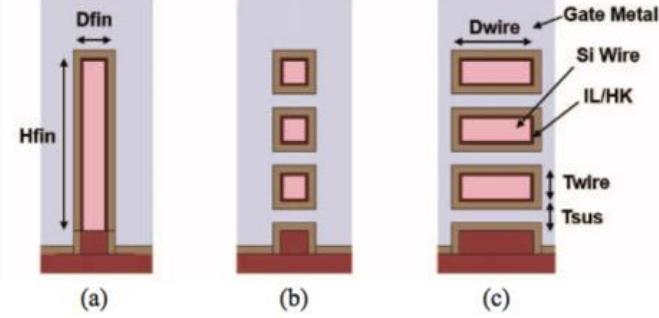
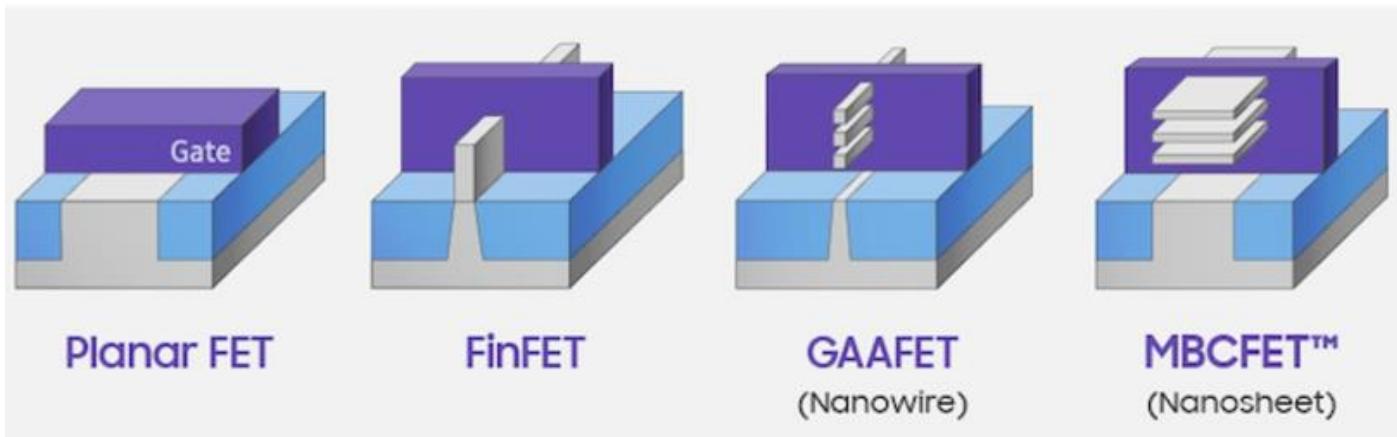


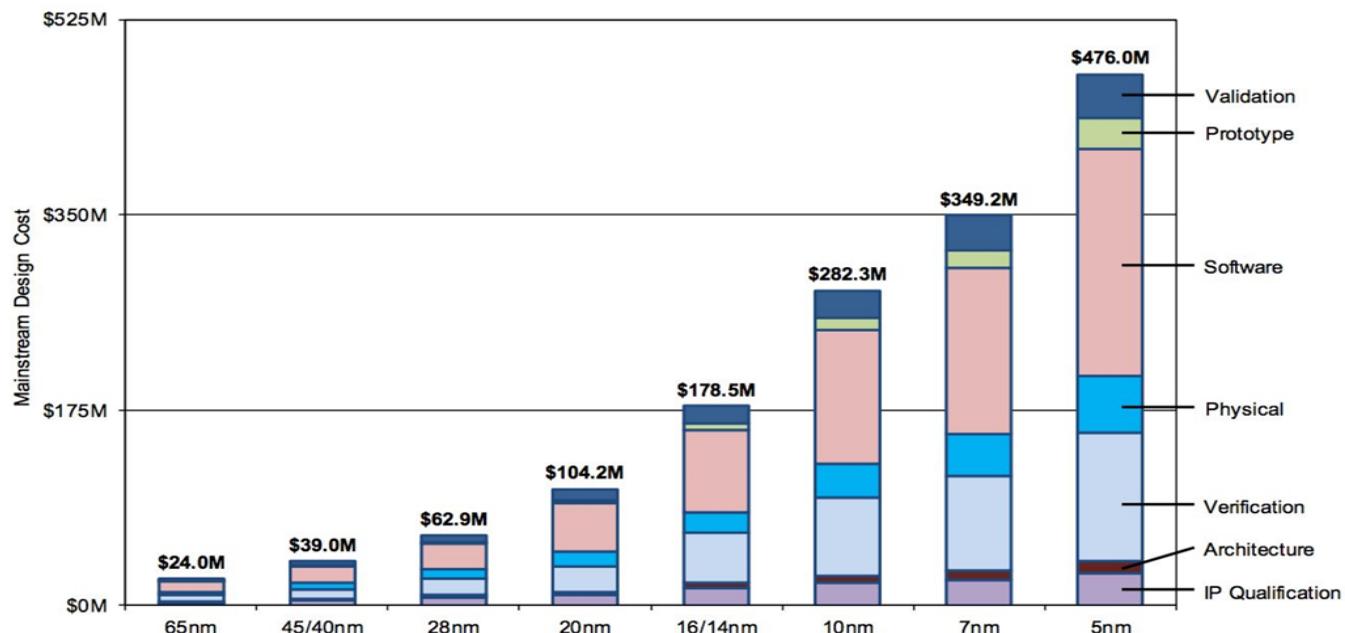
Fig. 5: Cross-section simulation of (a) finFET, (b) nanowire, and (c) nanosheet. Source: IBM.

New FETs

A **gate-all-around** (GAA) FET, abbreviated GAAFET, and also known as a surrounding-gate transistor (SGT), is 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.



Design Cost



1. Gate Delay (Tg)

Tg = f(input slew, output load)

- **Input slew** (tốc độ thay đổi tín hiệu đầu vào) càng lớn → Gate delay càng lớn.
- **Output load** (Cequ = tổng điện dung tương đương) càng lớn → Gate delay cũng càng lớn.

🔧 Cách cải thiện **Gate Delay**:

- **Tối ưu Cin** (diện dung đầu vào của tầng tiếp theo): dùng cell nhỏ nếu có thể mà vẫn giữ được timing.
- **Match Cequ và sức mạnh driver**: chọn size vừa đủ mạnh để giảm tiêu thụ power.
- **Dùng cell loại Low Vt** (điện áp ngưỡng thấp) cho critical path:
 - Pros: nhanh hơn!
 - Cons: **Rò rỉ dòng (Leakage) cao hơn** → chỉ nên dùng ở critical path.

2. Wire Delay

Wire delay chủ yếu do **R** (diện trở) và **C** (diện dung).

Công thức đơn giản (Elmore delay): $T_{pd} = \sum (R_{(n-i)} \times C_i)$

(tổng các điện trở và điện dung trên đường dây)

🔧 Cách cải thiện **Wire Delay**:

- **Giảm R và/hoặc C:**
 - Nếu dây dài: Resistance **R** chiếm ưu thế → tăng độ rộng dây để giảm R trước.
 - Nếu dây quá dài: **Chèn buffer** giữa đường dây → delay giảm một nửa, thậm chí còn hơn (1/3 hoặc 1/4).

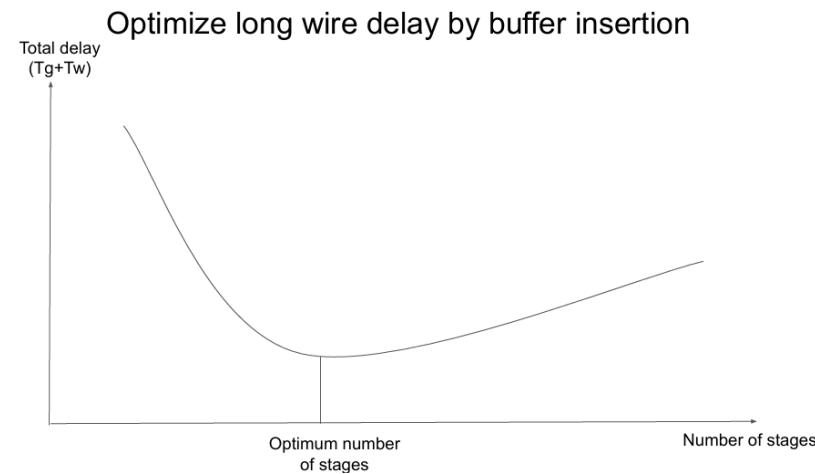
📝 Ý nghĩa biểu đồ:

- **Trục dọc:** Total delay = Gate delay (Tg) + Wire delay (Tw)
- **Trục ngang:** Number of stages (số lượng buffer/chia nhỏ đoạn truyền)

Điễn giải:

- Nếu **không chèn buffer** (ít stages): wire rất dài → **wire delay lớn**.
- Nếu **chèn quá nhiều buffer** (nhiều stages): tuy giảm wire delay nhưng **gate delay (Tg)** lại tăng do mỗi buffer cũng có delay riêng.
- **Có một số lượng tối ưu buffer** (optimum number of stages) mà tổng delay (Tg + Tw) là **nhỏ nhất**.

● **Điểm thấp nhất trên đồ thị** = số lượng buffer tối ưu cần chèn để đạt delay nhỏ nhất.



💬 Tóm lại:

- **Wire dài** → cần chèn buffer để delay tổng nhỏ nhất.
- **Không quá ít, không quá nhiều** buffer → cần tìm số lượng tối ưu.
- **Phổ biến trong P&R stage** (Place & Route): tool sẽ tự động buffer insertion cho các net dài.

💡 Công thức gần đúng trong thực tế:

Số lượng stages tối ưu (n_{opt}) có thể tính gần đúng theo:

$$n_{opt} \approx \sqrt{\frac{R_{wire} C_{wire}}{R_{buf} C_{buf}}}$$

- R_{wire}, C_{wire} : điện trở, điện dung của đường dây dài
- R_{buf}, C_{buf} : điện trở ra, điện dung vào của buffer

CLOCK SKEW

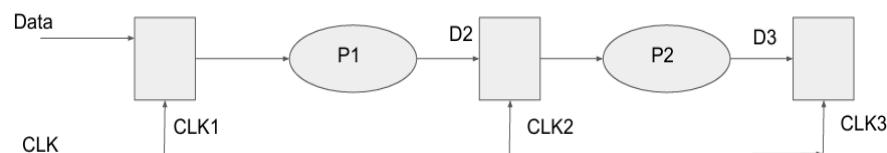
1. Tóm tắt hệ thống

- Bạn có **3 flip-flop** (DFF) nối với nhau: FF1 → logic P1 → FF2 → logic P2 → FF3.
- Mỗi DFF được clock bởi **CLK1, CLK2, CLK3** tương ứng.
- Clock đầu vào là CLK, nhưng do mạch clock tree nên các tín hiệu CLK1, CLK2, CLK3 sẽ **không đến cùng lúc** → **gây ra clock skew**.

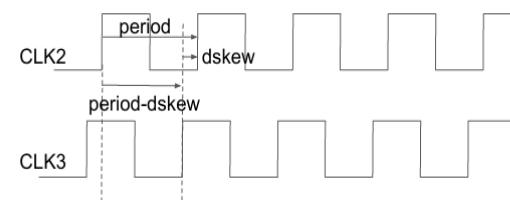
2. Giải thích công thức delay

- Td2 = Tclk1 + Tck2q + Tp1**
- o Tclk1: thời điểm CLK1 đến FF1.
- o Tck2q: thời gian từ khi clock đến cho đến khi FF1 xuất ra dữ liệu.
- o Tp1: delay của khối logic P1. → Đây là tổng thời gian cần để dữ liệu đi từ FF1 qua P1.
- Tclk2 = Tclk + skew2**
- o Clock đến FF2 (CLK2) bị trễ một lượng skew2 so với clock gốc CLK.
- Tclk3 = Tclk + skew2 - dskew**
- o Clock đến FF3 (CLK3) bị lệch thêm **dskew** so với CLK2.
- o Nếu dskew > 0 ⇒ CLK3 đến sớm hơn CLK2.
- Td3 = Tclk + skew2 + Tck2q + Tp2**
- o Tổng delay từ khi clock gốc đến cho đến khi đi hết P2.

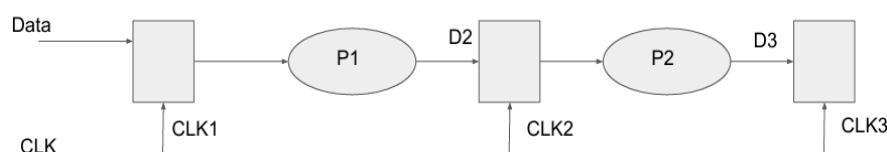
CLOCK SKEW



$Td2 = Tclk1 + Tck2q + Tp1$
 $Tclk2 = Tclk + skew2$
 $Tclk3 = Tclk + skew2 - dskew, dskew > 0 \rightarrow CLK3 \text{ is earlier than } CLK2$
 $Td3 = Tclk + skew2 + Tck2q + Tp2$
 The clock skew between CLK2 and CLK3 makes the path P2 more critical



CLOCK SKEW



- What causes clock skew?
 - CTS
- Consequences of clock skew
 - Reduced performance
 - Failed functional
- How to reduce clock skew?
 - Clock spines
 - Clock mesh
 - Htree
 - ...

3. Ý nghĩa về timing

- Vì CLK3 đến sớm hơn CLK2, nên khoảng thời gian cho dữ liệu di chuyển qua P2 sẽ ngắn lại.
- Điều này làm cho **đường P2 trở thành đường critical path** (đường giới hạn tốc độ chip).
- Nếu không xử lý tốt clock skew, **P2 có thể vi phạm timing** (setup violation).

4. Giải thích waveform (dạng sóng)

- **CLK2 và CLK3** có dạng sóng vuông nhưng lệch nhau dskew.
- Thay vì có full chu kỳ period để truyền dữ liệu từ P2 tới FF3, giờ chỉ còn:
 - (**period - dskew**) thời gian thực sự.
- Vì vậy, clock skew làm **giảm window thời gian** truyền dữ liệu → khó đạt timing hơn.

Nguyên nhân gây ra Clock Skew

- CTS (Clock Tree Synthesis):**
 - Là quá trình thiết kế mạch clock phân phối xung nhịp từ nguồn clock (gốc) tới các flip-flop khắp chip.
 - Trong thực tế, do:
 - Chiều dài dây clock khác nhau,
 - Điện trở (R), điện dung (C) trên dây,
 - Các driver (buffer) clock không đồng nhất, → tín hiệu clock sẽ đến các flip-flop tại các thời điểm khác nhau.
 - Đây chính là nguyên nhân tạo ra **clock skew**.

2. Hậu quả của Clock Skew

- Giảm hiệu suất (Reduced Performance):**
 - Clock skew làm giảm thời gian thực tế cho dữ liệu truyền qua các logic.
 - Từ đó làm **giảm tần số tối đa** mà chip có thể chạy được.
- Hỗn chúc năng (Failed Functional):**
 - Nếu dữ liệu chưa kịp ổn định trước khi clock tới (setup violation) hoặc bị hold violation, chip sẽ:
 - **Sai kết quả tính toán,**
 - **Gây lỗi logic,**
 - **Không hoạt động đúng.**

3. Cách giảm Clock Skew

- **Clock Spines:**
 - Thiết kế clock theo dạng xương sống (spine) dọc theo chip → clock đi đều, hạn chế lệch.
- **Clock Mesh:**
 - Tạo một lưới dày các đường clock kết nối với nhau → phân phối clock đều khắp nơi.
 - Giảm tối đa skew nhưng tốn diện tích và năng lượng.
- **H-tree:**
 - Clock được chia nhánh theo hình cây chữ H đều đặn → đảm bảo độ dài đường clock tới các flip-flop gần bằng nhau.
- **Các phương pháp khác:**
 - Buffer insertion (chèn buffer đúng chỗ),
 - Delay balancing (cân bằng delay),
 - Post-CTS optimization v.v.

POWER

Tổng công suất tiêu thụ của mạch:

$$\text{Total Power} = P_{\text{switch}} + P_{\text{short circuit}} + P_{\text{leakage}}$$

Trong đó:

1. Switching Power (P_{switch})

- Công suất tiêu thụ khi **mạch chuyển trạng thái** (từ 0 → 1 hoặc 1 → 0).
- Công thức:

$$P_{\text{switch}} = aCV^2f$$

- C = Điện dung tải (load capacitance),
- V = Điện áp cung cấp,
- f = Tần số hoạt động,
- a = Độ hoạt động switching (xác suất chuyển trạng thái).

➡ Switching Power chiếm phần lớn trong công suất khi tần số cao hoặc tải lớn.

2. Short-Circuit Power ($P_{\text{short circuit}}$)

- Công suất tiêu thụ khi cả NMOS và PMOS cùng bật trong thời gian ngắn lúc chuyển mạch (slew rate không lý tưởng).
- Phụ thuộc vào:
 - Slew: tốc độ cạnh lên/xuống của tín hiệu.
 - Tần số hoạt động.

➡ Cách giảm:

- Dùng cell mạnh (strong cell) → transistor to (large width) hoặc điện áp ngưỡng thấp (low V_t).

3. Leakage Power (P_{leakage})

- Công suất tiêu hao dù **mạch đang không hoạt động** (khi standby).
 - Do dòng rò qua transistor (subthreshold leakage, gate leakage...).
- ➡ Cách giảm:
- Dùng cell yếu (weak cell), tức là transistor nhỏ (min width) hoặc điện áp ngưỡng cao (high V_t).

Tùy thuộc vào ứng dụng (mobile, server, IoT...) thì loại công suất nào sẽ chiếm ưu thế → ta tối ưu phù hợp.

SIGNAL INTEGRITY

Vấn đề về Độ nguyên vẹn tín hiệu (Signal Integrity) và cách khắc phục:

1. Driver Optimization:

- Chọn độ mạnh driver **vừa đủ** với tải:
 - Không được quá yếu (gây chậm trễ),
 - Cũng không quá mạnh (gây overshoot, ringing...).

2. Buffer Insertion:

- Khi dây rất dài → ngay cả driver lớn cũng không đủ.
- **Thêm buffer** để tái tạo tín hiệu, chống suy hao (attenuation).

3. Crosstalk (Nhiều xuyên kẽm):

- Khi dây gần nhau, tín hiệu dây này ảnh hưởng dây kia do điện dung ký sinh.

→ Các cách xử lý:

- **Tăng khoảng cách** giữa các dây để giảm điện dung ghép (coupling cap).
- **Thêm dây chắn (shielding)** đối với các tín hiệu quan trọng.
- **Tăng độ mạnh** của driver cho dây nạn nhân (victim nets).
- **Giảm độ mạnh** của driver dây gây nhiễu (aggressor nets).

4. Sử dụng NDR (Non Default Rule):

- Quy định riêng cho các đường tín hiệu quan trọng:
 - Dây rộng hơn,
 - Khoảng cách lớn hơn bình thường,
 - Giảm ảnh hưởng nhiễu.

Questions

1. Clock skew:

- a. What is clock skew? Is it important? Why?
- b. What causes clock skew? How to reduce the clock skew?
- c. Is global clock skew or local clock skew important? Why?

2. Why is transition time measured from 10 to 90%?

3. Does improving gate delay also reduce wire delay?

4. What is the plus/minus of increasing the transistor width?

5. What are the components of power dissipation in a design? Which one is the most important?

6. What is crosstalk? When does crosstalk become significant? How to avoid/reduce crosstalk?

1. Clock Skew:

a. Clock Skew là gì? Nó có quan trọng không? Tại sao?

Clock skew là sự chênh lệch về thời gian đến của xung clock tại các phần tử chốt khác nhau trong cùng một mạch số.

Clock skew rất quan trọng vì nó ảnh hưởng trực tiếp đến thời gian thiết lập (setup) và thời gian giữ (hold) dữ liệu, từ đó tác động đến tính đúng đắn chức năng và hiệu suất tổng thể của thiết kế. Clock skew lớn có thể gây lỗi timing (vi phạm setup hoặc hold).

b. Nguyên nhân gây ra clock skew? Cách giảm clock skew?

Nguyên nhân chính bao gồm: sự bất cân đối trong mạng phân phối clock (Clock Tree Synthesis - CTS), sự khác biệt về tải điện dung, độ dài dây dẫn không đồng đều, và ảnh hưởng từ biến thiên tiến trình (PVT variations).

Các phương pháp giảm clock skew bao gồm:

- Thiết kế mạng clock đối xứng (clock spines, clock mesh, H-tree).
- Buffer insertion và sizing hợp lý.
- Sử dụng kỹ thuật balancing delay.
- Dùng công nghệ CTS tối ưu hóa skew-aware.

c. Clock skew toàn cục (global) hay clock skew cục bộ (local) quan trọng hơn? Tại sao?

Cả hai đều quan trọng nhưng ảnh hưởng theo cách khác nhau. Clock skew cục bộ tác động trực tiếp đến các đường truyền tín hiệu nhanh và nhỏ, trong khi clock skew toàn cục ảnh hưởng đến sự đồng bộ của toàn hệ thống. Trong nhiều trường hợp, local skew ảnh hưởng nhiều hơn đến timing critical paths và cần được kiểm soát chặt chẽ hơn.

2. Tại sao thời gian chuyển mạch được đo từ 10% đến 90%?

Khoảng đo từ 10% đến 90% biên độ tín hiệu loại bỏ ảnh hưởng của nhiễu nhỏ và hiện tượng overshoot/undershoot, giúp phản ánh trung thực tốc độ chuyển đổi logic thực sự của tín hiệu.

3. Việc cải thiện gate delay có đồng thời cải thiện wire delay không?

Không hoàn toàn. Gate delay liên quan đến thời gian đáp ứng của các cổng logic, còn wire delay chủ yếu phụ thuộc vào điện trở (R) và điện dung (C) của dây dẫn. Tuy nhiên, việc giảm thời gian switching của gate có thể gián tiếp giảm yêu cầu về thời gian tổng thể, nhưng không trực tiếp giảm wire delay.

4. Ưu điểm và nhược điểm của việc tăng chiều rộng transistor là gì?

- Ưu điểm:** Giảm điện trở on-resistance → cải thiện tốc độ switching và giảm gate delay.
- Nhược điểm:** Tăng điện dung gate → làm tăng công suất chuyển mạch (dynamic power) và tiềm ẩn nguy cơ gây crosstalk do dây dẫn to hơn.

5. Các thành phần của tiêu thụ công suất trong thiết kế? Thành phần nào quan trọng nhất?

- Switching Power (Pswitch):** Tỉ lệ thuận với hoạt động chuyển mạch và tải điện dung.
- Short-Circuit Power (Pshort-circuit):** Xảy ra trong thời gian ngắn khi cả NMOS và PMOS đều dẫn.
- Leakage Power (Pleakage):** Công suất rò rỉ do dòng điện rò khi transistor ở trạng thái tắt.
Tùy thuộc vào ứng dụng: với tần số cao hoặc khối lượng switching lớn, switching power chiếm ưu thế; trong các thiết kế tiết kiệm năng lượng hoặc standby, leakage power là yếu tố quan trọng nhất.

6. Crosstalk là gì? Khi nào crosstalk trở nên đáng kể? Cách tránh/giảm crosstalk?

- Crosstalk** là hiện tượng tín hiệu trên một đường dây gây ra nhiễu điện từ trên đường dây lân cận, do điện dung và điện cảm ký sinh.
- Crosstalk trở nên đáng kể khi dây dẫn song song kéo dài, khoảng cách giữa các dây nhỏ, hoặc khi switching speed cao.
- Biện pháp giảm crosstalk:**
 - Tăng khoảng cách giữa các dây dẫn.
 - Chèn dây chắn (shielding critical nets).
 - Chèn buffer để chia nhỏ đoạn dây dài.
 - Điều chỉnh độ mạnh yếu của driver.
 - Sử dụng NDR (Non-Default Rules) để kiểm soát spacing trong layout.