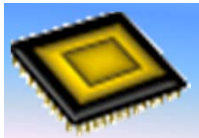
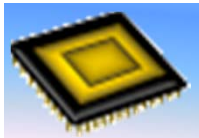
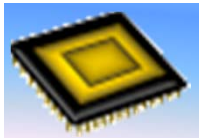


CHAPTER 5

CMOS Circuit and Logic Design



Outline



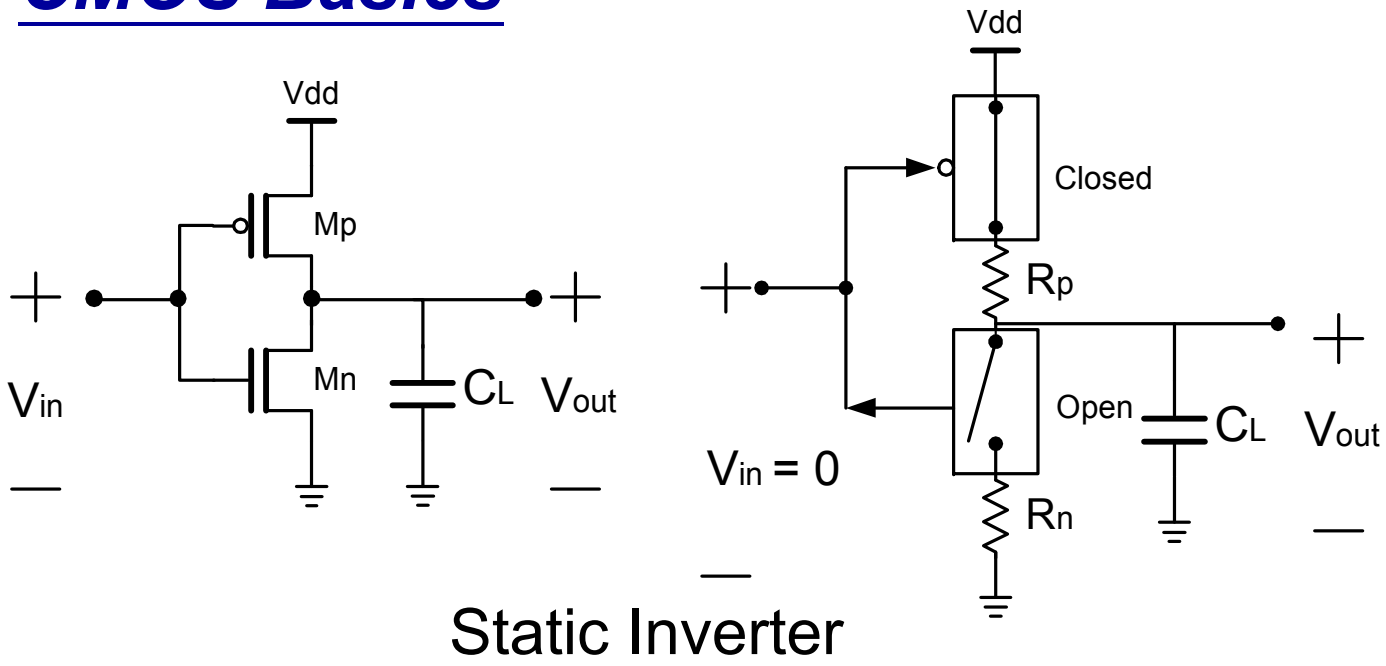
Chapter 5

- The demand for high-performance CMOS VLSI circuit：
  - High operation speed
  - High packing density
  - Low power dissipation
- Logic Structure
  - Static logic：No floating output nodes
  - Dynamic logic：a lower bound on the speed of the clock
  - Transmission gate logic：MOS action as a switch
  - Asynchronous logic：no globe clock

5.1 Static Logic Circuit

5.1.1 Static Logic Circuit

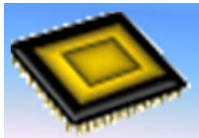
（A）CMOS Basics



Static Inverter

Active pull down device：NMOS

Active pull up device：PMOS

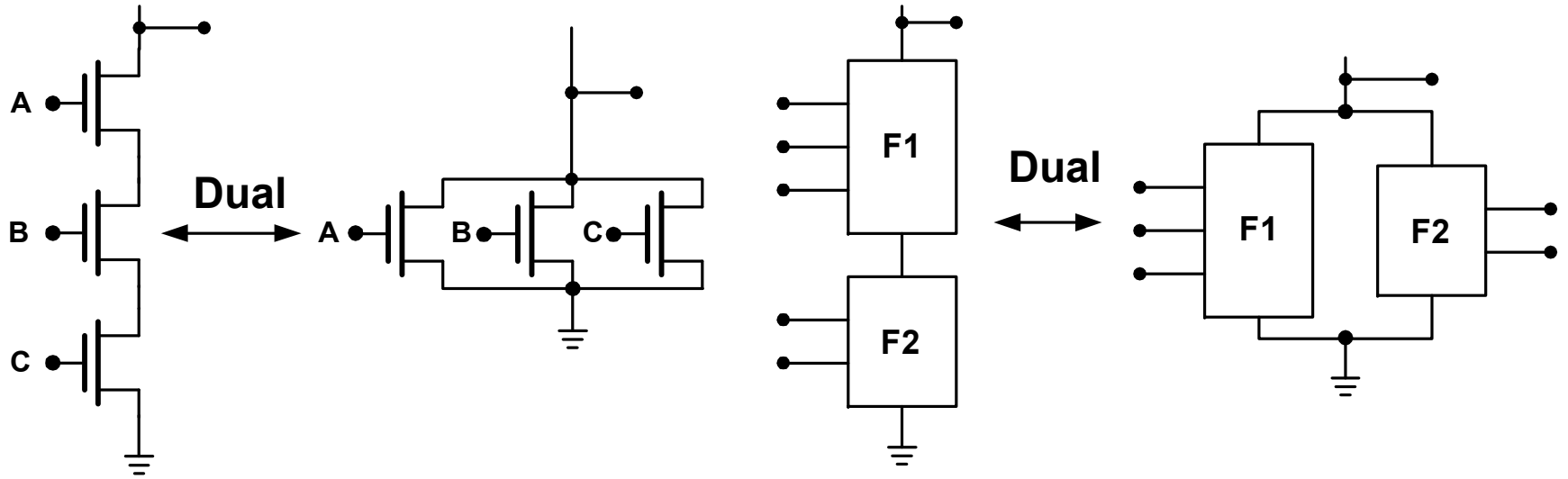


Chapter 5

- The output of CMOS must make a full voltage swing between VDD and GND . This is important to guarantee the zero static power dissipation and increase the noise margin .
- CMOS is a ratio less logic , so that makes the circuit design simpler , Functionally , it is always correct without taking into account the ratio between pull-up and pull-down device .
- Only transition power dissipation
- Output rise and fall transition time can be made the same by account the ration between pull-up and pull-down device .

（B）Combination Logic

NMOS Logic Formation and Naturally Dual Network



F = A*B*C　　F = A+B+C　　F = F1 * F2　　F = F1 + F2

(a) Rule 1：
Series - connected
NMOSFETs

(b) Rule 2：
Parallel - connected
NMOSFETs

(c) Rule 3：
Series - connected
Logic Blocks

(d) Rule 4：
Parallel - connected
Logic Blocks

- Conversion of Dual

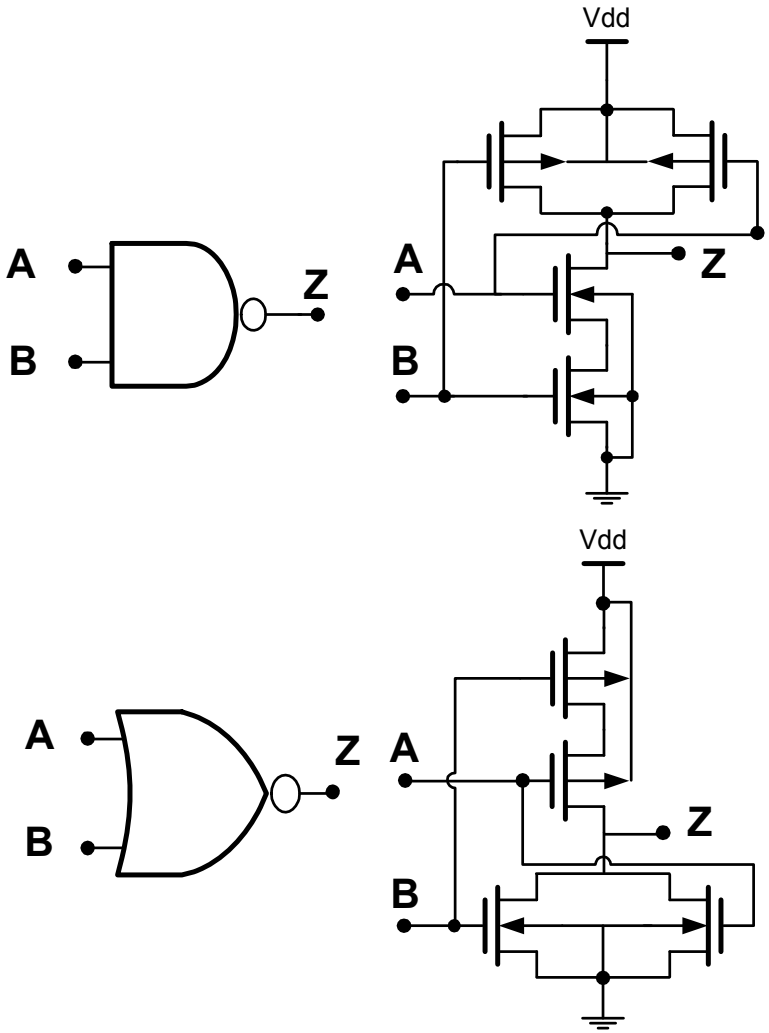
  AND ⇔ OR

  SERIAL ⇔ PARALLELR

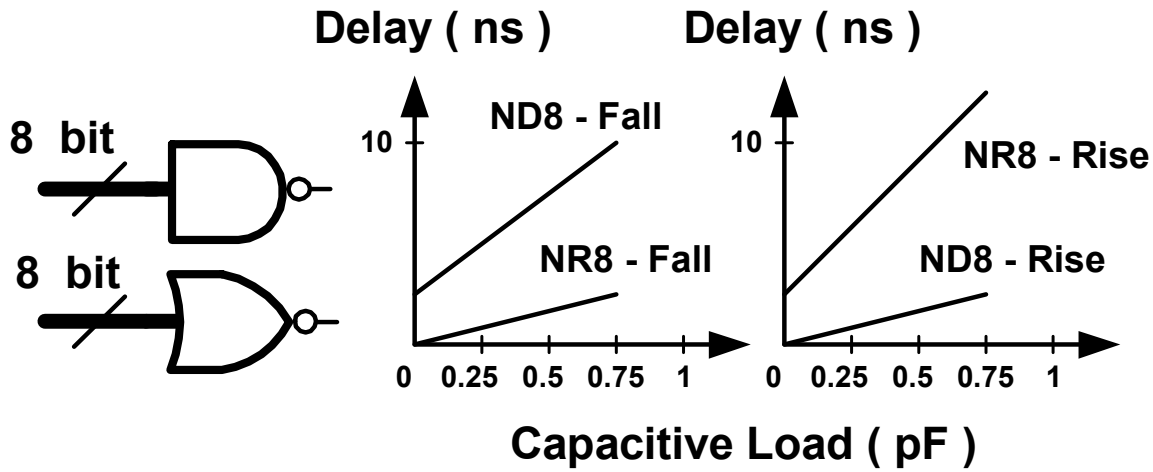
V.S.

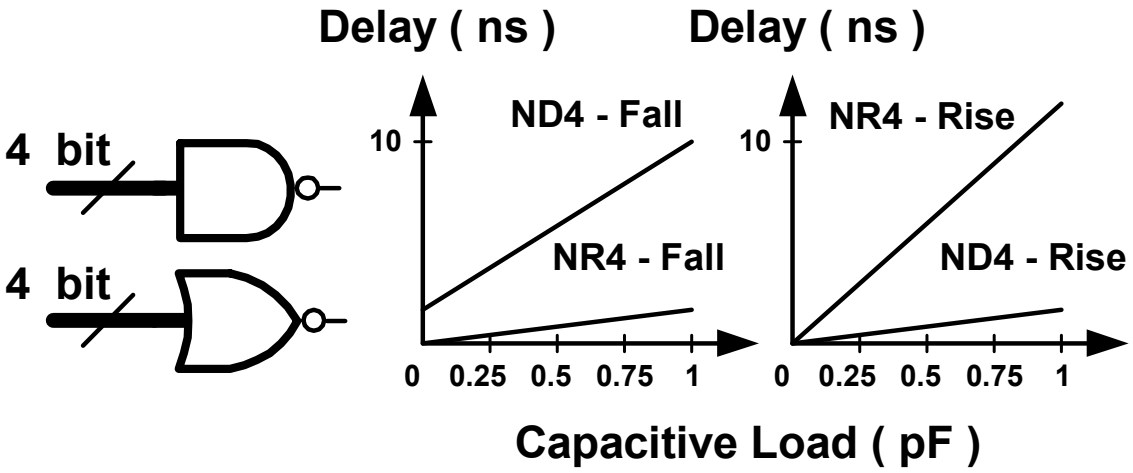
- NMOS Network ⇔ PMOS Network

（C）NAND Gate and NOR Gate

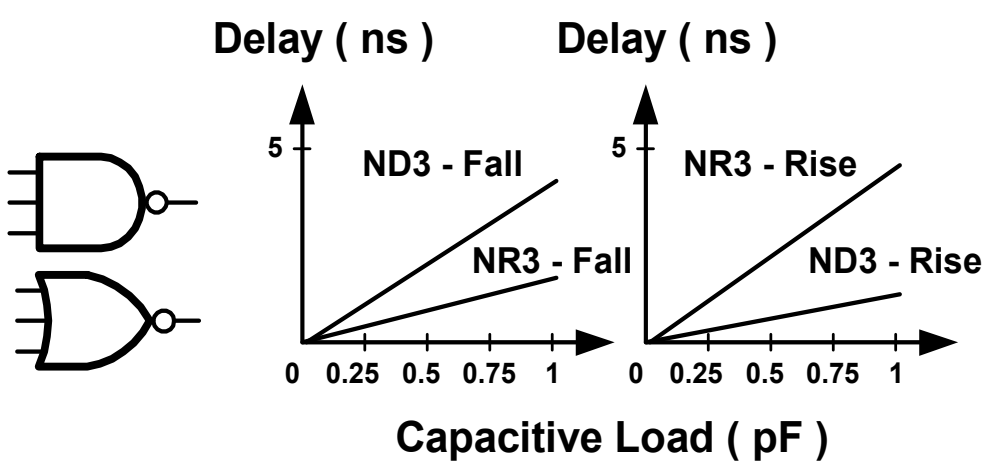


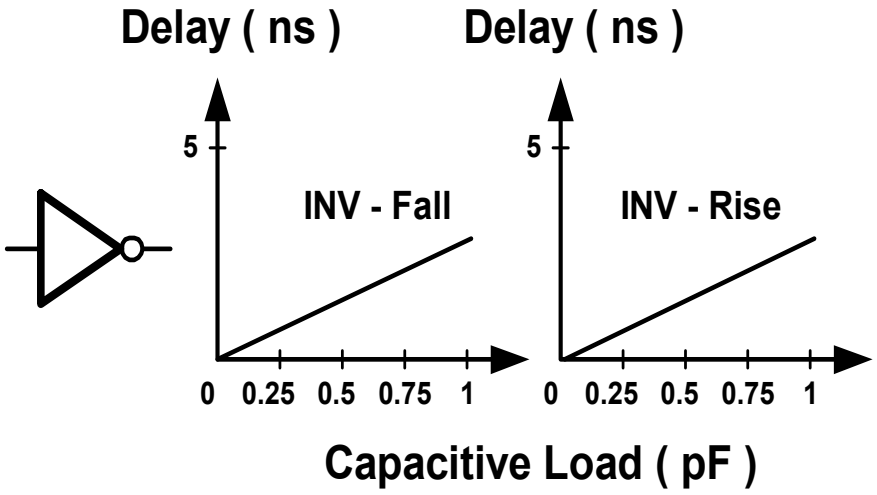
- PMOS substrate connected to the highest supply voltage V_{DD}
- NMOS substrate connected to the lowest supply voltage GND

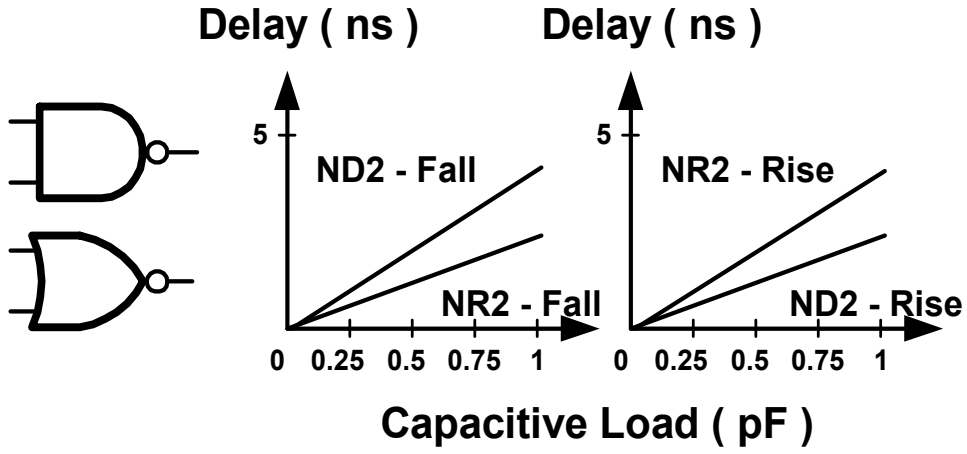




- Gate delay of logic gates
- A function of capacitive load , and logic gate types .



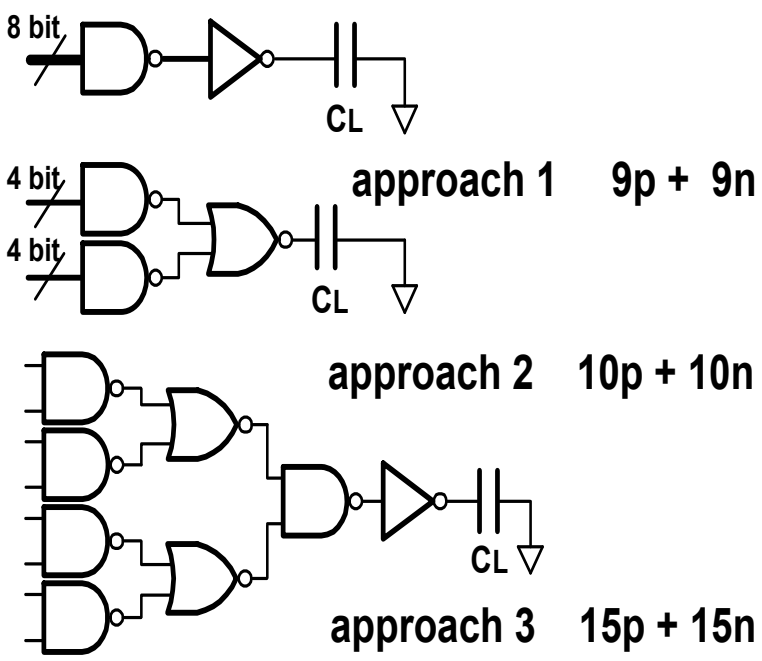




- A complex gate can be implemented by several gate

EXAMPLE

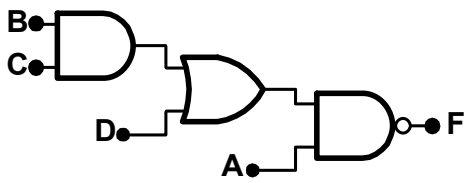
- As an example of a simple logic decision consider the implement of an 8-input AND gate driving 1pF load (for instance，a row decoder in a RAM or ROM)，we may use the following：
- Approach 1 – An 8-input NAND and an inverter
- Approach 2 – Two 4-input NAND and 2-input NOR
- Approach 3 – Four 2-input NAND two 2-input NOR，a 2-input NAND and an inverter。

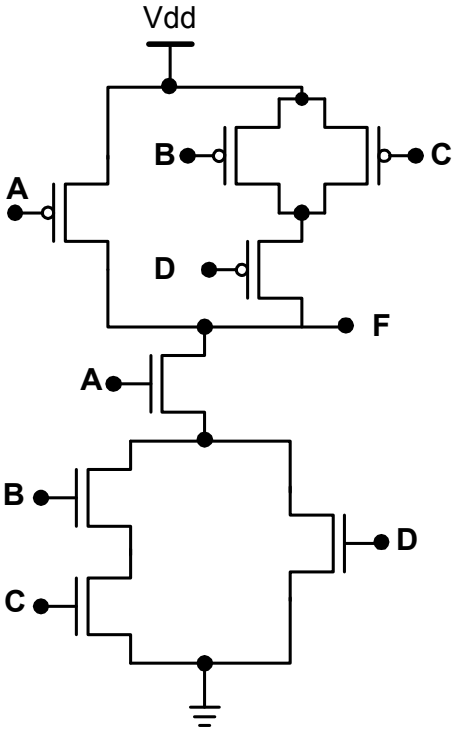


- Comparison of approaches to designing an 8-input AND gate

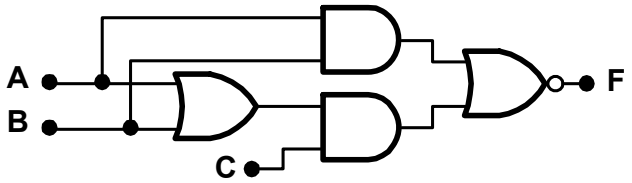
APPROACH	DELAY STAGE 1	DELAY STAGE 2	DELAY STAGE 3	DELAY STAGE 4	TOTAL DELAY （SPICE） ns
1	3.37	2.82			6.2
ND8->	ND8	INV			6.5
INV	Falling	Rising			
2	4.36	0.88			5.24
ND4->	ND4	ND2			5.26
NR2	Falling	Rising			
3	0.31	0.4	0.31	2.17	3.19
ND2->	ND2	NR2	ND2	INV	3.46
NR2->	Falling	Rising	Falling	Rising	
ND2->					
INV					

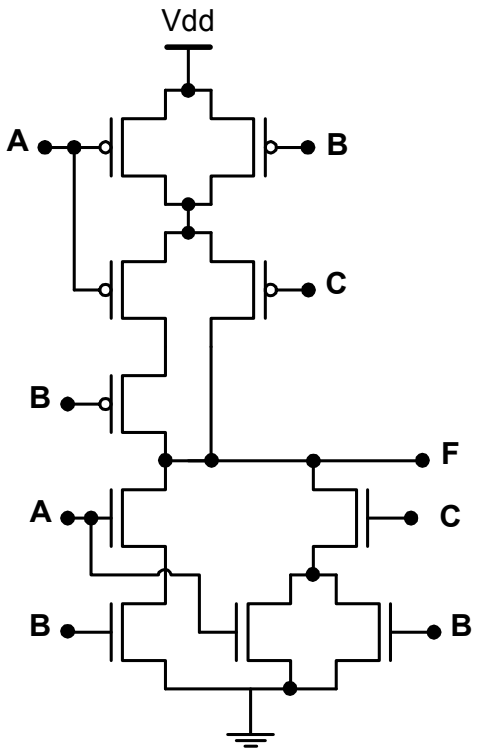
（D）Complex Combination Logic Gate





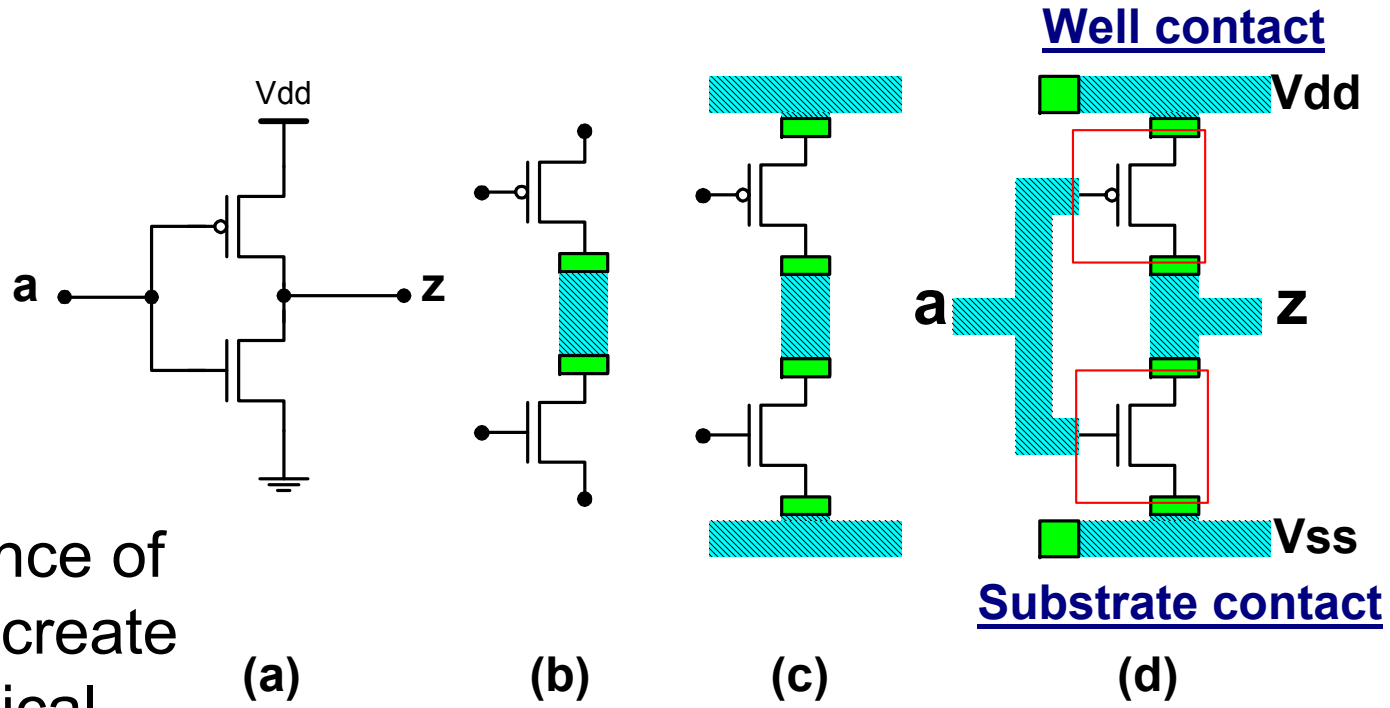
CMOS Circuit for $F = \overline{ABC + AD}$





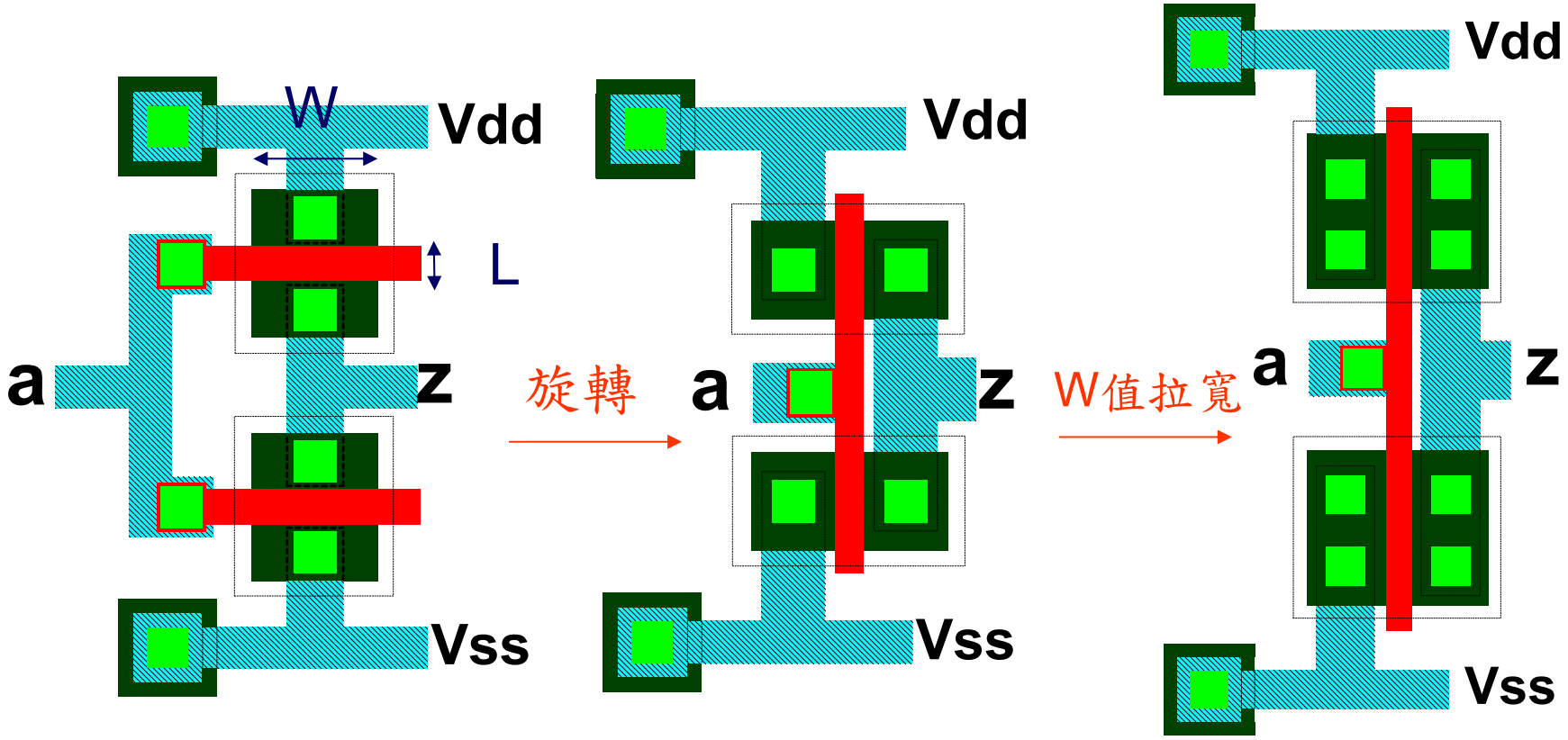
AOI CMOS Circuit for $F = \overline{AB + AC + BC}$

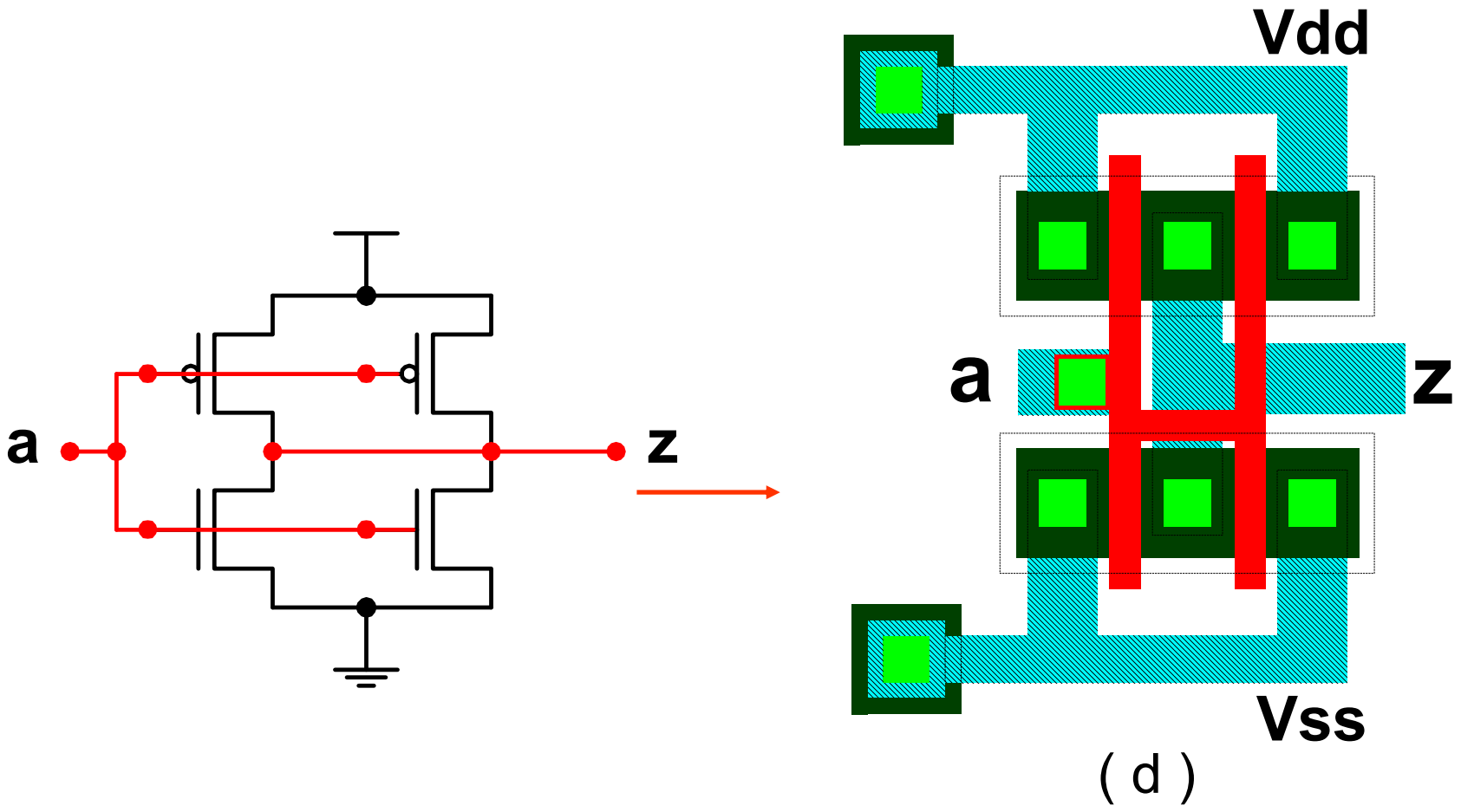
(E) Symbol Layout



A sequence of steps to create the physical layout of an inverter

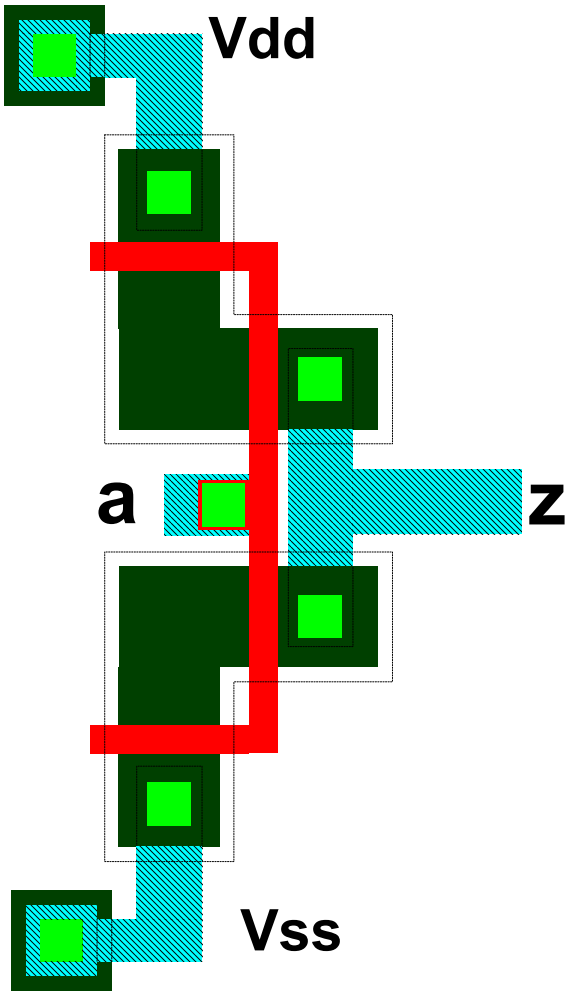
- Symbolic Layout for CMOS inverter



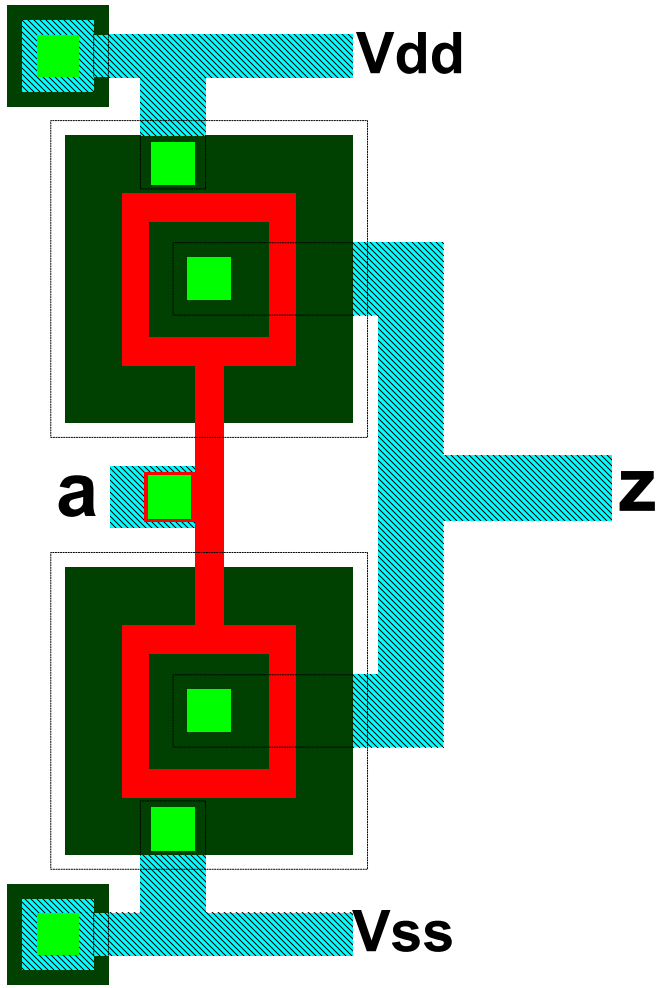


(d)

- Large inverter can be created by paralleling inverters

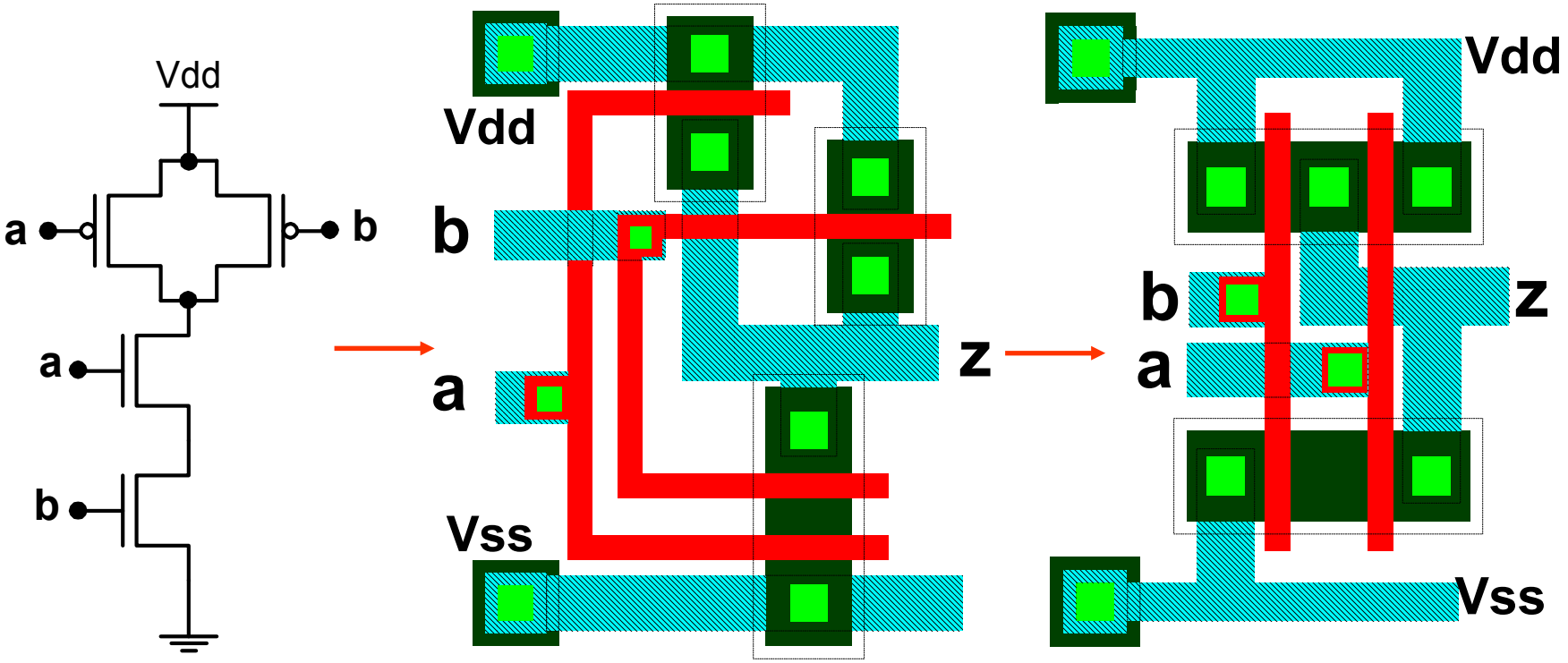


(e)

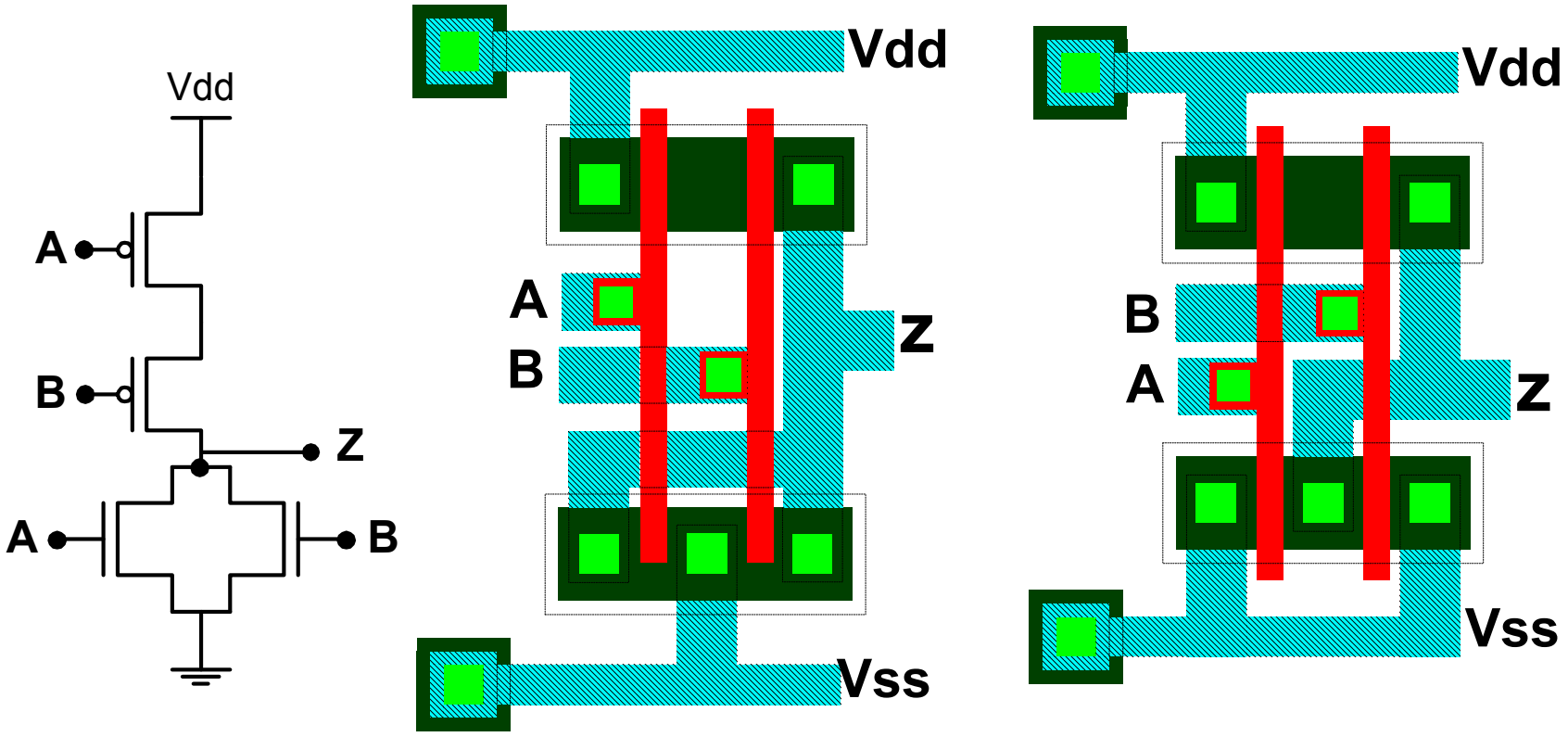


(f)

- Symbolic layouts for the CMOS NAND gate

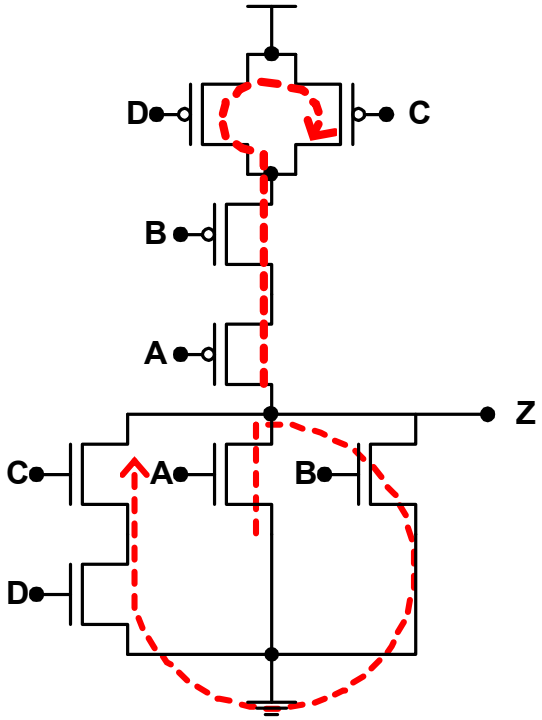


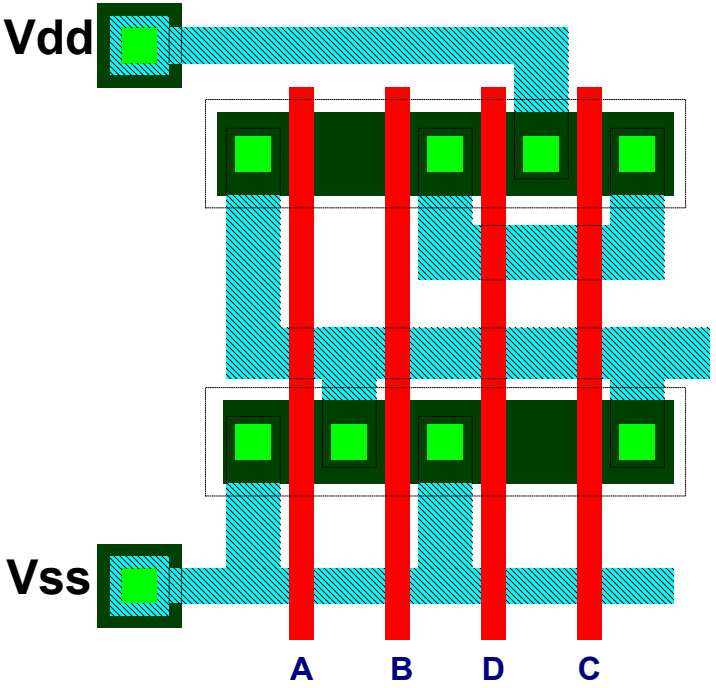
- Symbolic layouts for the CMOS NOR gate

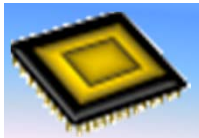


(F) Complex Logic Gate Layout

- The PMOS and NMOS connection can be represented as an P-graph and N-graph .
- Each edge of the graph represents a MOS .



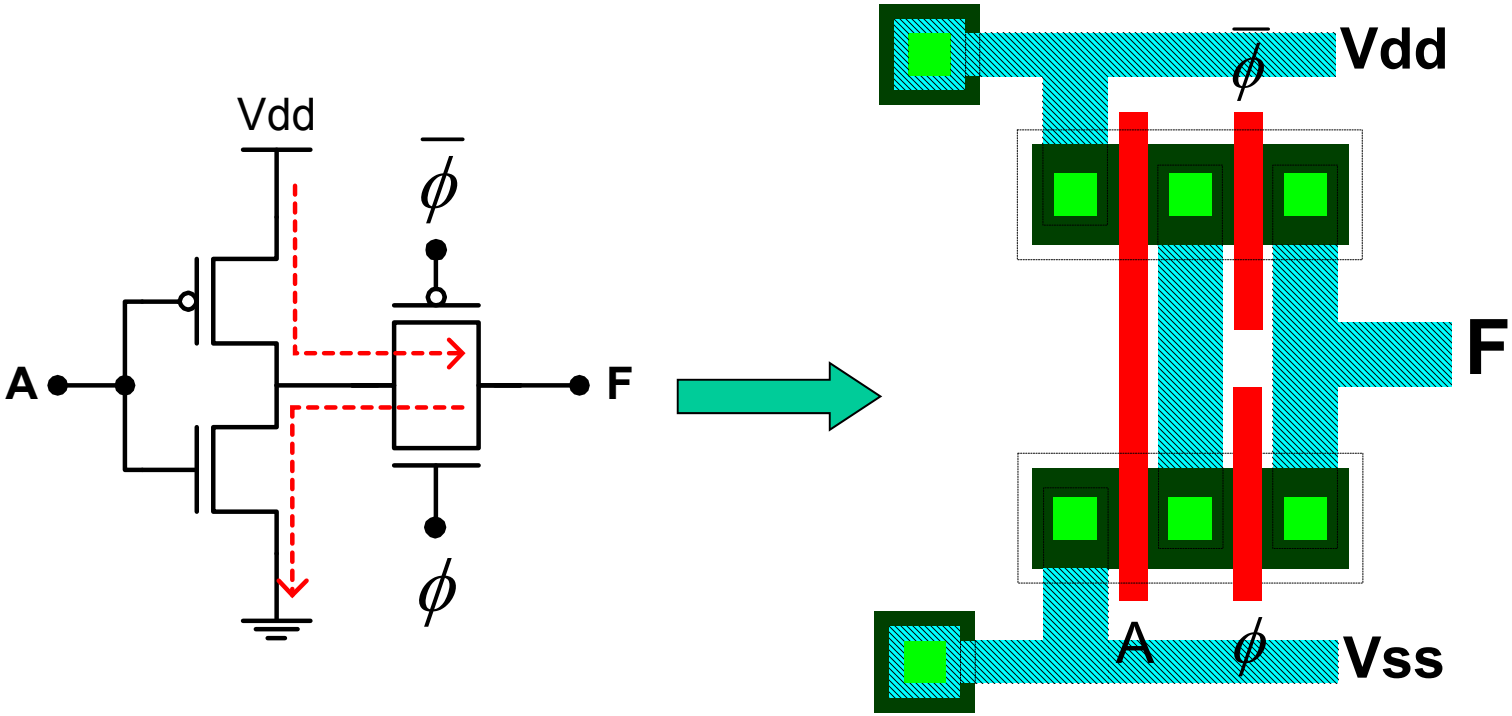




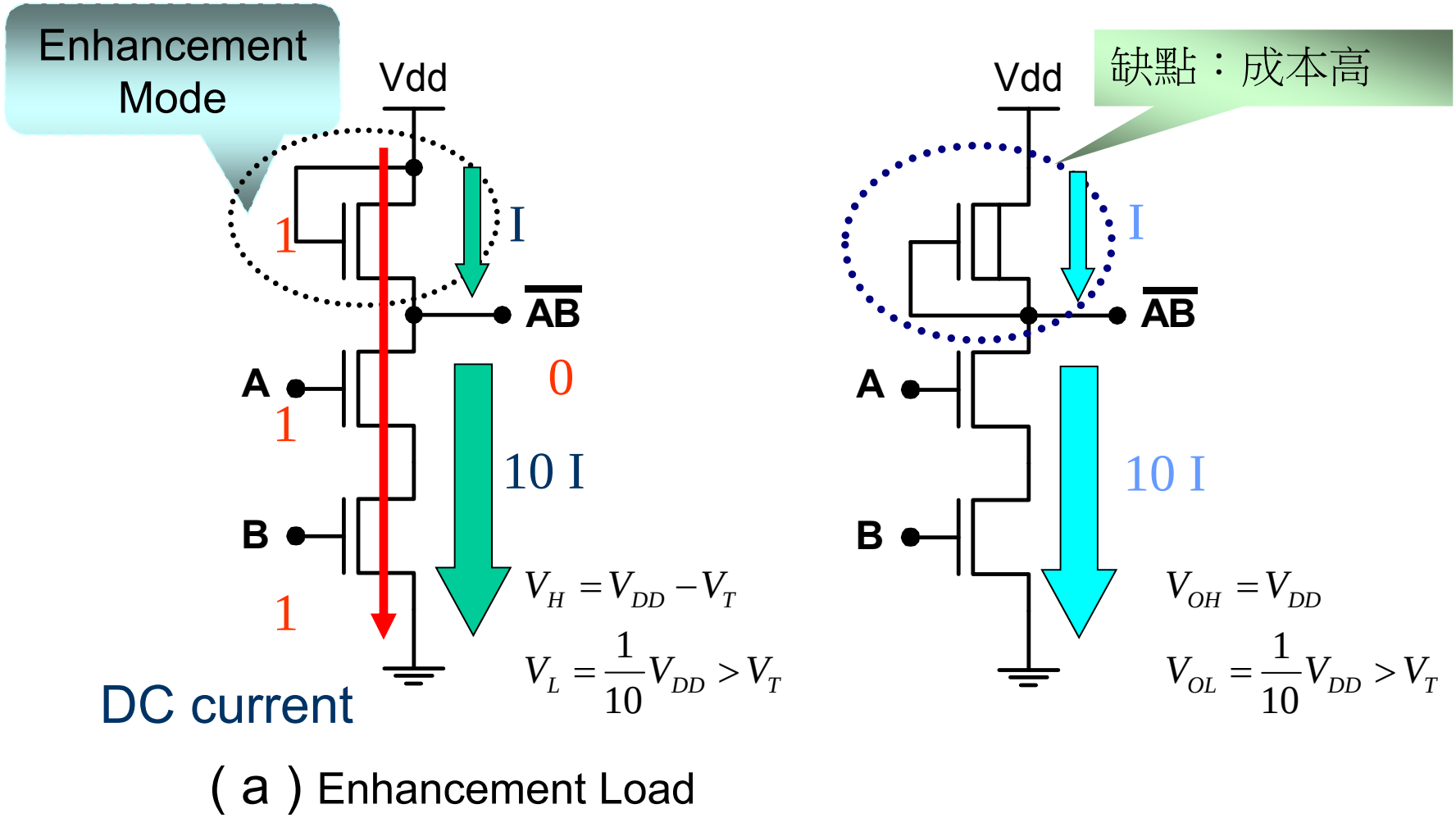
Chapter 5

- Euler path：A sequence of edges in the graph .
- The gate in the Euler path may share a common source-drain connection and may be connected together with no breaks .
- The algorithm are as follows
  1. Find all Euler paths that cover the graph .
  2. Find a p- and n- Euler path that have identical labeling (a labeling is an ordering of gate labels on each vertex) .
  3. If the paths in step 2 are not found , then break the gate in the minimum numbers of places to achieve step 2 by separate Euler paths .

Example：



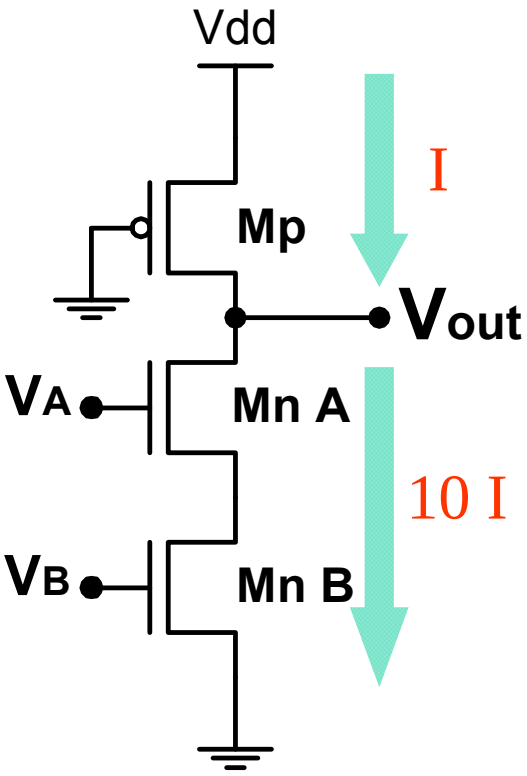
5.1.2 NMOS Logic



(a) Enhancement Load

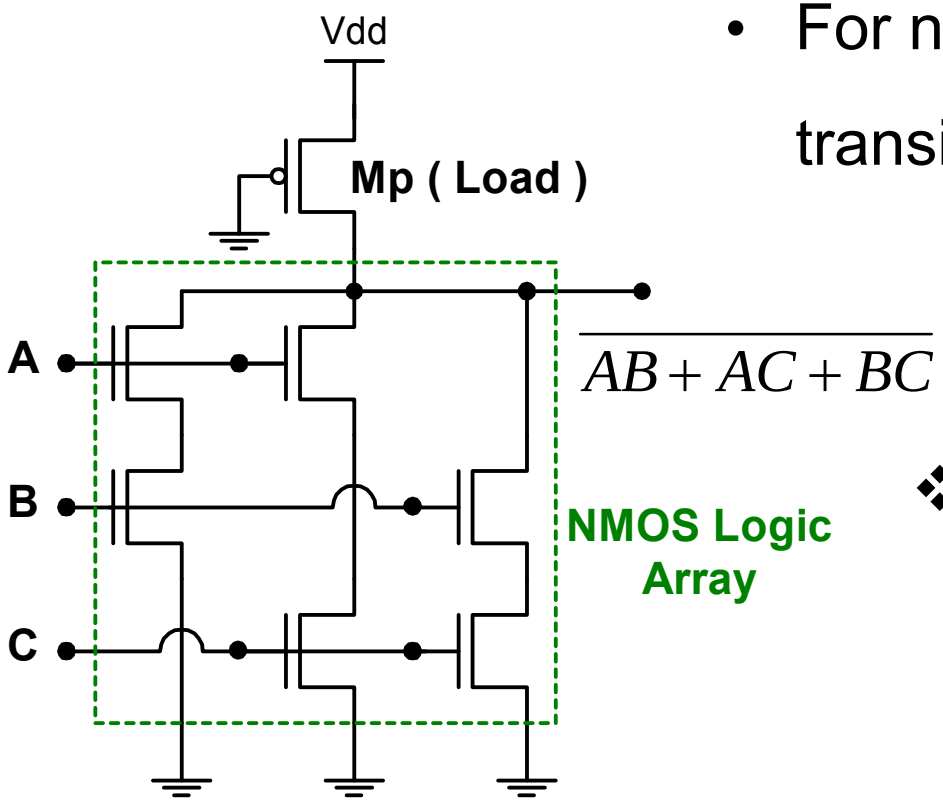
5.1.3 Pseudo NMOS Logic

(a) Pseudo-NMOS NAND2 circuit



- Less Area
- Low cap Load (only NMOS)
- Speed is compatible (or slower) than static CMOS
- RATIO LOGIC：pull up current is always flow even the output logic value is “ 0 ” .
- DC power dissipation

(b) Pseudo-NMOS Logic Array



$$\overline{AB + AC + BC}$$

- For n-input gate , need n+1 MOS transistor

❖Design Pseudo-NMOS logic must consider：

$$\frac{\mu_p \times \frac{W_p}{L_p}}{\mu_n \times \frac{W_n}{L_n * n}} = \frac{1}{10}$$

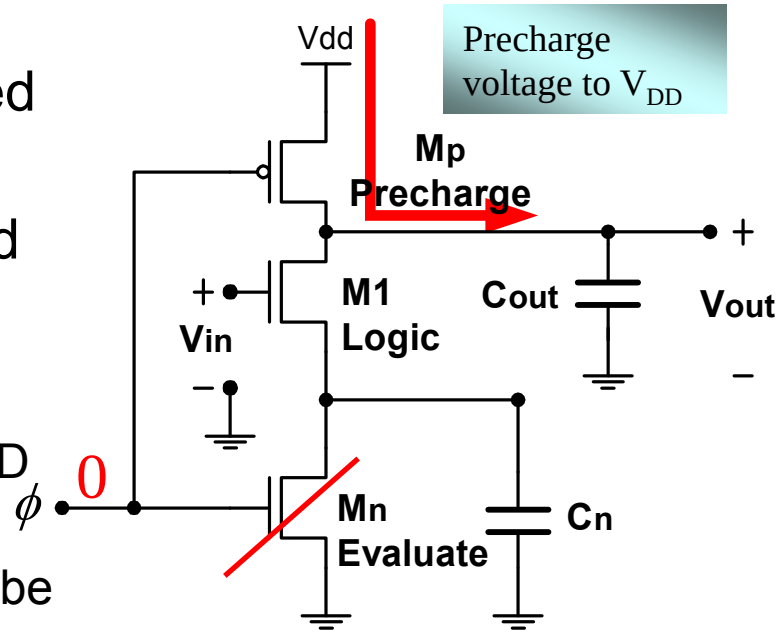
串聯個數

5.2 Dynamic Logic Circuit

(A) *Basic Concept*：

- As mentioned previously , the static logics have a pull-up and pull-down network . However they are activated at different times .
- The pull-up network can be replaced by a precharge PMOS M_P .
- **Advantages**：
  - i. No direct current path from V_{DD} to GND ➔No DC power dissipation .
  - ii. Speed is enhanced . (transistors can be made wider and input never cause the output pull-up .)

❖ **Note**：input must be made glitch free .

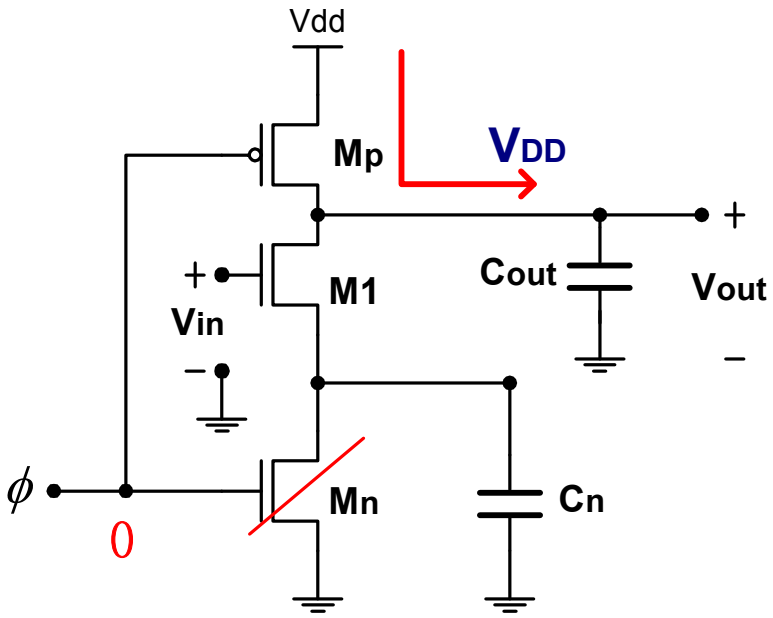


Basic Dynamic NMOS inverter

(B) Logic operations：

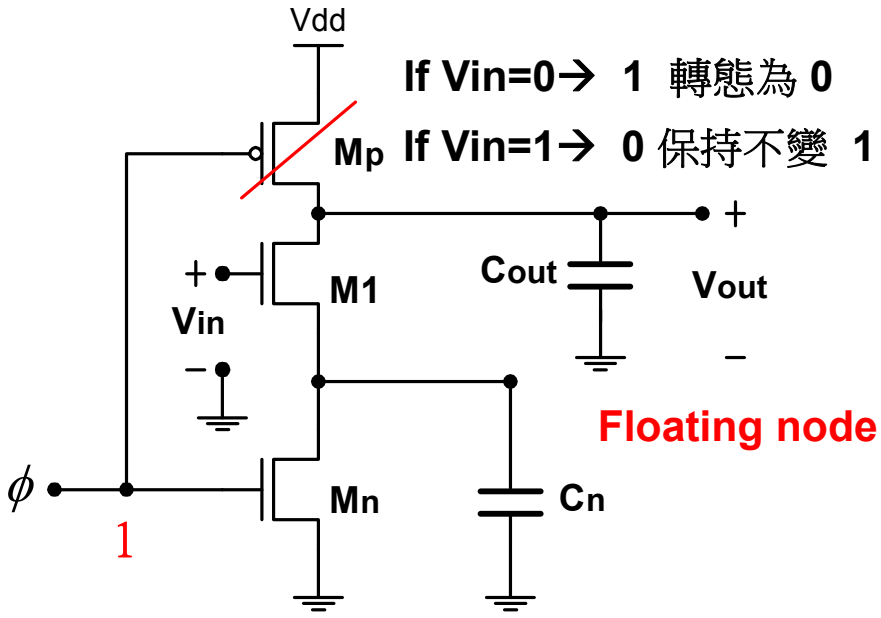
Two phase of operations controlled by clock ϕ

ϕ = 0：precharge
Vout = VDD

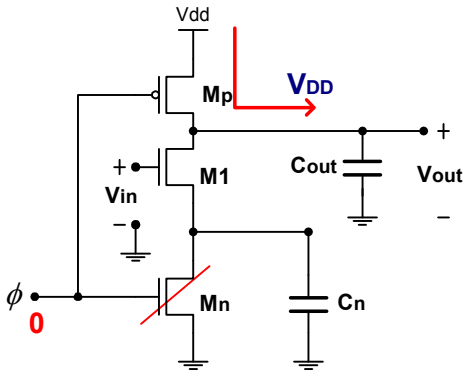


PRECHARGE

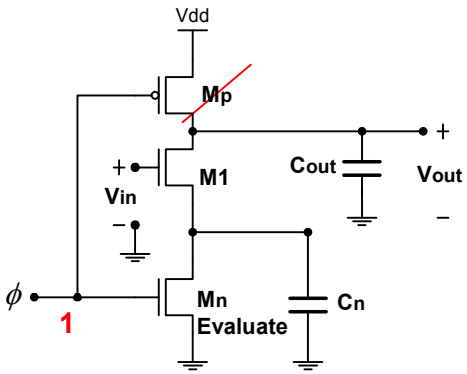
ϕ = 1：Evaluate
Vout：conditional discharge



EVALUATE
(Conditional Discharge)

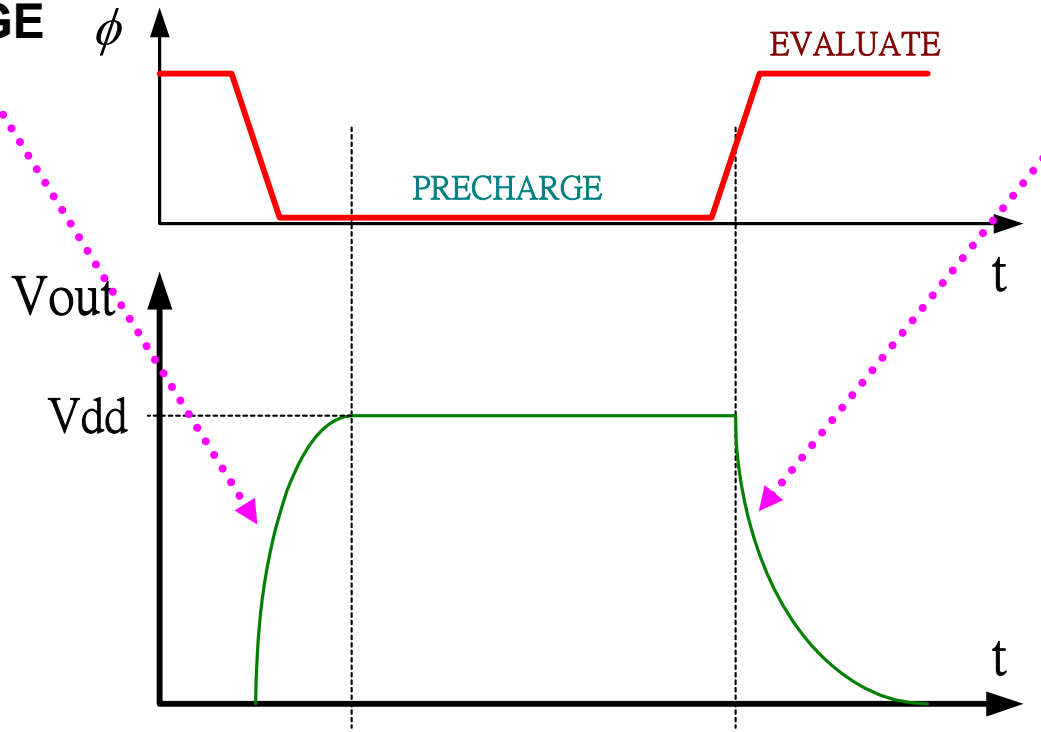


PRECHARGE

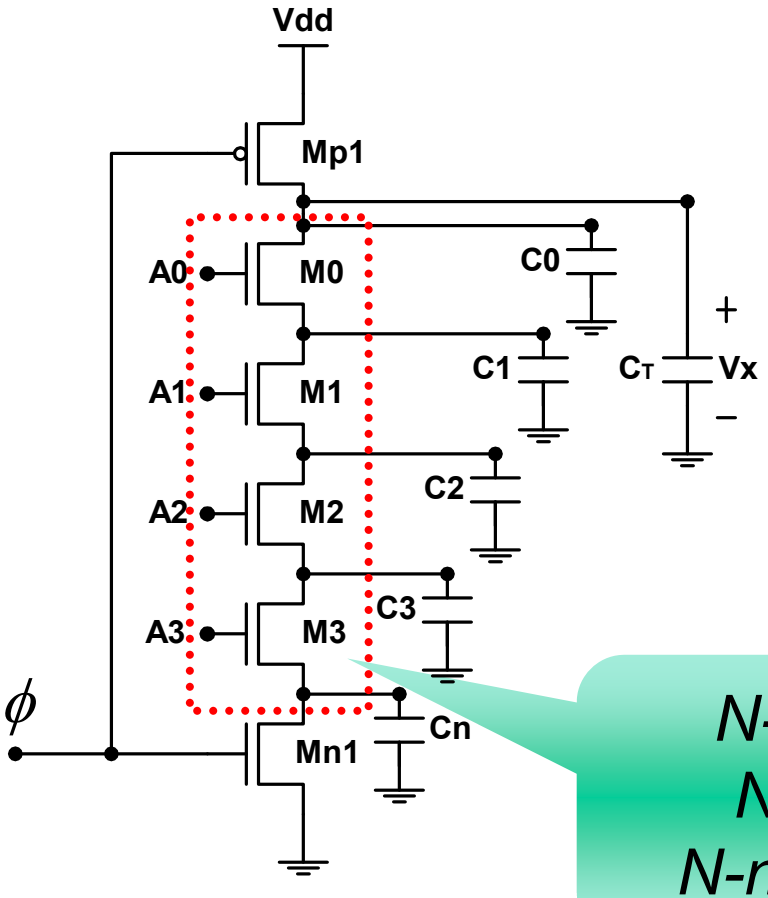


EVALUATE

(Conditional Discharge)



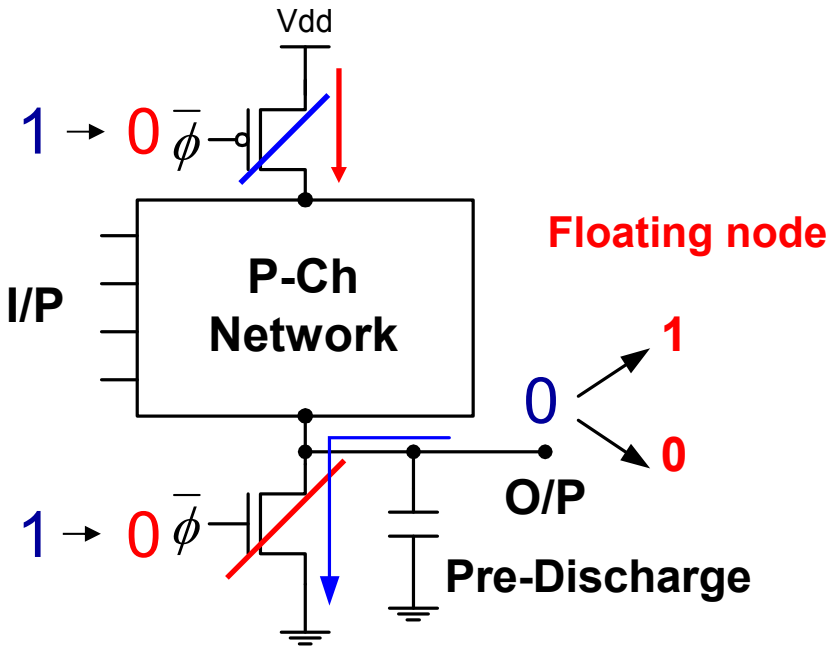
(C) N-Rich Dynamic Logic :



- C_T precharged to V_{DD} during $\phi = 0$ (precharge)
- N-channel logic block stacked with a clocked NMOS may discharge the output to GND during $\phi = 1$ (evaluation)

N-block
N-tree
N-network

(D) P-Rich Dynamic Logic :

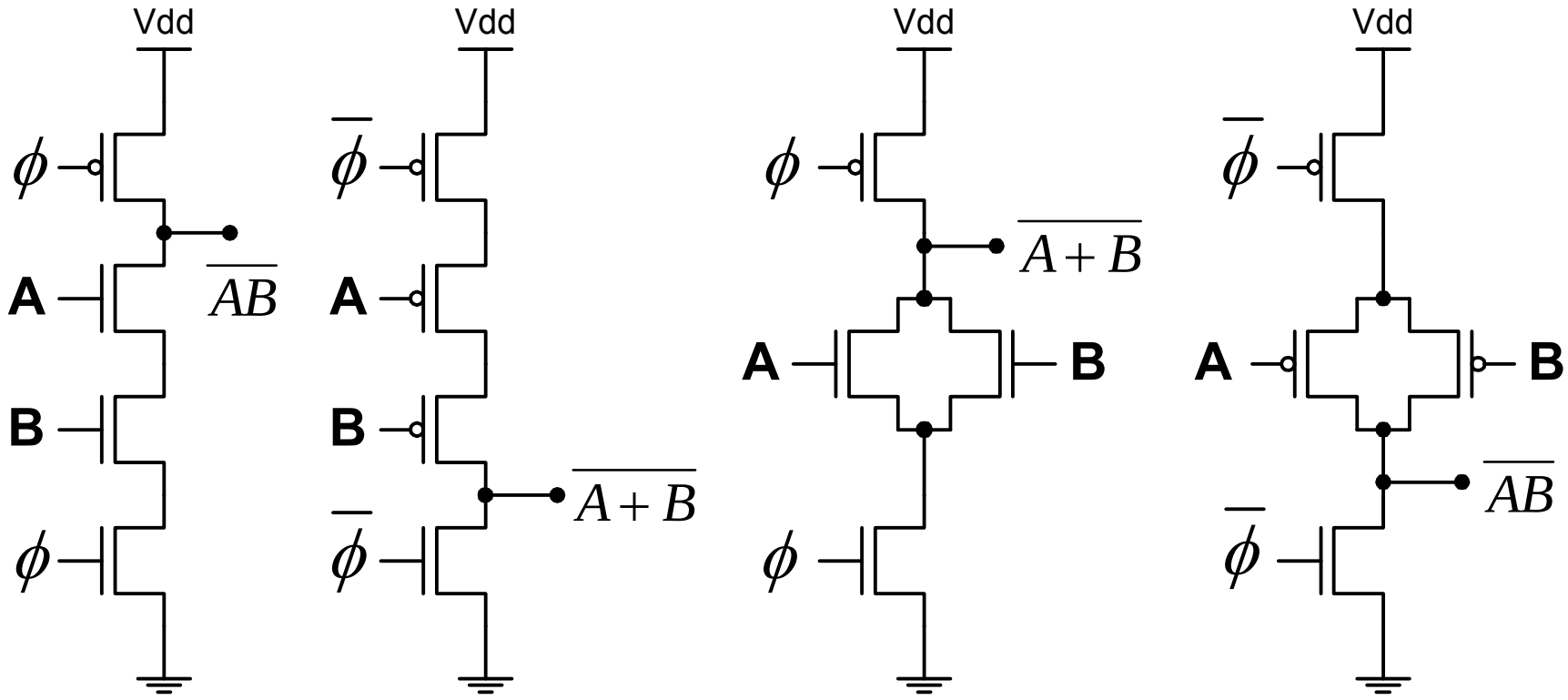


$\phi = 0(\overline{\phi} = 1)$: Pre-Discharge
O/P precharge to GND .

$\phi = 1(\overline{\phi} = 0)$: Evaluation
P-Ch network may precharge the O/P to V_{DD} .

Example：$\phi = 0(\overline{\phi} = 1)$ **Precharge**

$\phi = 1(\overline{\phi} = 0)$ **Evaluation**

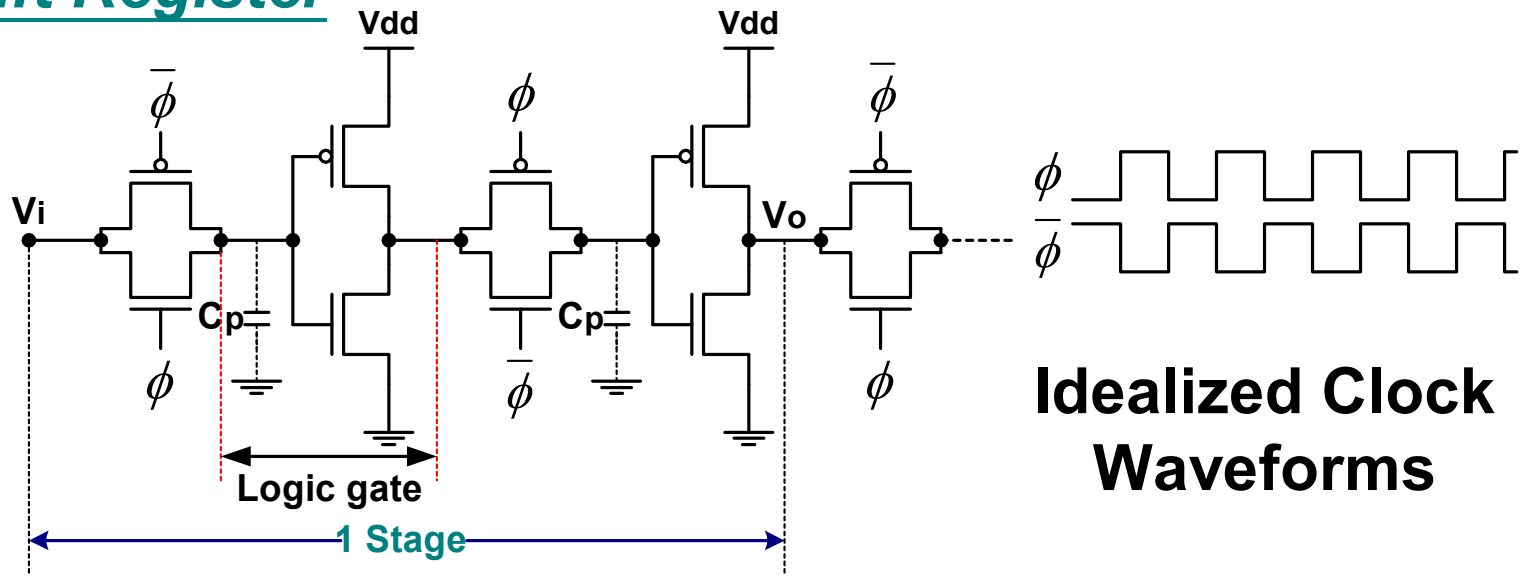


5.2.1 Clock static CMOS Logic

* Static holding , dynamic shifting

(A) Pseudo 1-phase (or 2-phase) Logic

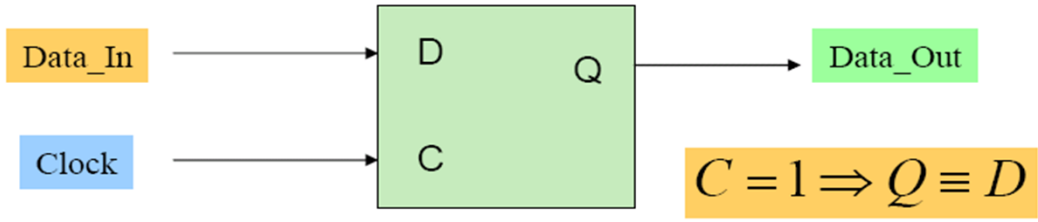
- **Shift Register**

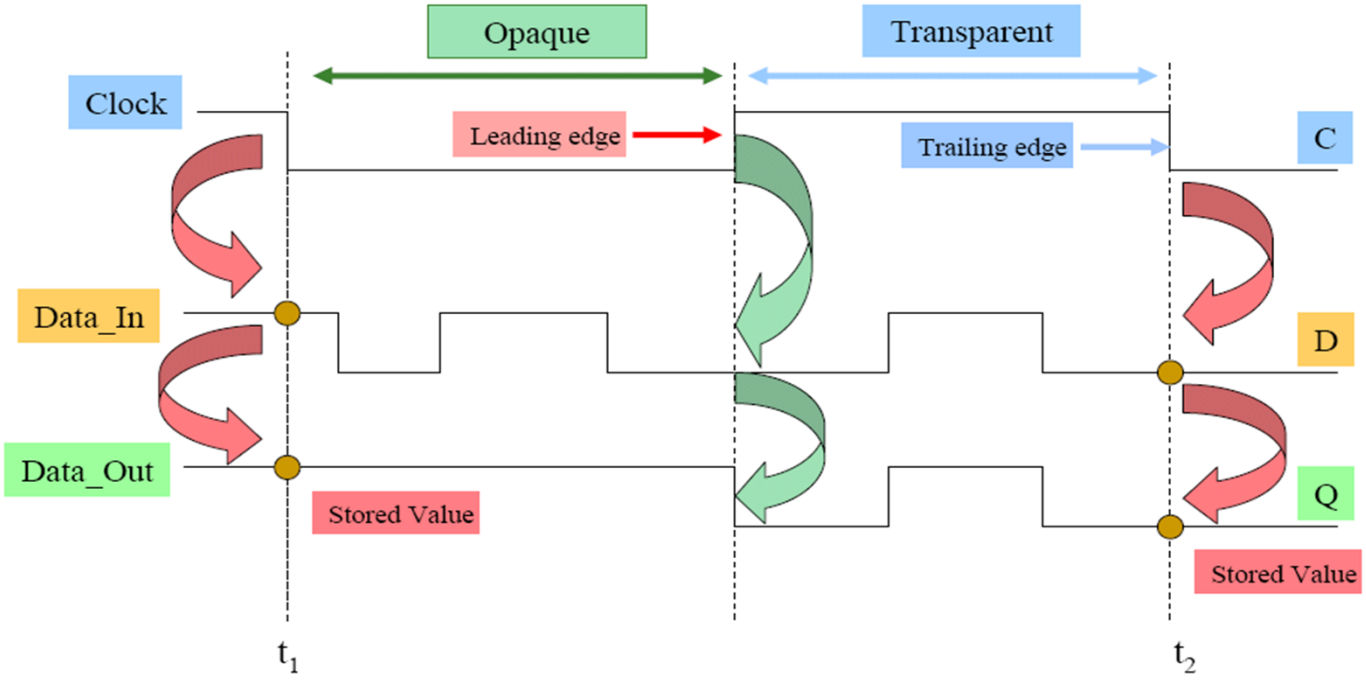


- Data is held in the input capacitance Cp of the inverter (or static logic gate) temporarily , Not Static Holding .
- Not race free if clock skew exists .
- Simpler structure than the fully static shift register .

Latch Concept

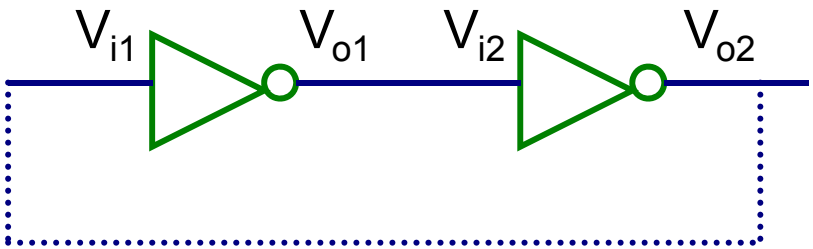
Positive-Polarity Latch

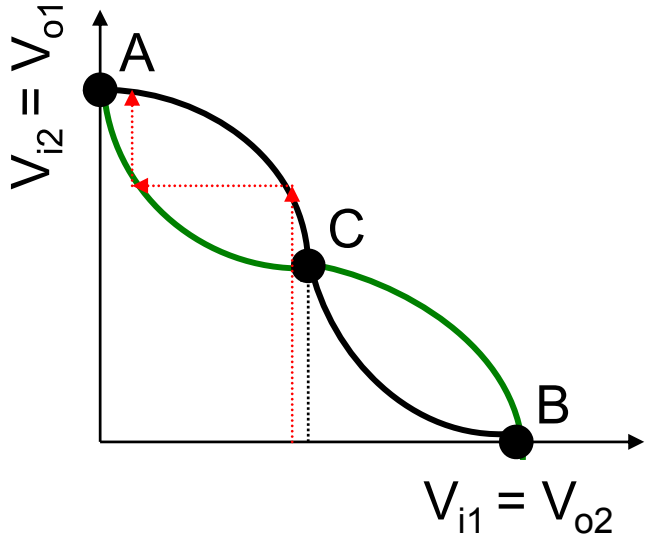




Latch—Bistable latch

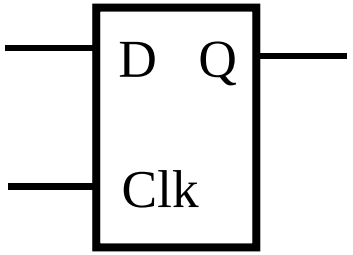
- The cross-coupling of two inverters
  - bistable circuit
  - 0 or 1

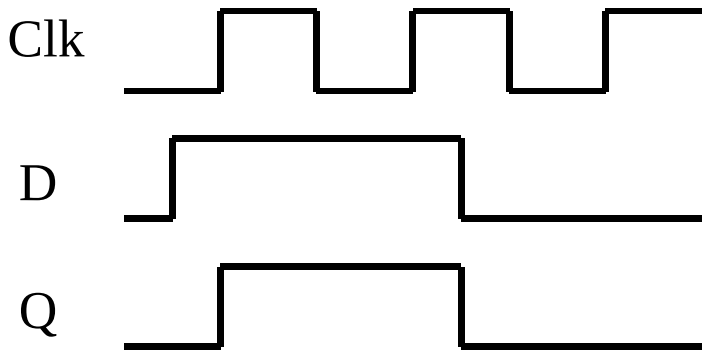




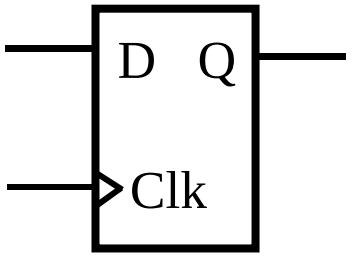
Latch VS. FlipFlop

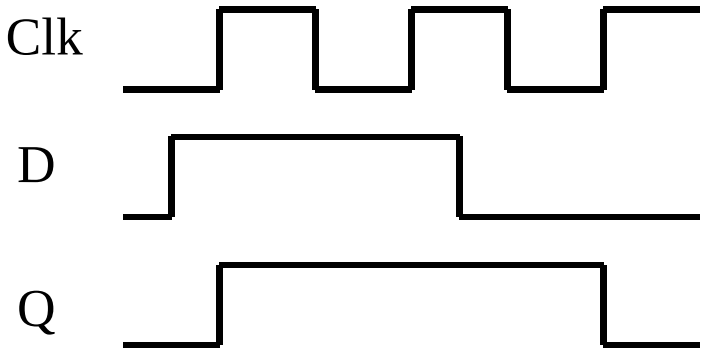
- Latch stores data when clock is low



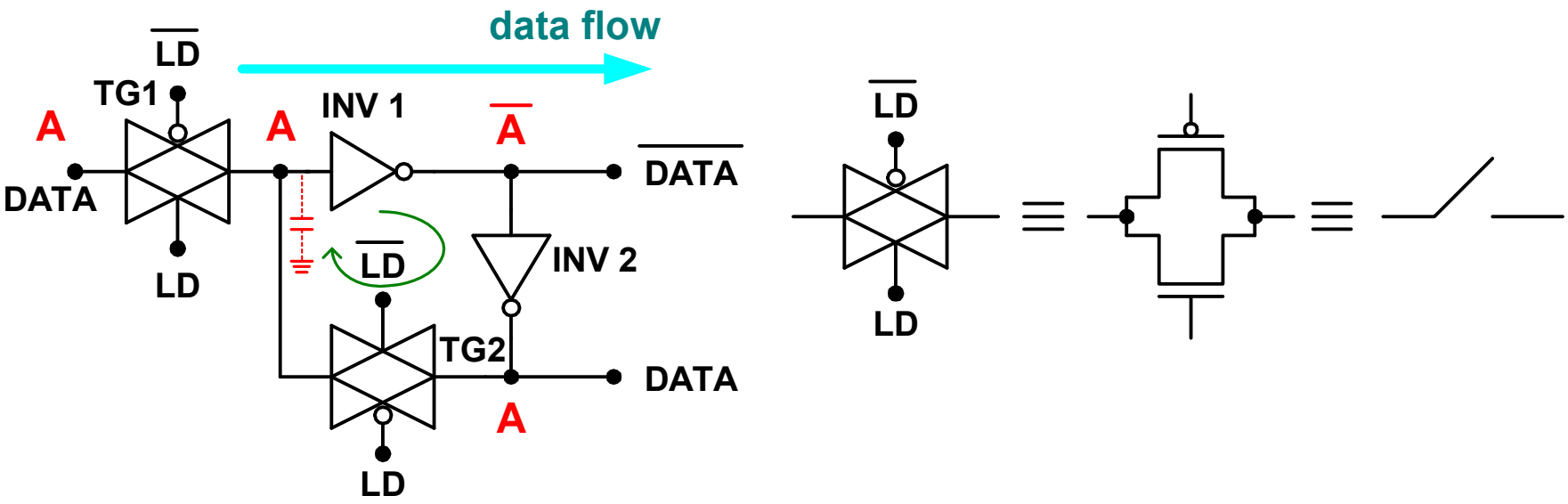


- Register stores data when clock rises



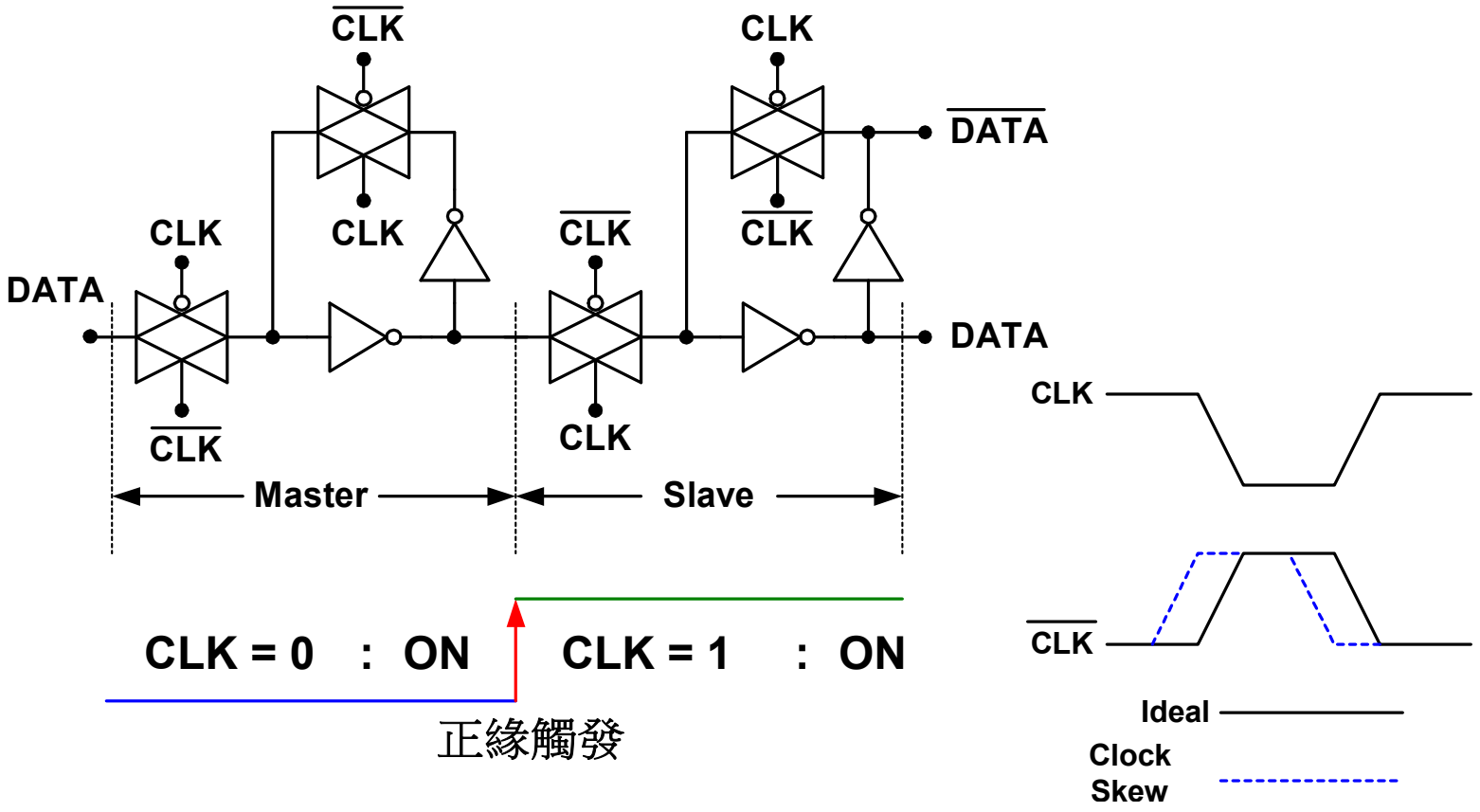


Basic Transmission Gates Latch

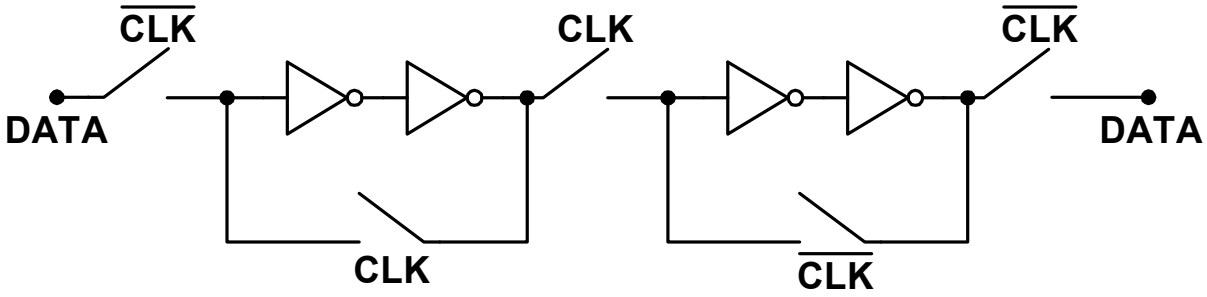


- TG2 is in the feedback loop
- Load signal LD=1 , TG1 allows data enter the latch , TG2 is open .
  LD=0 , TG1 is open , TG2 provides feedback .

Master - Slave D type Flip Flop



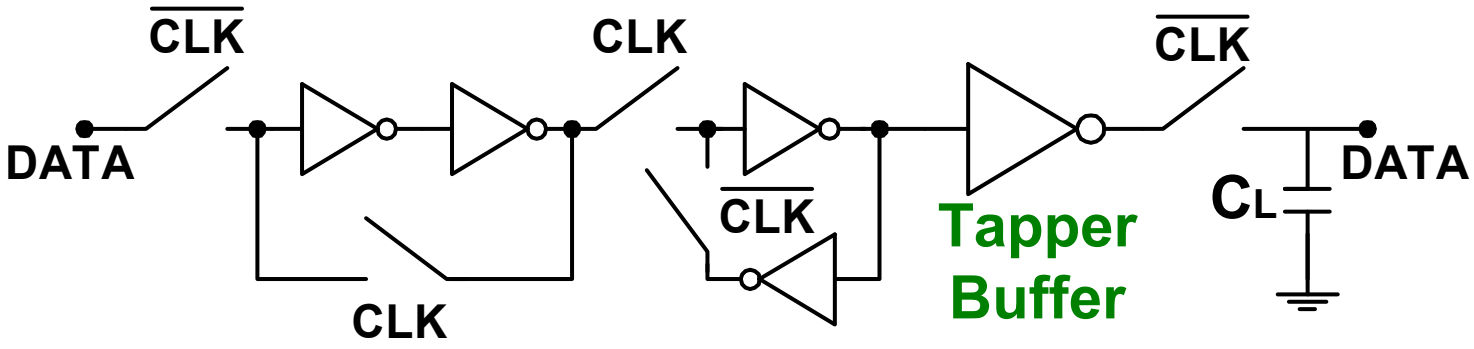
- CLK=0 , DATA pass the master stage and previous data is latched by the slave stage .
- CLK=1 , DATA latched by the master stage and sent into the slave stage .
- Could race if CLK and $\overline{CLK}$ overlaps ➔ Both master and slave are turned on .



→ **Clock skew race problem**

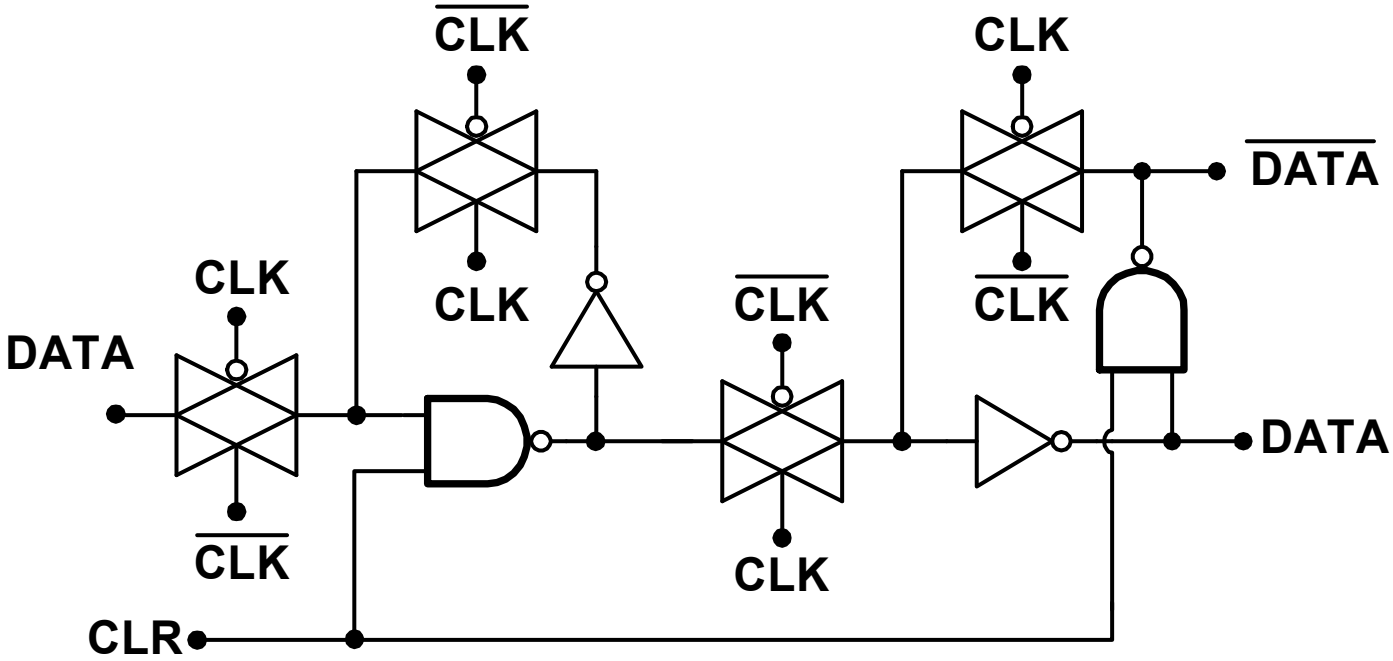
Master - Slave D type Flip Flop Improved Version

- Improve latch stability



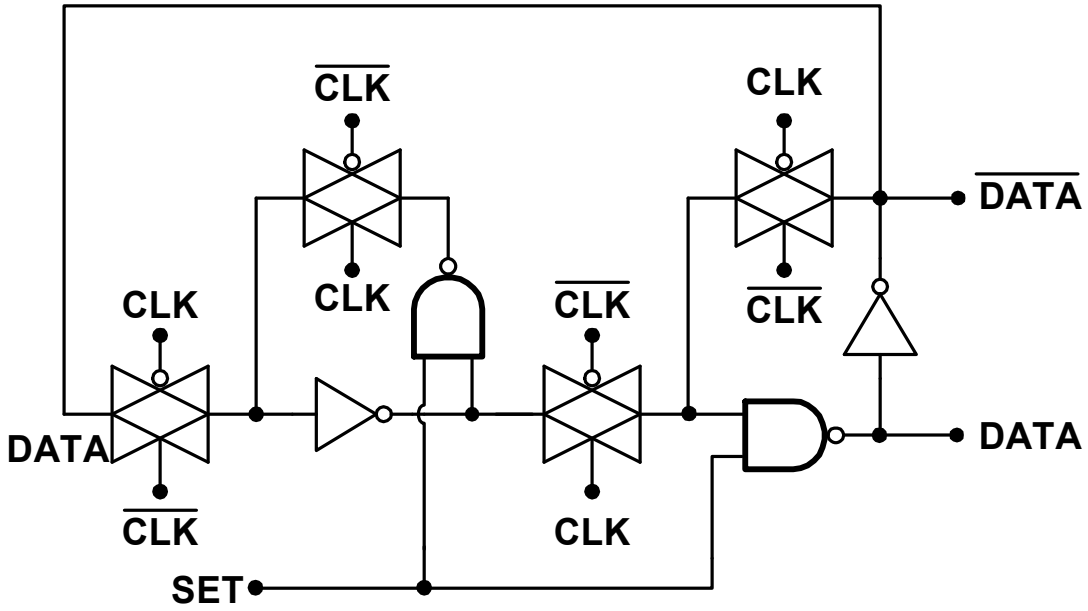
* Clock skew → race

Master - Slave D-type F/F with Clear



- **CLR=0：Clear data , DATA=0 ,** $\overline{\mathrm{DATA}} = 1$

Toggle Flip-Flop



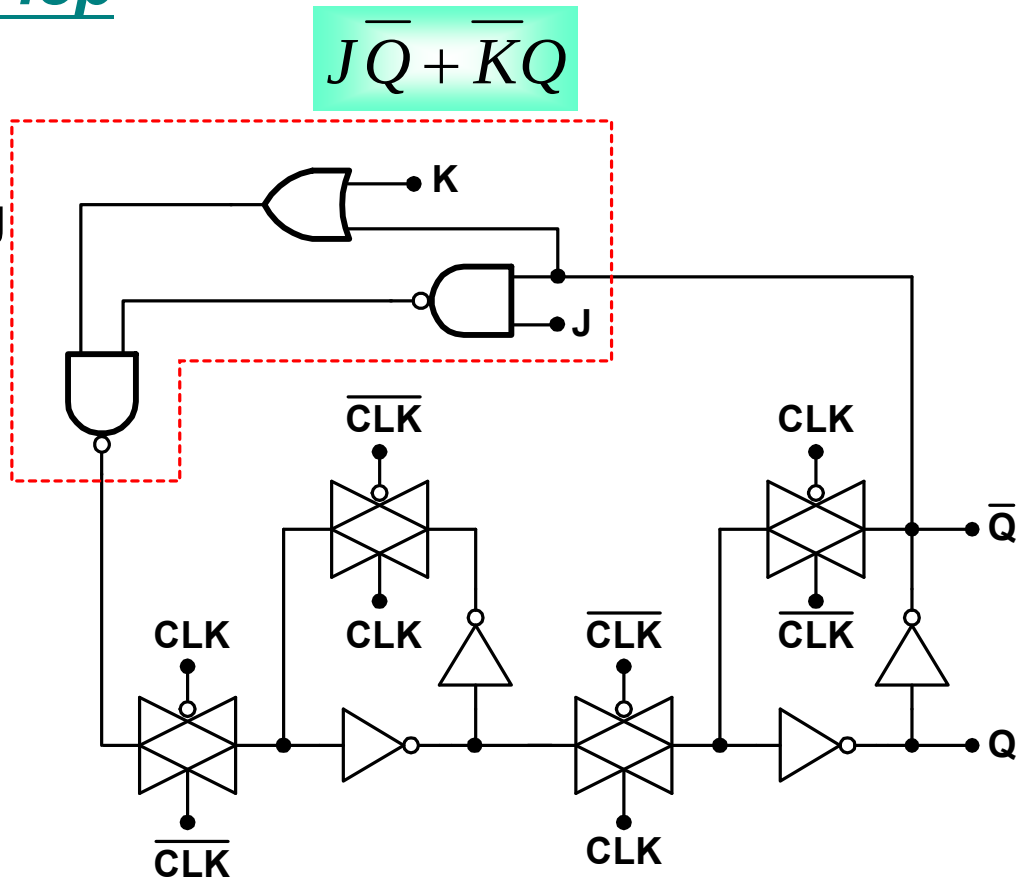
- DATA (t) = $\overline{DATA}(t-1)$
- Operation is controlled by CLK and $\overline{CLK}$
- SET allows an initial state set
  SET=0：Set data DATA=1，$\overline{DATA}=0$

JK Master-Slave Flip-Flop

- JK F/F is obtained by combining the input with feedback

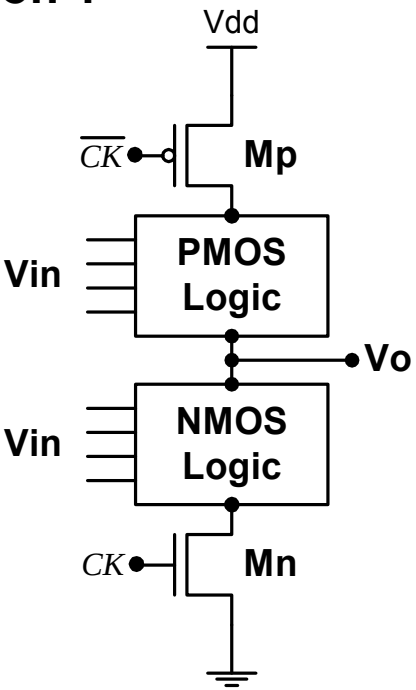
$$Q(t+1) = J\overline{Q} + \overline{K}Q$$

J	K	$Q(t+1)$
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

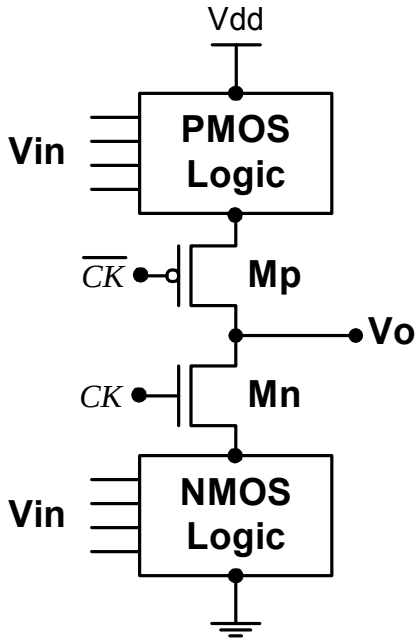


(B) C²-MOS (Clocked CMOS) Logic

Version 1：

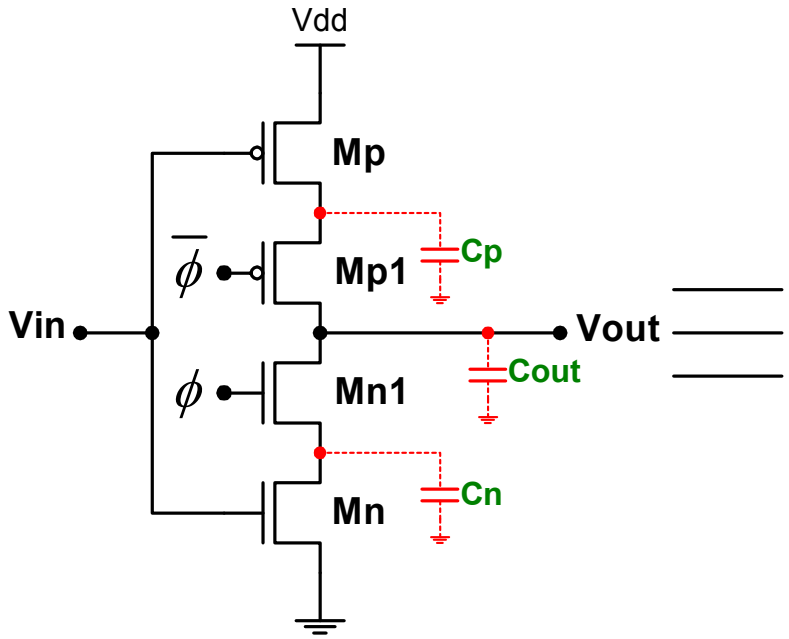


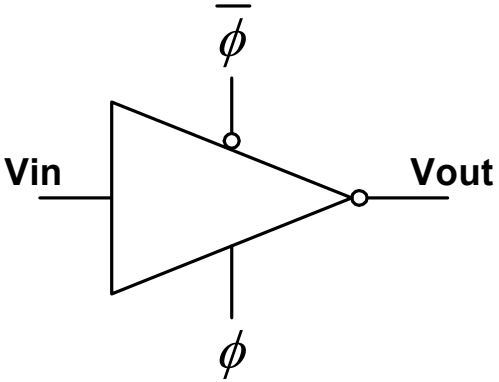
Version 2：

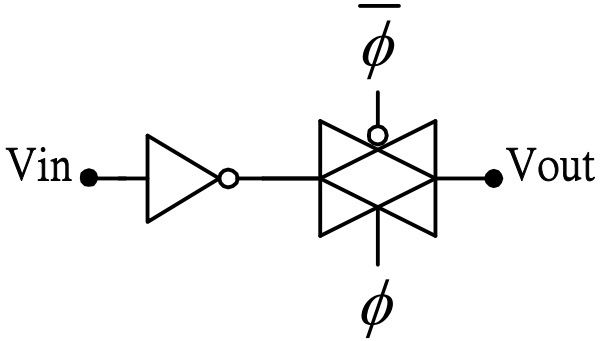


- CK=1：Active；CK=0：Disable
- Mp , Mn Switching or isolating devices
- Suitable for FF construct

C^2-MOS Inverter (Latch)





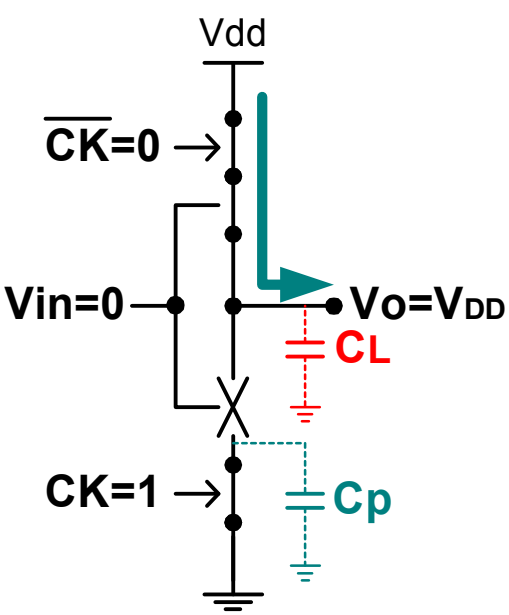


- C^2-MOS Latch has better clock skew race immunity than transmission gate latch
- (Homework：shown by SPICE)

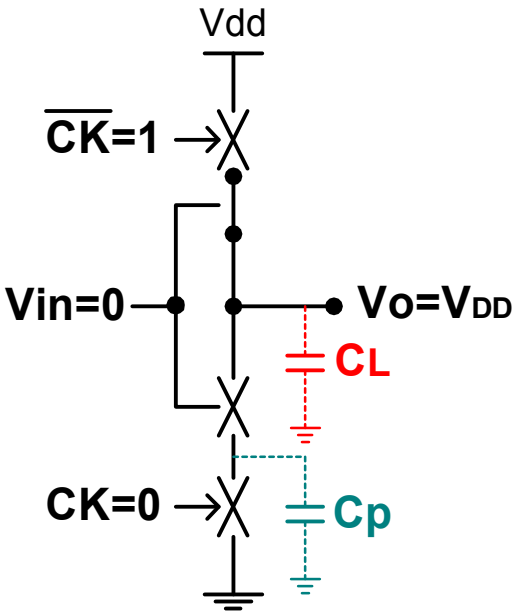
Charge Redistribution Problem (1/2)

Version 1 :

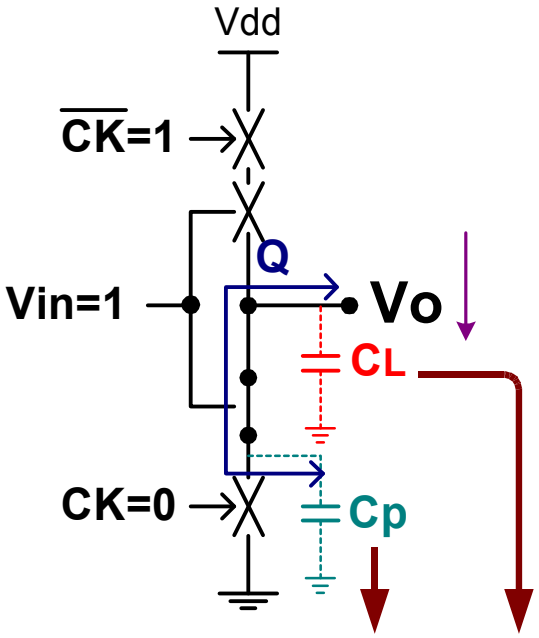
Active



Disable



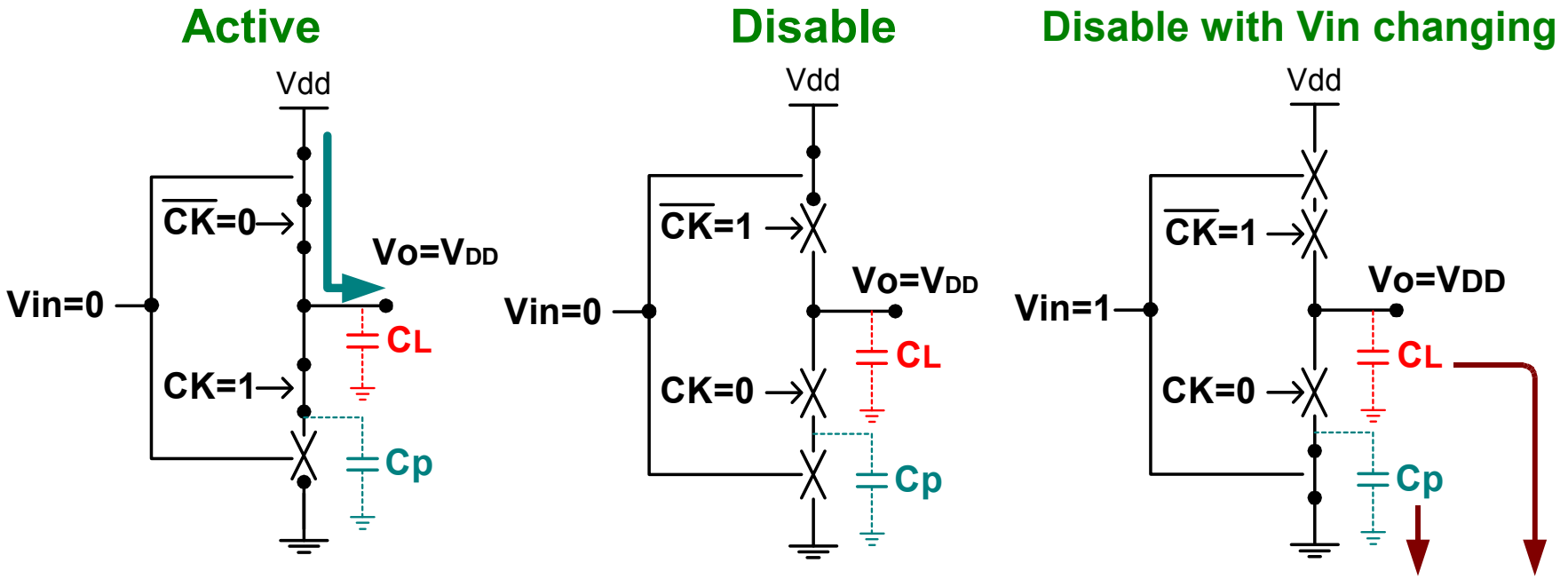
Disable with Vin changing



Charge Redistribution

Charge Redistribution Problem (2/2)

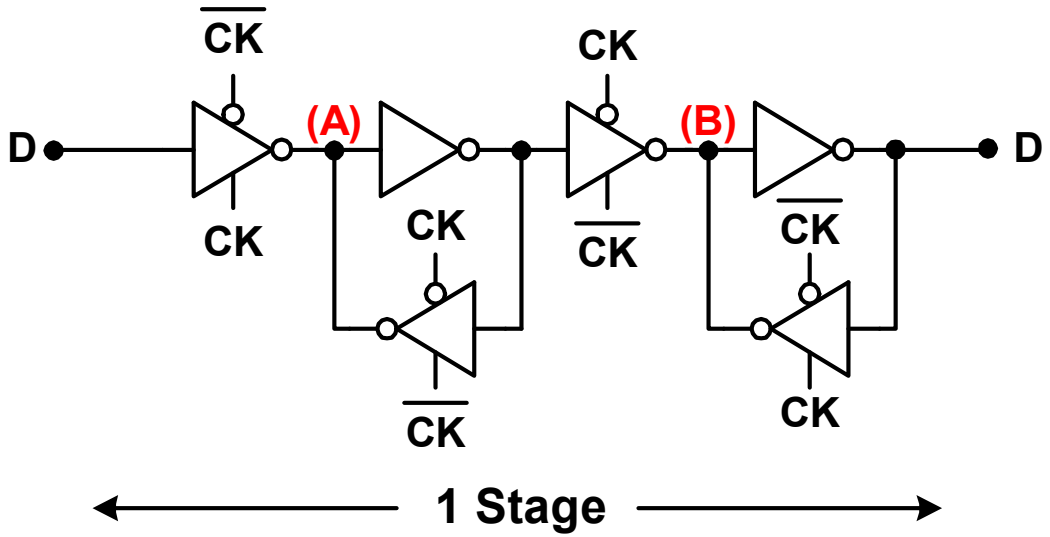
Version 2 :



No Charge Redistribution

- Version 1 has the charge redistribution problem due to incomplete output isolation
- For safety design , Version 2 is recommended

D-type F/F



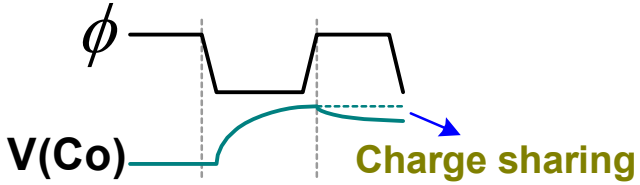
- Node (A) and (B) are sensitive nodes . If charge redistribution occurs at these two nodes before latch forms
  ➔ Error generation.

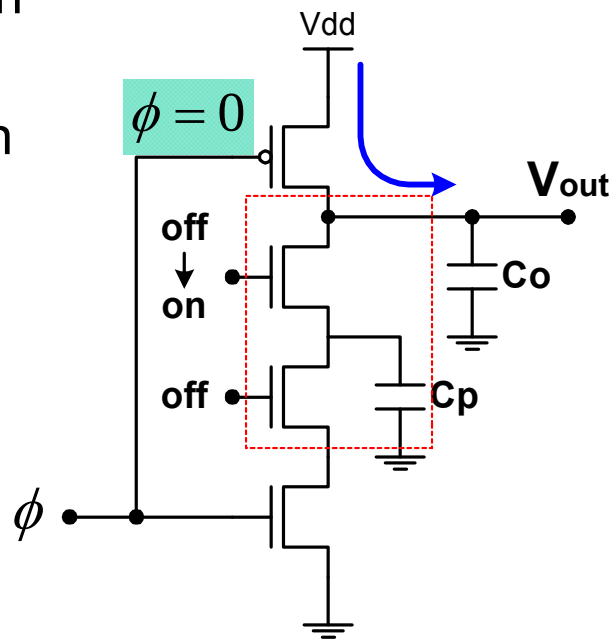
5.2.2 Problems of Dynamic CMOS Logic

- Because of precharging at the output node, dynamic CMOS techniques are common to the following Limitations

(A) Charge Redistribution：(Charge Sharing)

- The O/P signal of dynamic CMOS logic rely on the capacitance storage
- As Input signals variation charge redistribution between output capacitance C_0 and parasitic capacitance Cp . V_{out} lose it's value





(B) Leakage Current and Noise Margin

- Due to leakage of storage node , power supply variation
- Input may be altered from ideal 0 and V_{DD}

  ➔ NMOS logic section input may driven to weak inversion

  ➔ Leakage
- Serious problem in low frequency operation

  ➔ Dynamic ckt requires a lower bound on the speed of operation frequency

(C) Internal Delay Problem (Internal Race)

1. During CK=0 (Precharge)
   N1 and N2 are charged to V_{DD}
2. During CK=1 (Evaluation)
   Internal delay in gate1 will cause N1 from 1→0

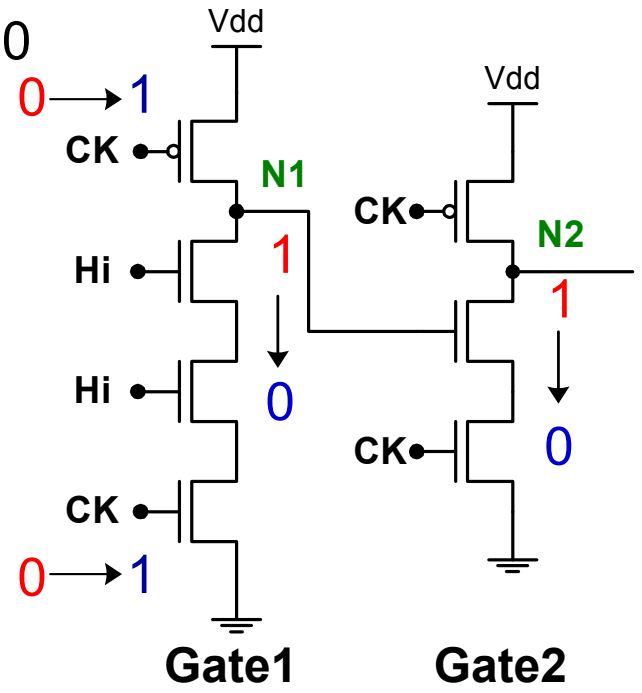
➔ The delay time can cause an incorrect discharge of N2

∵ During CK=1 and while N1 is "1" , there is a direct path from N2 to GND

➔ Precharged information in N2 is loss by this internal delay of gate1

❖ Solved by using :

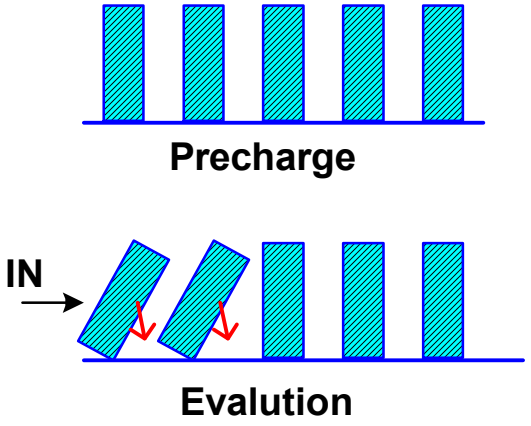
- Domino Logic
- N-P pair
- Zipper CMOS Logic
- 4-phase dynamic logic

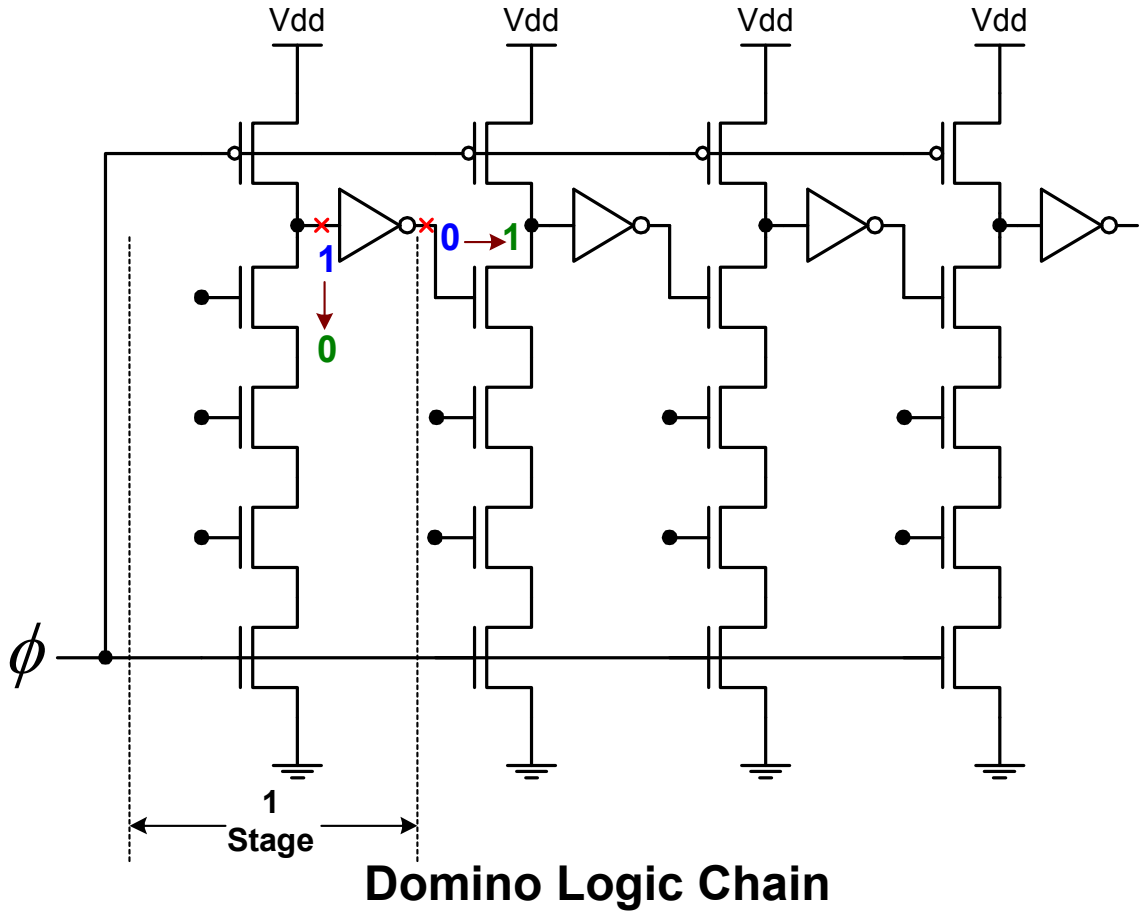


5.2.3 CMOS Domino Circuit [Bell lab] (1/3)

Ref：R.H. Krambeck , C.M. Lee , H.F. Stephen Law " How speed Compact Circuits with CMOS " IEEE JSSC , Vol. sc-17 , No.3 P.P.614 ~ 619 , 1982

- Basic Cell：





Domino Logic Chain

CMOS Domino Circuit [Bell lab] (2/3)

- Consist of a pair of a dynamic CMOS and a static inverter
- **ψ=0：Precharge**
- Dynamic gate is charged to V_{DD} , so that the static inverter is discharge to GND
  - ➔ All N-rich nodes driven by domino circuit is low
  - ➔ All NMOS transistor are off
- **ψ=1：Evaluation**
- Domino circuit output makes single transition from Low to High or keeps to Low
  - ➔ No glitches (internal race) and all gates can be switched from precharge to evaluation with same clock edge

CMOS Domino Circuit [Bell lab] (3/3)

Advantages：

- 1. Faster than the corresponding static CMOS circuit
- 2. Low power dissipation
- 3. Full Pull-Down current is available
- 4. Small load cap
- 5. Use of single clock edge to activate the circuit
- 6. No internal race problem(glitch free operation)

Disadvantages：

- 1. Lack of inverted output→ Decrease logic flexibility and therefore requires more transistors for a given logic function

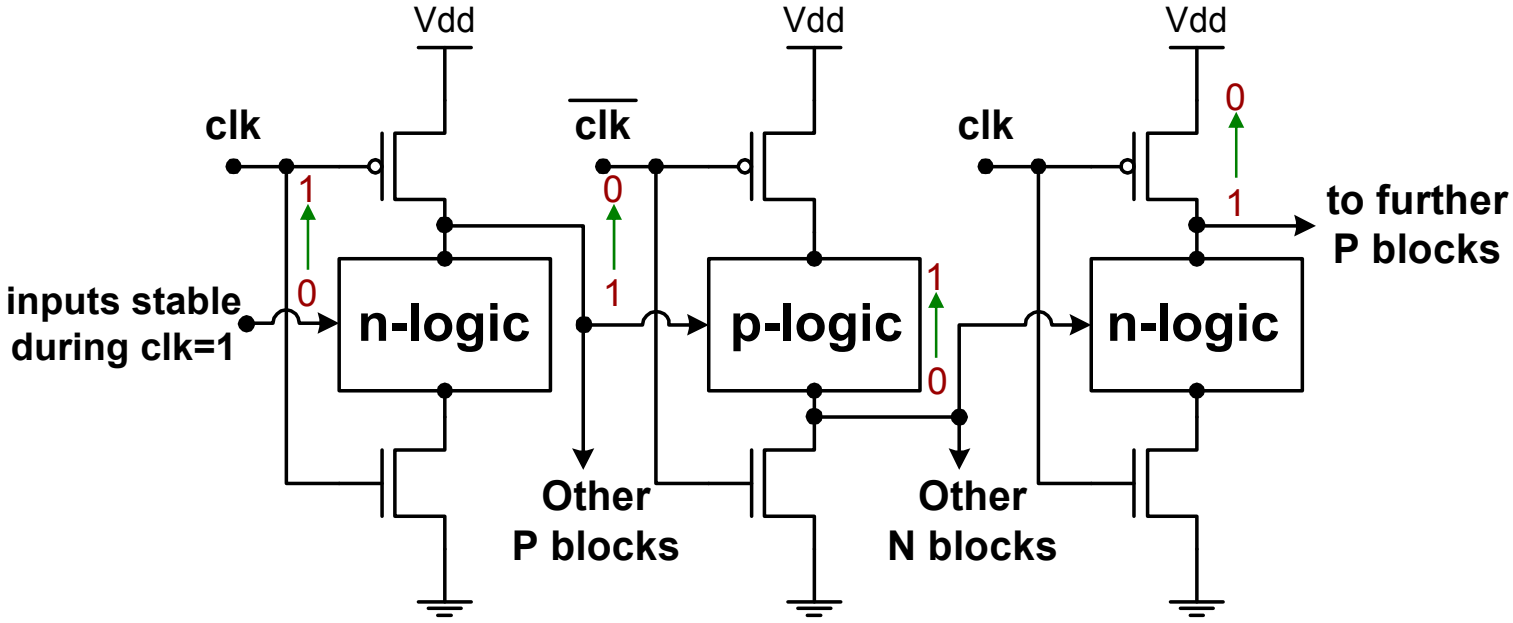
  EX: No way to implement XOR gate
- 2. Charge redistribution problem

5.2.4 Modified Domino Logic -N-P pair Dynamic Circuit (1/3)

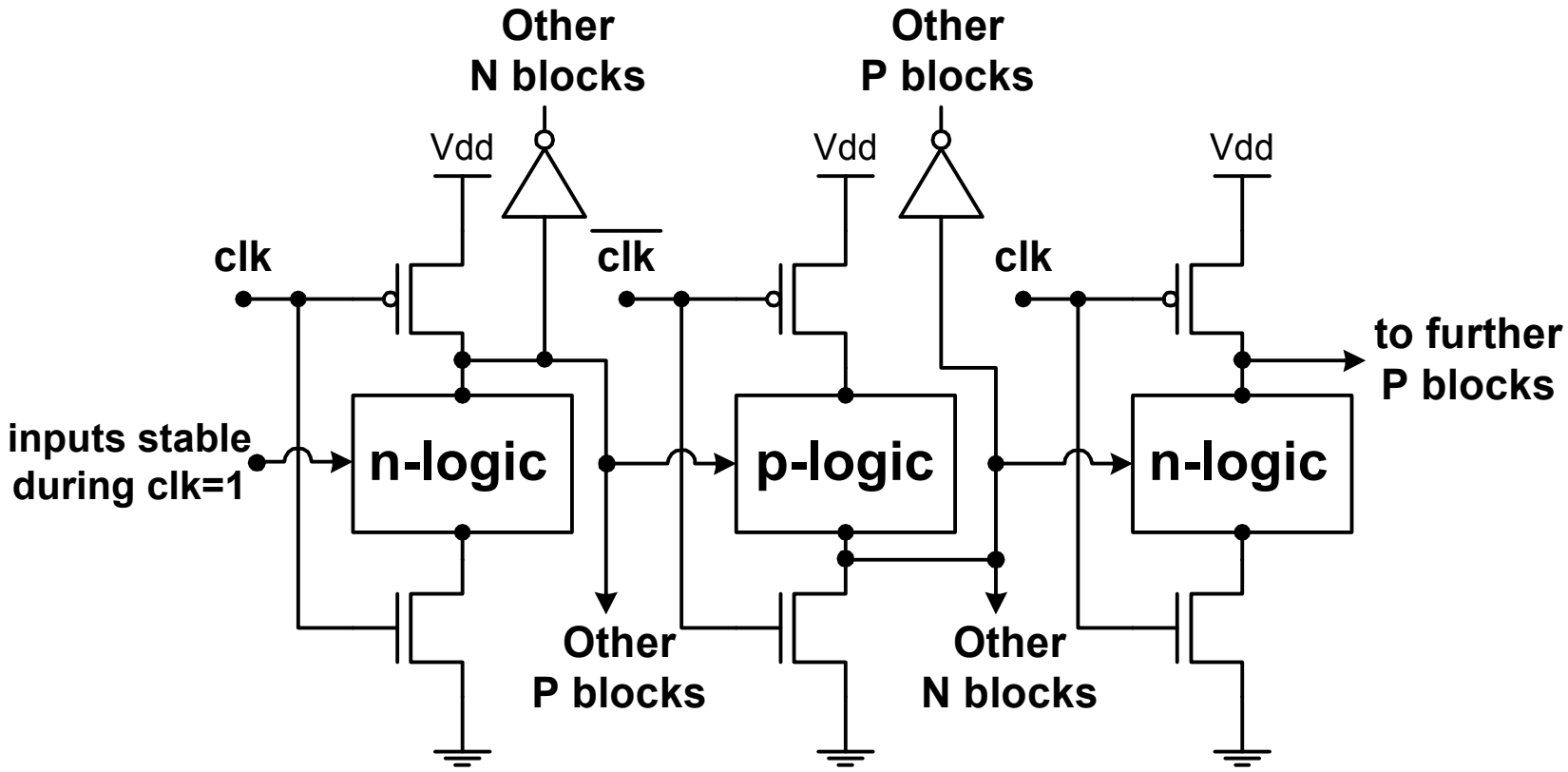
Ref：IEEE JSSC , Vol. sc-19 , P.P. 263 , Apr. 1984

Principles：

1. Remove domino buffer (inverter)
2. P-rich gate can be directly drive an N-rich gate without glitch problem and vice versa ➔ consist of alternating the two types of CMOS dynamic logic gates



Modified Domino Logic -N-P pair Dynamic Circuit (2/3)



Modified Domino Logic -N-P pair Dynamic Circuit (3/3)

- **Advantages：**
  - 1. Smaller area than fully static gates
  - 2. Smaller parasitic cap.
  - 3. Glitch free operation ➔ No internal race problem
  - 4. Inverted output is available
- **Disadvantages：**
  - 1. Still has the charge redistribution problem
  - 2. Low operation speed in PMOS blocks

➢ How to Avoid Charge Redistribution Problem？

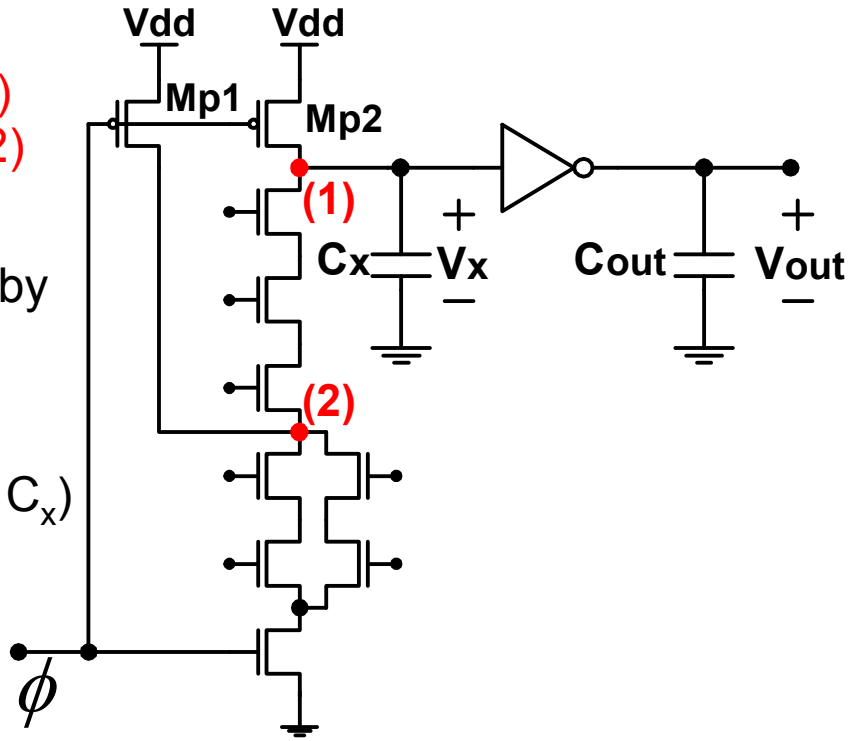
5.2.5 Modified Domino CMOS with Reduced Charge Redistribution

Ref :

- *IEEE JSSC , Vol. sc-21 , P.P. 304 ~ 306 , Apr. 1986*
- *IEEE JSSC , Vol. sc-21 , P.P. 786 ~ 793 , Aug. 1986*
- *IEEE JSSC , Vol. sc-19 , P.P. 263 ~ 266 , Apr. 1984*

(A) Internal Node Precharge

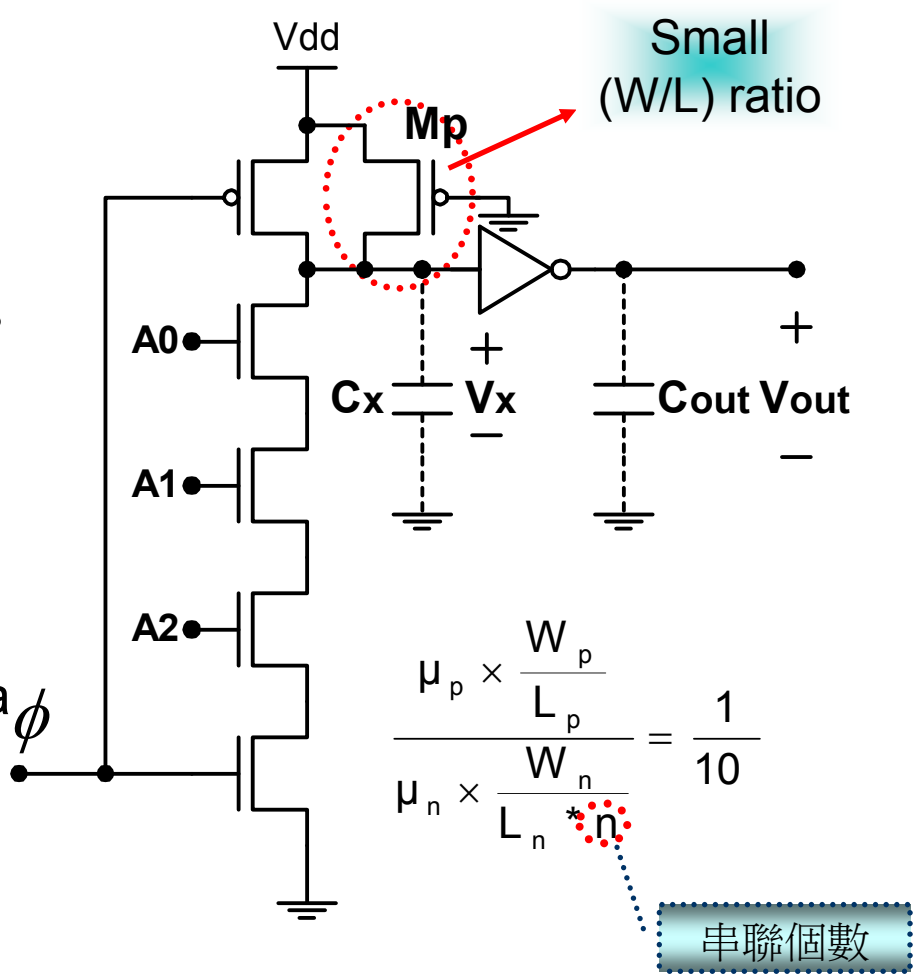
- Mp1 is used to precharge the internal node (2) to avoid charge redistribution between node (2) and (1)
- Charge redistribution can be more eliminated by
  a. Carefully control of layout dependent capacitance
  b. Increase output inverter size (i.e. increase C_x)
- Internal node precharge may slow down the discharge operation in the Evaluation phase



(B) Static Domino gate

- Domino gate may be made static by adding a weak PMOS (small W/L ratio)
- Static capability to allow lower frequency operation or to avoid the risk of storing data on floating nodes
- Pull-up time could be slower than pull-down speed
- Can moderate the charge redistribution problem and improve noise margin
- W/L of the Mp must be low to keep a low DC power dissipation (EX : L=10um , W=2um)
- Clocked PMOS can be omitted
- Not Valid For High Speed Circuit

Charge Compensation Circuit

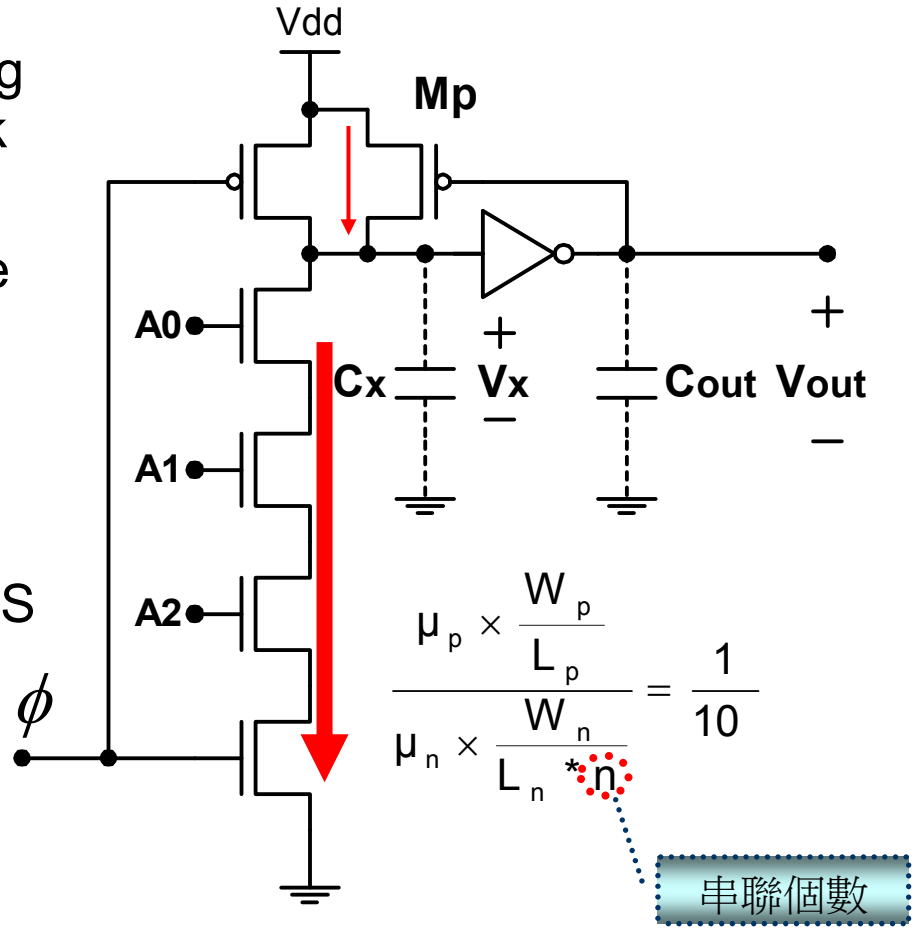


(C) Feedback PMOS Charge Compensation Circuit

- Domino gate may be made latching by placing a weak PMOS feedback transistor
- The charge compensation scheme can effective solve the charge redistribution problem , but the operating speed is decreased as compared to standard domino
- Standard domino with series NMOS higher than 5 is unable . Due to charge redistribution .

Solution：Zipper CMOS

Alternate Charge Compensation Scheme

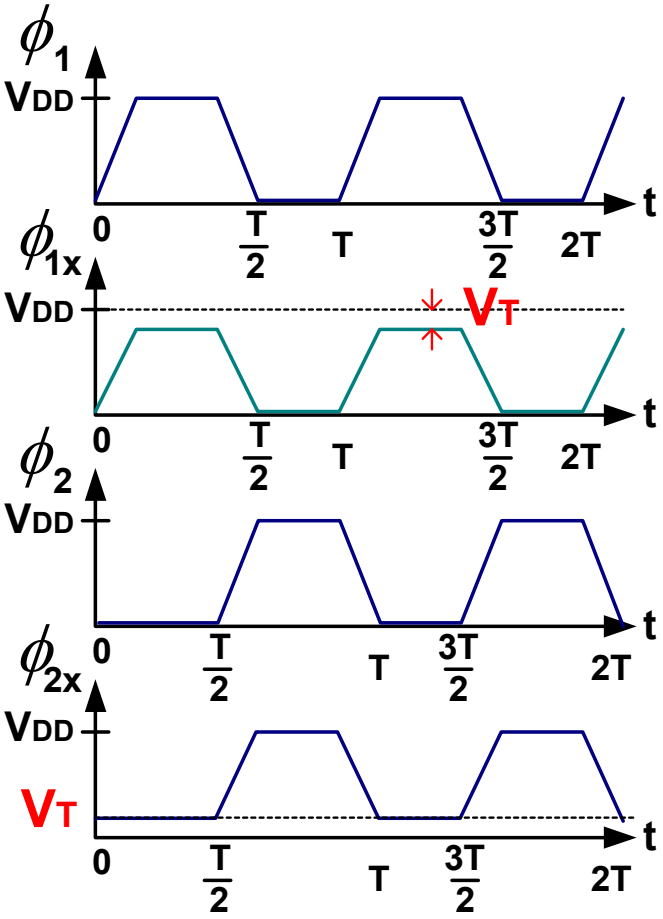


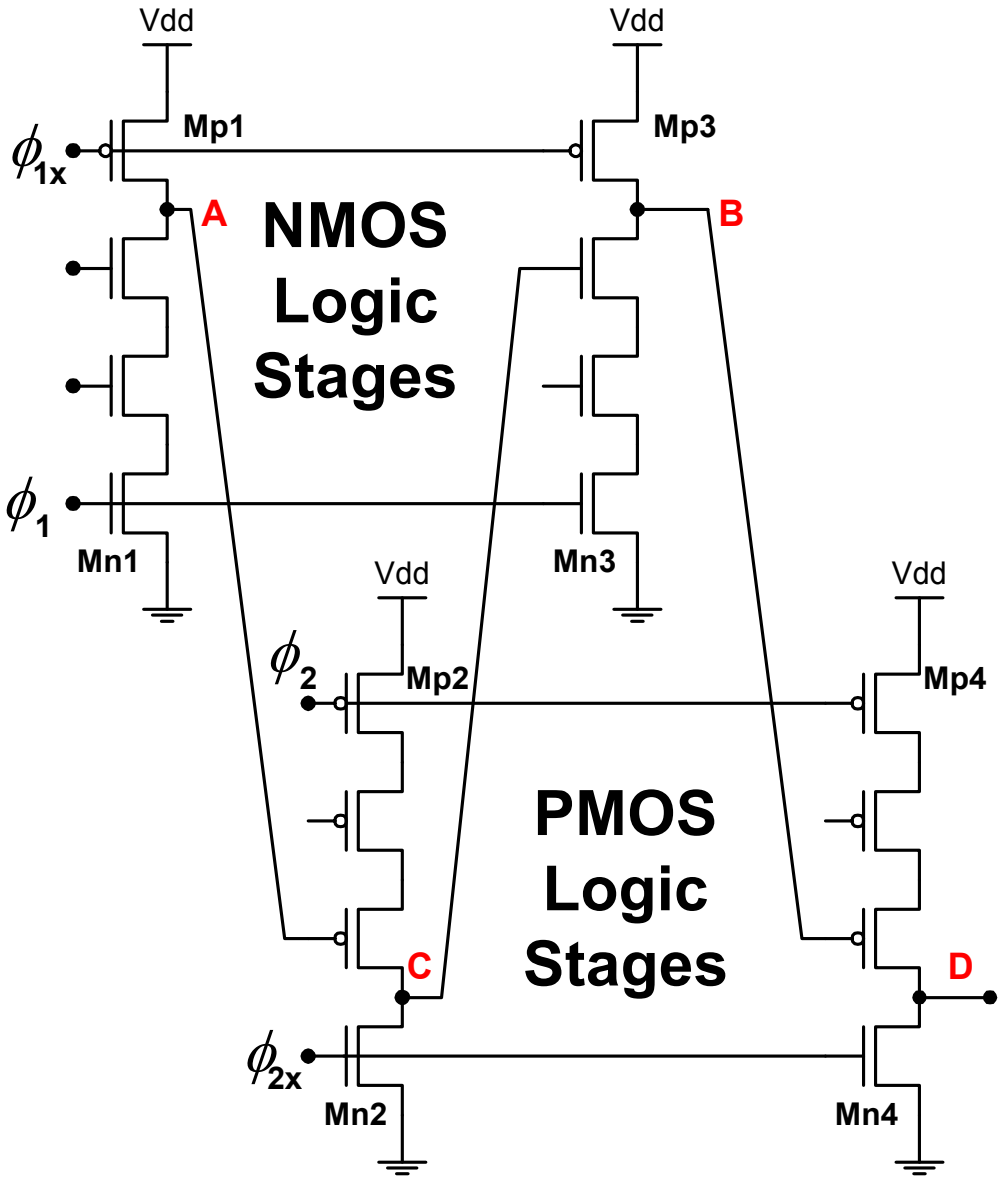
5.2.6 Zipper CMOS (1/3)

- The gate connection is similar as the N-P pair domino structure
- The precharge PMOS in the N-block and the predischarge NMOS in the P-block is controlled by a special Zipper clock ψ_{1X} , ψ_{2X} where the voltage swing of ψ_{1X} and ψ_{2X} are

  $\psi_{1X} = 0\ V \sim V_{DD} - |V_{TP}|$; $\psi_{2X} = |V_{TN}| \sim V_{DD}$
- In the Evaluation phase$\psi_1 = V_{DD}$; $\psi_{1X} = V_{DD} - |V_{TP}|$; $\psi_2 = 0\ V$; $\psi_{2X} = |V_{TN}|$. The PMOS Mp1 controlled byψ_{1X} is nearly turn-on while the Zipper clock$\psi_{1X} = V_{DD} - |V_{TP}|$, If the charge redistribution is occurred at node A , V_A is pulled-down , Mp1 is turned-on by$\psi_1 = (V_{DD} - |V_{TP}|)$ to precharge node A to V_{DD} ➔No charge redistribution

Zipper CMOS (2/3)





Zipper CMOS (3/3)

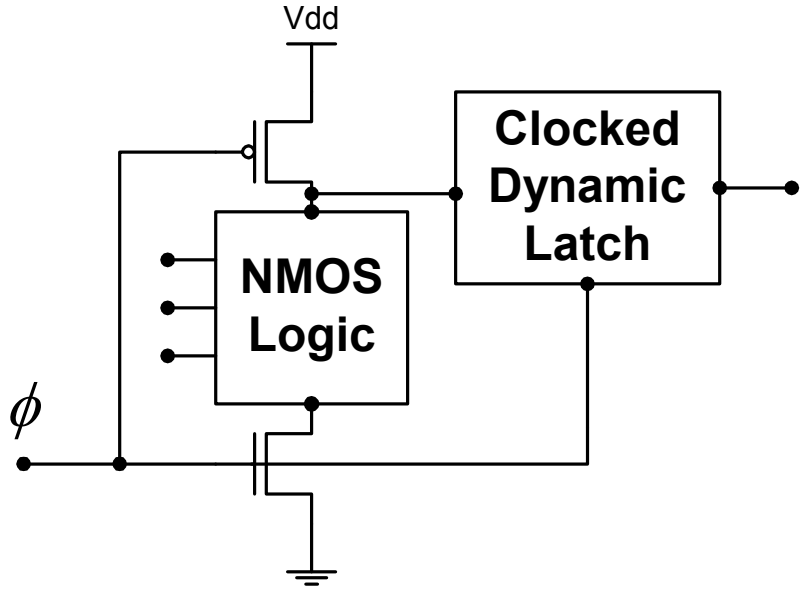
- **Advantages：**
  - 1. No charge redistribution problem
  - 2. No internal race problem if carefully design
  - 3. No DC power dissipation
  - 4. The operating speed is compare to the standard domino
- **Disadvantages：**
  - Requires special clock generator for Zipper CMOS logic , and more clock line

5.2.7 NORA – CMOS pipelined circuits

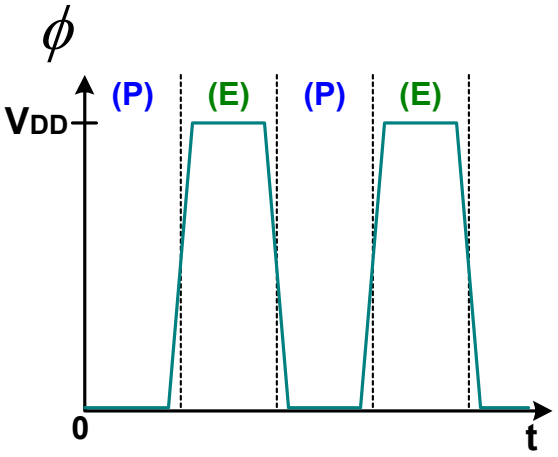
Ref : IEEE JSSC , Vol. sc-18 , P.P. 261 ~ 266 , June . 1983

- NO RAce circuit

(A) Basic Concept of Dynamic Latch



(a) General Structure



(b) Clocking

- Each section contains two units：logic unit and latch unit
- Latch unit as the pipeline – section control unit

(B) NORA Circuit (1/3)

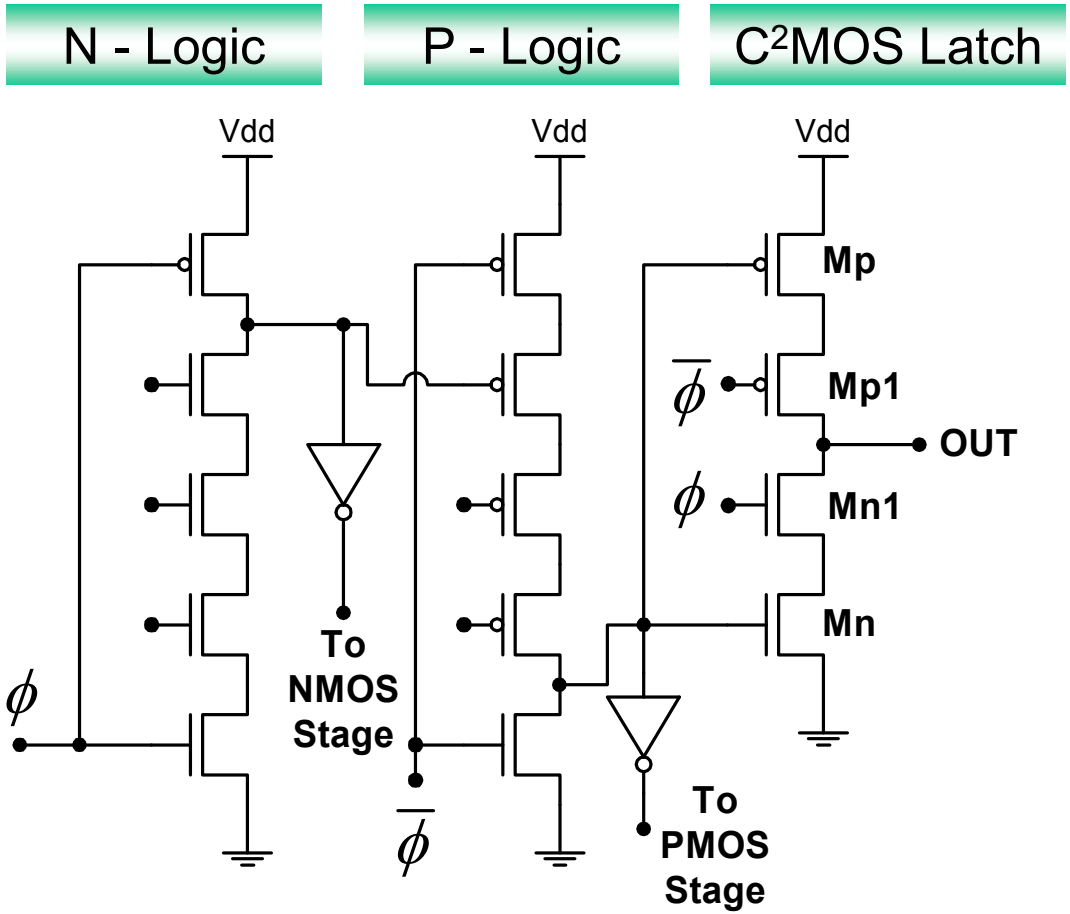
ψ- section :

ψ=0 Precharge ; Data out is hold

ψ=1 Evaluation ; Data out Valid

Configuration :

1. Direct couple of N-P dynamic logic blocks
2. Indirect couple of same type logic
   ➔ Domino logic
3. Data out must stored in C²MOS latch



(a) NORA ψ- section

\	Logic	Latch
$\phi = 0$	Precharge	OFF
$\phi = 1$	Evaluation	ON

(B) NORA Circuit (2/3)

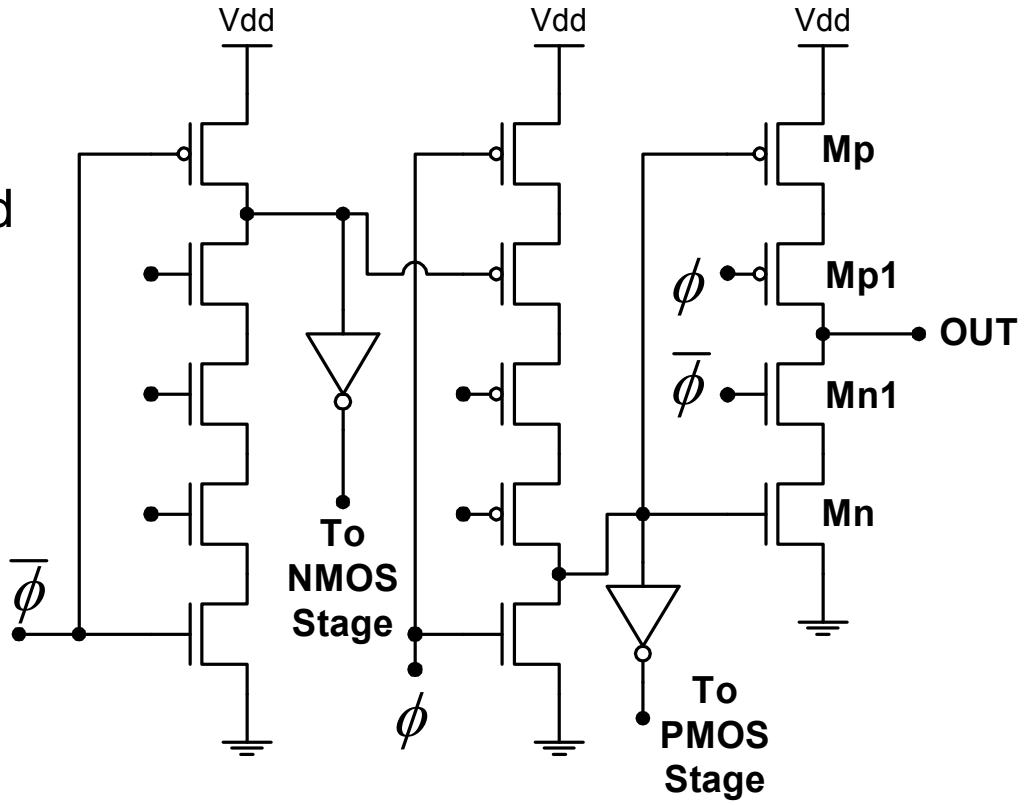
N - Logic | P - Logic | C²MOS Latch

$\overline{\psi}$- section :

ψ=1 Precharge ; Data out is hold

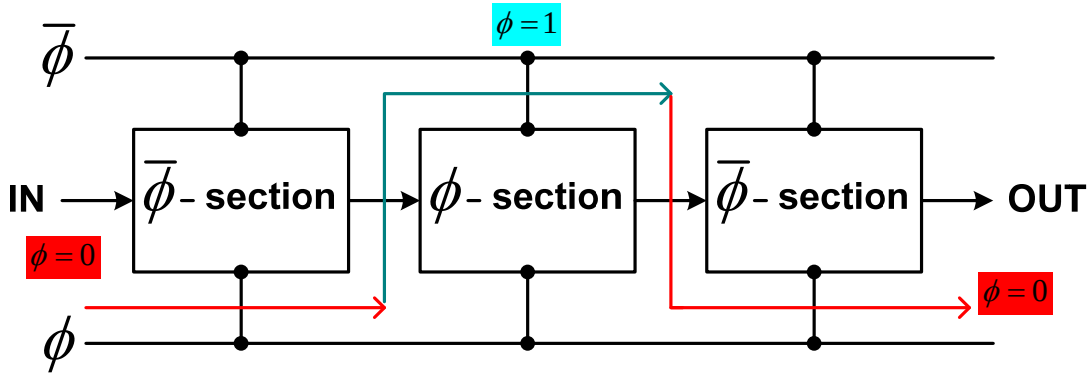
ψ=0 Evaluation ; Data out Valid

\	Logic	Latch
$\phi = 1$	Precharge	OFF
$\phi = 0$	Evaluation	ON



(b) NORA $\overline{\psi}$- section

(B) NORA Circuit (3/3)



$\phi = 0$	Evaluation	Precharge	Evaluation
$\phi = 1$	Precharge	Evaluation	Precharge

NORA Cascades

Characteristic：

- Inverted and noninverted signals are provided
- N-P (P-N) sequence can be of arbitrary depth
- Data flows and latched in this way from **$\overline{\psi}$→ψ→$\overline{\psi}$ as a pipeline structure**

(C) NORA race free properties and logic composition rules

(一) Internal Delay Race free Property (1/2)

- **Definition：**
  - The capability of the dynamic block to keep its precharge signal during the delay time of the previous blocks to set up the internal inputs
- **Properties：**
  - a. During the precharge： ” Precharge to cut-off the corresponding input MOS ”
  - b. During the Evaluation： ” Internal inputs are Glitch-Free ” (can make only are transition)

(一) *Internal Delay Race free Property* (2/2)

- ***Composition Rules*** ：

  - a. The number of static inversion gates between two N-P (P-N) dynamic block is EVEN or 0
  - b. The number of static inversion gates between two N-N (P-P) dynamic block is ODD
  - c. For mixing dynamic blocks with static CMOS , the static must be put between dynamic blocks and C^2MOS latch . Static functions can be used after the C^2MOS stage .

(二) *Clock Race Free* (1/11)

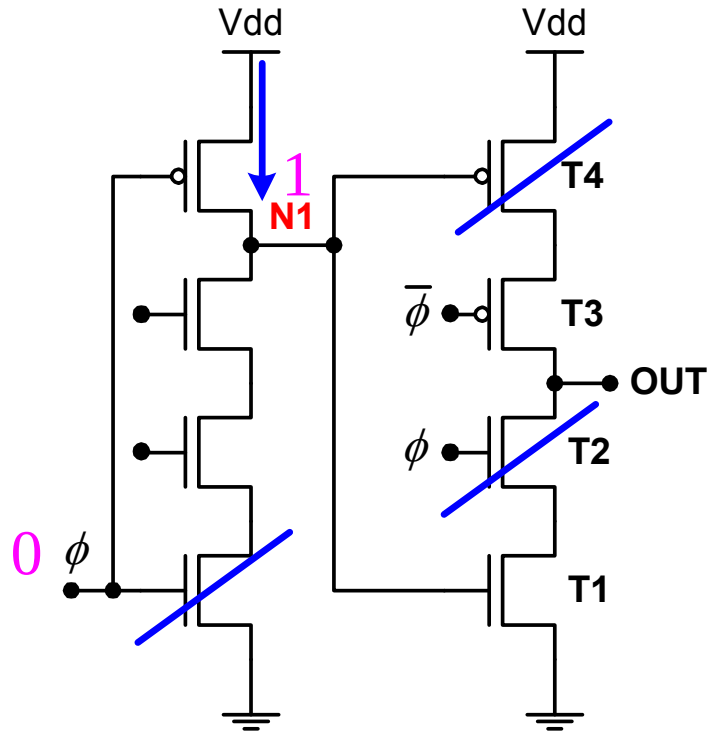
- ***Definition*** ：

- NORA pipelined section keeps its output results in spite of clock skew (i.e. high – high or low – low overlaps) . The latch signal should not be altered by precharge signal or by input variations

(二) *Clock Race Free* (2/11)

Case1：Precharge Racefree (1/3)

(The case that the precharge signal is altered by the input)



ψ=0 , N1=1 ➔ T2 OFF , T4 OFF

OUT can keep its value even if

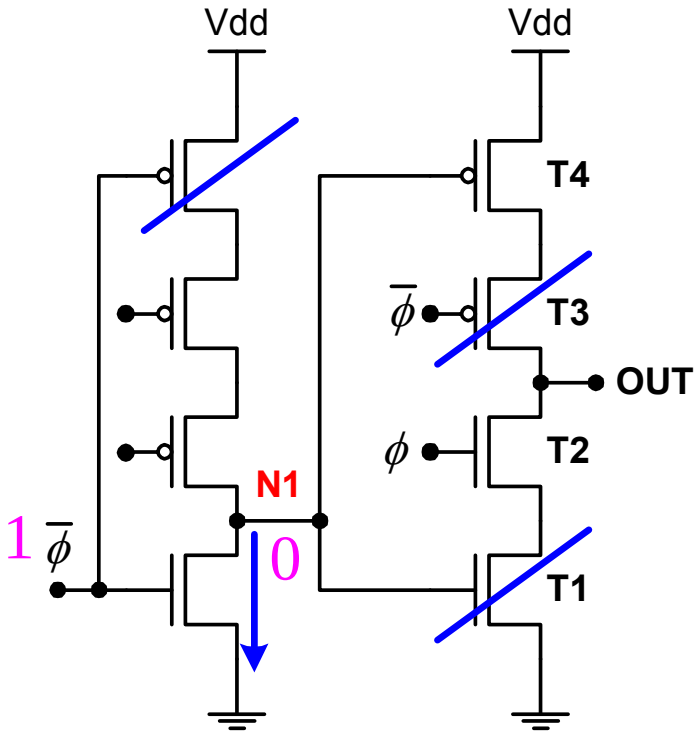
$\overline{\psi}$=0 (T3 ON)

Even Inversion

(二) *Clock Race Free* (3/11)

Case1：Precharge Racefree (2/3)

(The case that the precharge signal is altered by the input)



$\overline{\psi}$=1 , N1=0 ➔ T3 OFF , T1 OFF

OUT can keep its value even if

ψ=1 (T2 ON)

Even Inversion

(二) *Clock Race Free* (4/11)

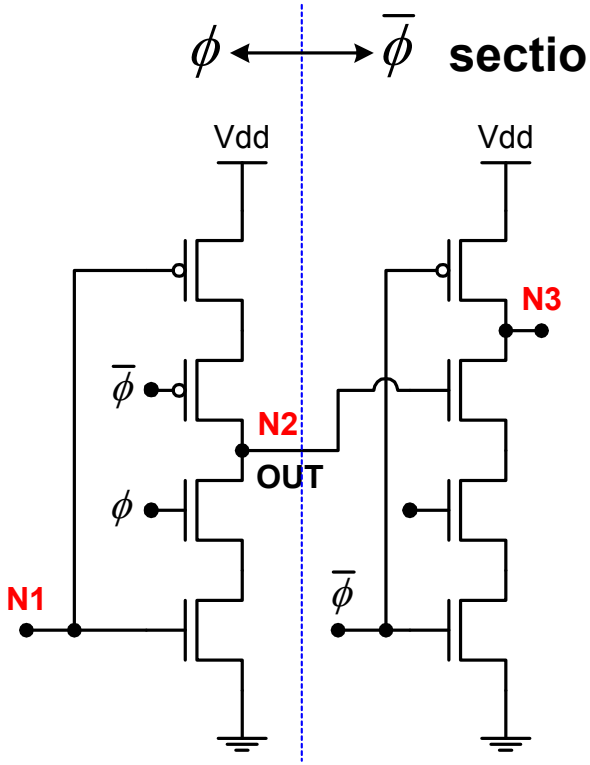
Case1：Precharge Racefree (3/3)

Composition Rules：

- There is an even number of inversions between the last dynamic block and the C^2MOS output latch
- There could be signal races for clock signals with very slow rise and fall time (10 to 20 times the gate delay) !

(二) Clock Race Free (5/11)

Case2：Input Variation Racefree (1/3)



Even Inversion

- **The case that the precharge signal is kept by inputs (ψ—section)**

  OUT=0 , N1=1

  OUT could be changed only when N1=0 and $\overline{\psi}$=0

- Precharge signal (ψ=1 and $\overline{\psi}$=0) has no effect on the OUT signal (in the ψ– section)
- Consider the input ($\overline{\psi}$– section) :

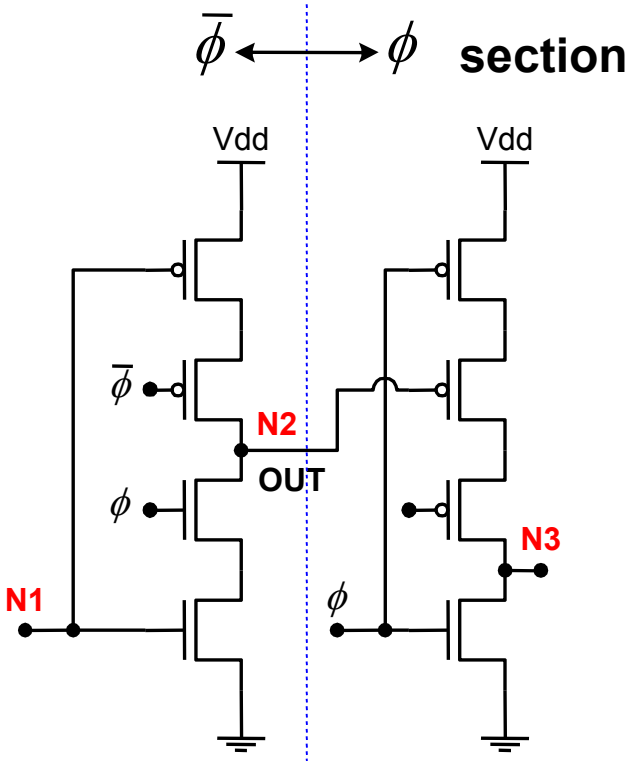
  N3 could be changed only when N2 : 0 → 1 and $\overline{\psi}$=1

As $\overline{\psi}$=1 ; N2 : 0 → 1 ➔ Impossible !

➔ Input variation Race free

(二) Clock Race Free (6/11)

Case2：Input Variation Racefree (2/3)



Even Inversion

- **The case that the precharge signal is kept by inputs (ψ—section)**

  OUT=1 , N1=0

  OUT could be changed only when N1=1 and ψ=1

- Precharge signal (ψ=1 and $\bar{\psi}$=0) has no effect on the OUT signal (in the ψ– section)
- Consider the input ($\bar{\psi}$– section) :

  N3 could be changed only when N2 : 1 → 0 and ψ=0

As ψ=0 ; N2 : 1 → 0 ➔ Impossible !

➔ Input variation Race free

(二) *Clock Race Free* (7/11)

Case2：Input Variation Racefree (3/3)

Composition Rules：

- a. There exits at least a dynamic block in every chain of the pipelined section that there is an even number of inversions between the C^2MOS latch and this dynamic block
- b. The total number of inversions between the two CMOS latch stage is even

(二) *Clock Race Free* (8/11)

Example (1/3) :

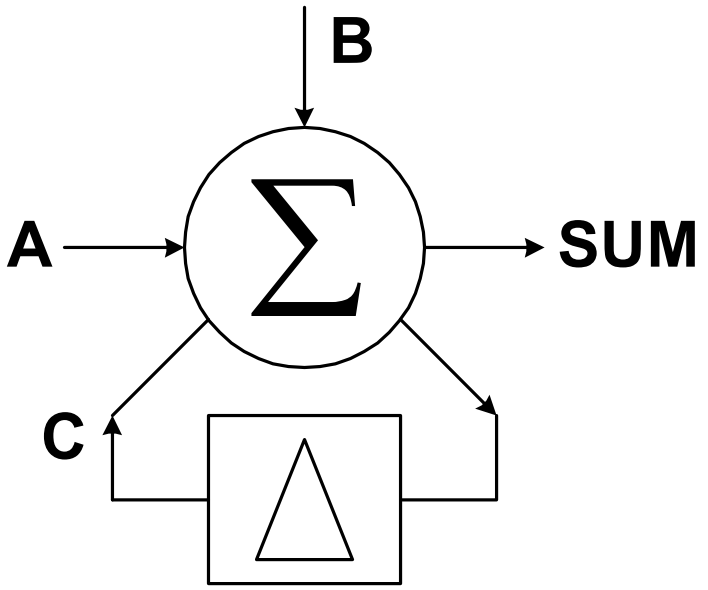
NORA Serial – Parallel Multiplier

$A=(a_3a_2a_1a_0)$

$B=(b_3b_2b_1b_0)$

The binary product is given by

$A \times B=(p_6p_5p_4p_3p_2p_1p_0)$



(a) Serial Adder

(二) Clock Race Free (9/11)

Example (2/3) :

The product terms are of the from

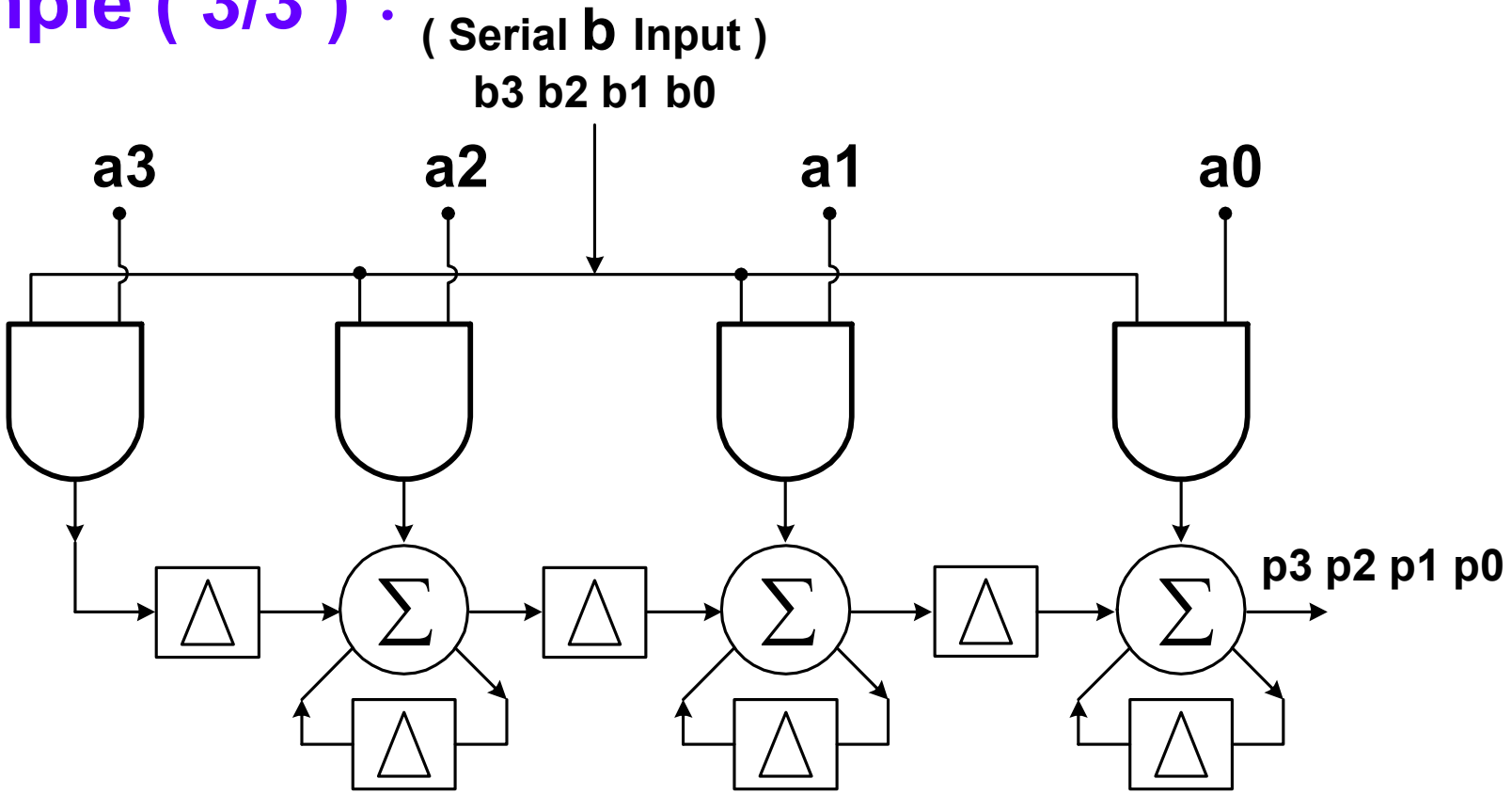
$$P_i = \sum_{\alpha+\beta=i} \left(a_\alpha b_\beta \right)$$

The 4 – bit multiplication yields

$$\begin{aligned}
P_0 &= a_0b_0 \\
P_1 &= a_0b_1 + a_1b_0 \\
P_2 &= a_0b_2 + a_1b_1 + a_2b_0 \\
P_3 &= a_0b_3 + a_1b_2 + a_2b_1 + a_3b_0 \\
P_4 &= a_1b_3 + a_2b_2 + a_3b_1 \\
P_5 &= a_2b_3 + a_3b_2 \\
P_6 &= a_3b_3
\end{aligned}$$

(二) *Clock Race Free* (10/11)

Example (3/3) :

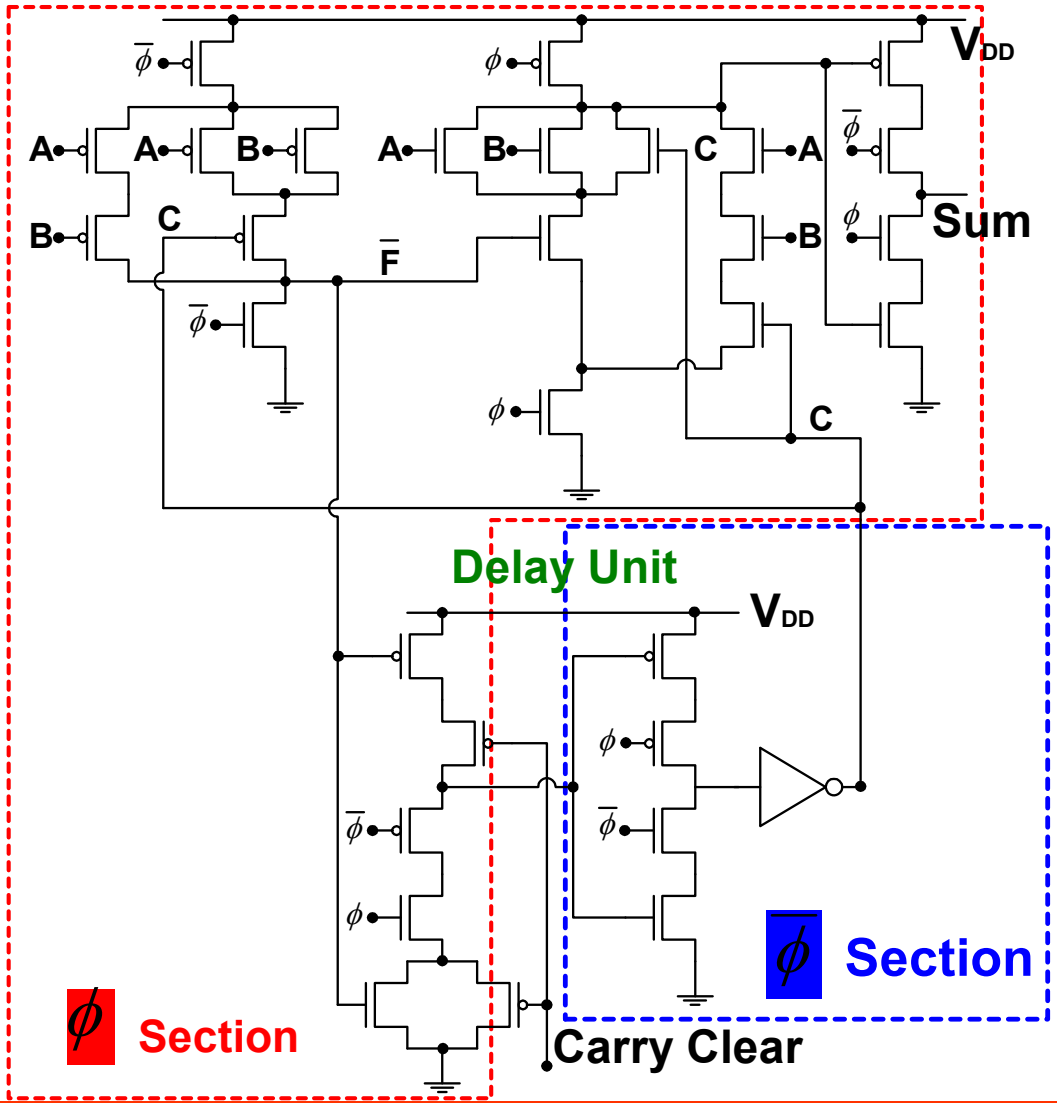


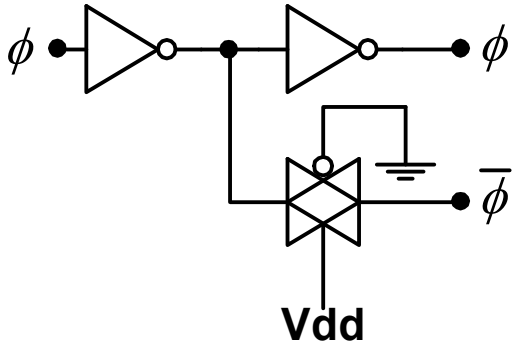
(b) Function Diagram
4-bit Serial-Parallel Multiplier

(二) *Clock Race Free* (11/11)

- **Advantages：**
  - 1. Simple , small hardware
  - 2. Regular
  - 3. Extensive
- **Disadvantages：**
  - Requires long time for one multiplication

- NORA serial Adder Circuit





Can reduce overlapping between ψ and $\bar{\psi}$

5.2.11 True – Signal - Phase – Clock (TSPC) CMOS

Ref : [1] IEEE JSSC vol. SC-22 PP.899-901, Oct.1987

[2] IEEE JSSC vol. SC-24 PP.62-70, Feb 1989

[3] IEEE JSSC vol. 25 PP.225-233, Feb 1990

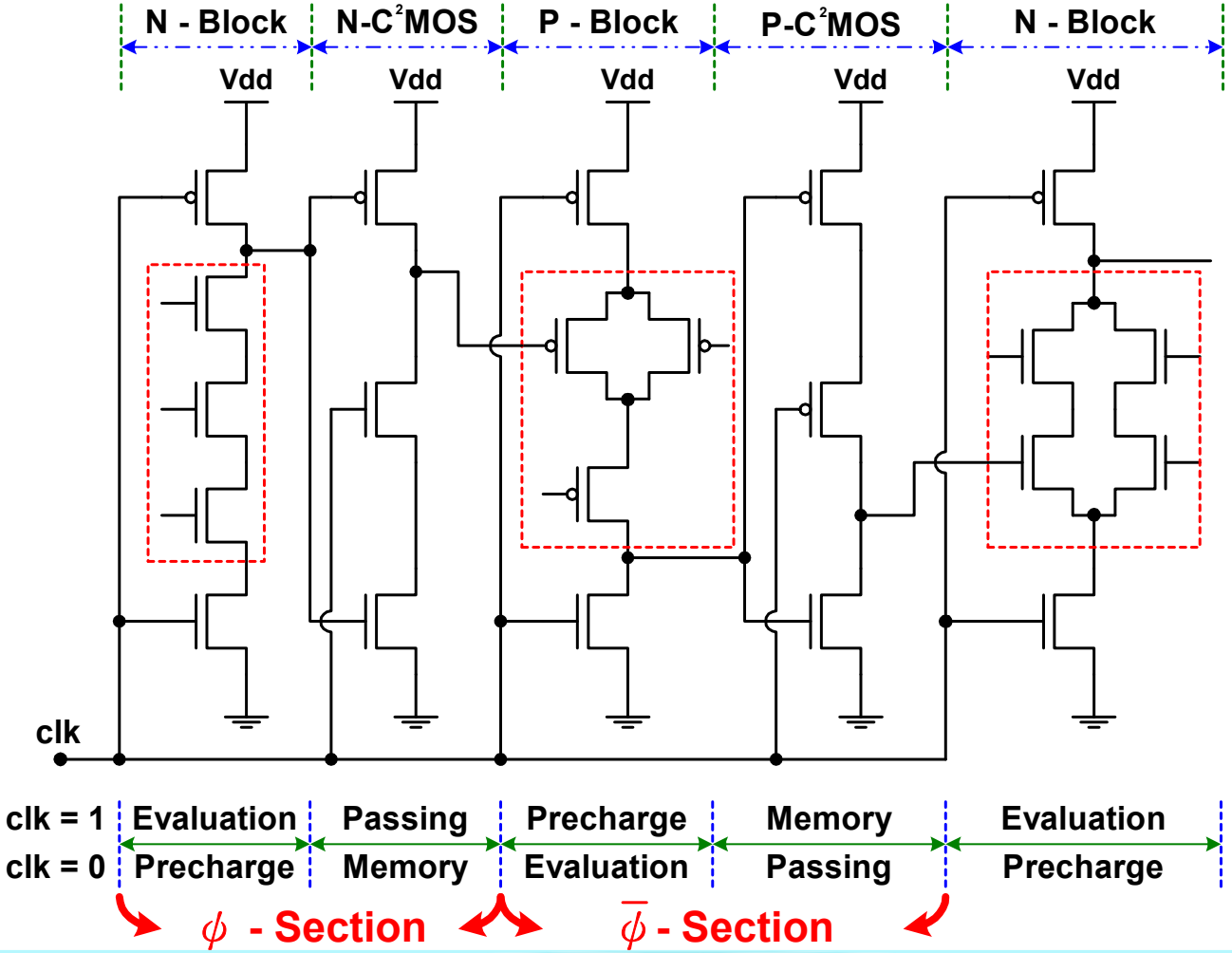
True-Single-Phase-Clock Login

The simplest clocking strategy

(一)、Precharge (Dynamic) Logic (1/2)

- **N - Block + N - C^2MOS = (ψ- section in the NORA circuit)**
- **P - Block + P - C^2MOS = ($\bar{\psi}$- section in the NORA circuit)**

(一)、Precharge (Dynamic) Logic (2/2)

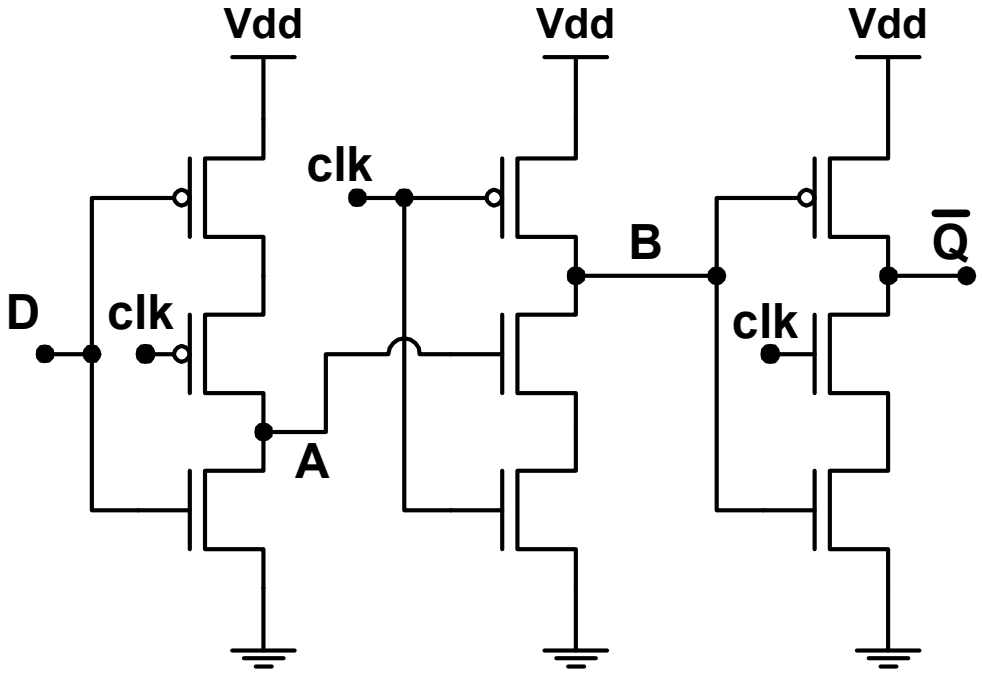


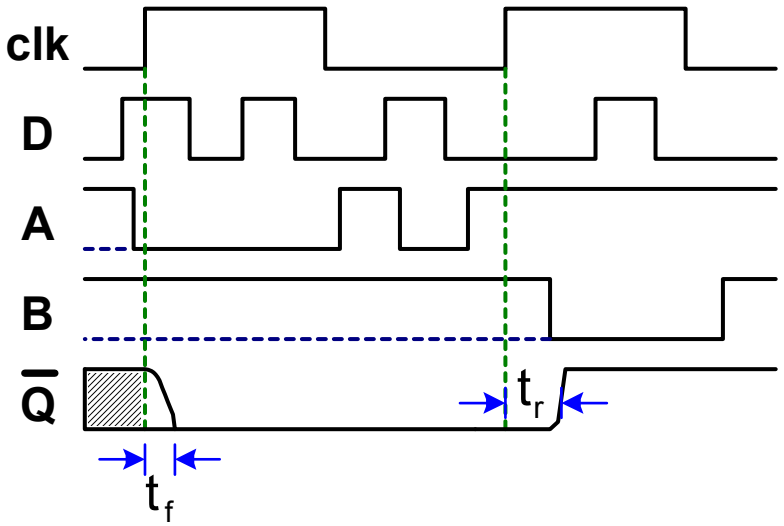
A pipeline of precharged logic circuits using the single-phase clock

(二)、Storage Elements

- **D-type FF 'S (1/3)**

(1) Positive edge triggered FF (ETDFF)

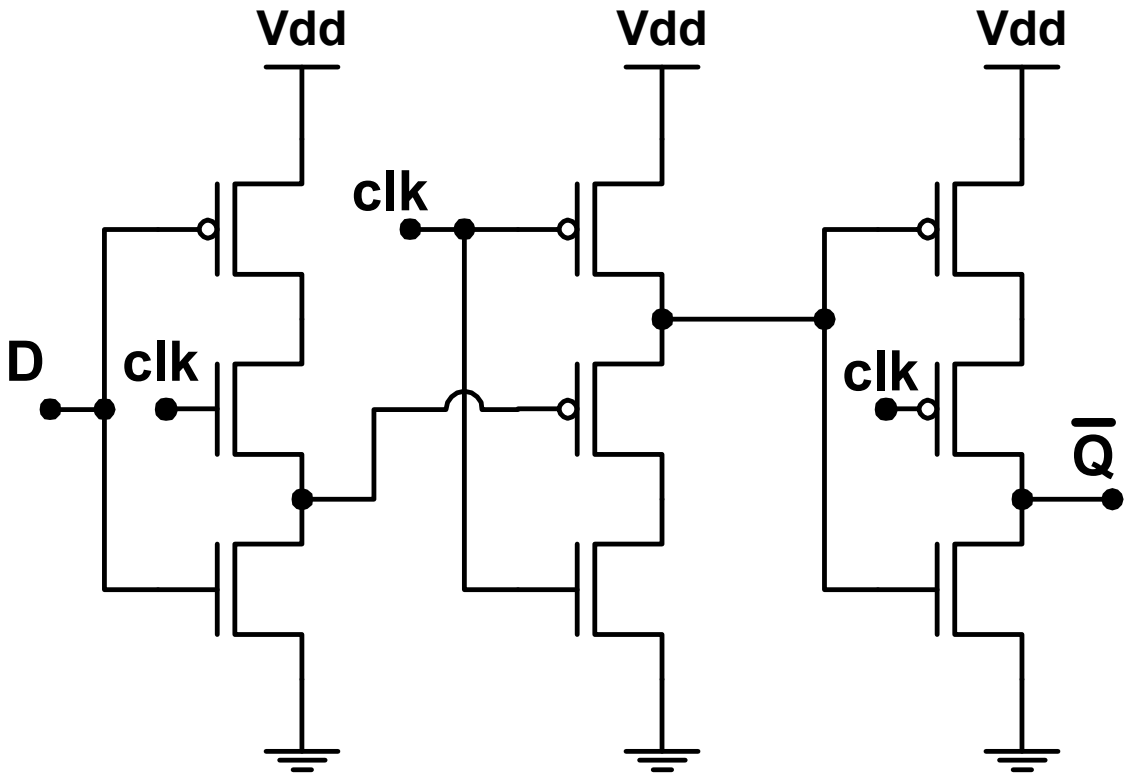




(二)、Storage Elements

- D-type FF 'S (2/3)

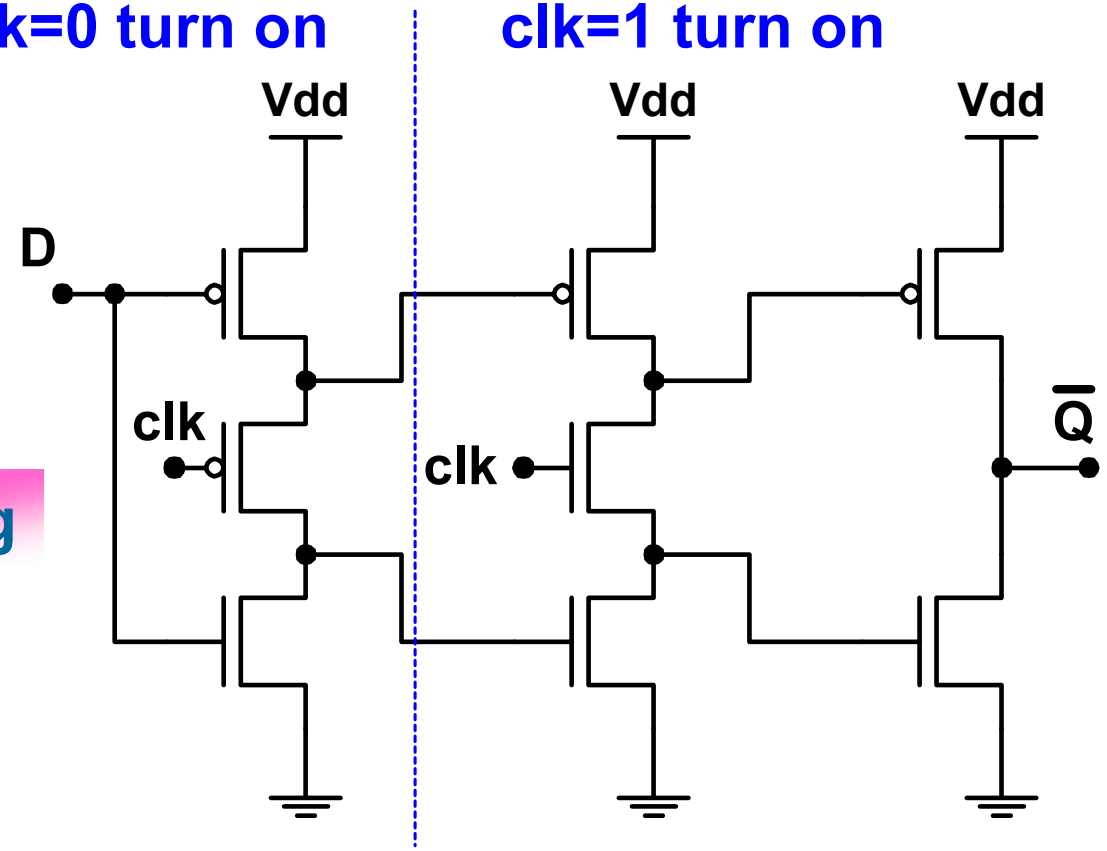
(2) Negative ETDFF



(二)、Storage Elements

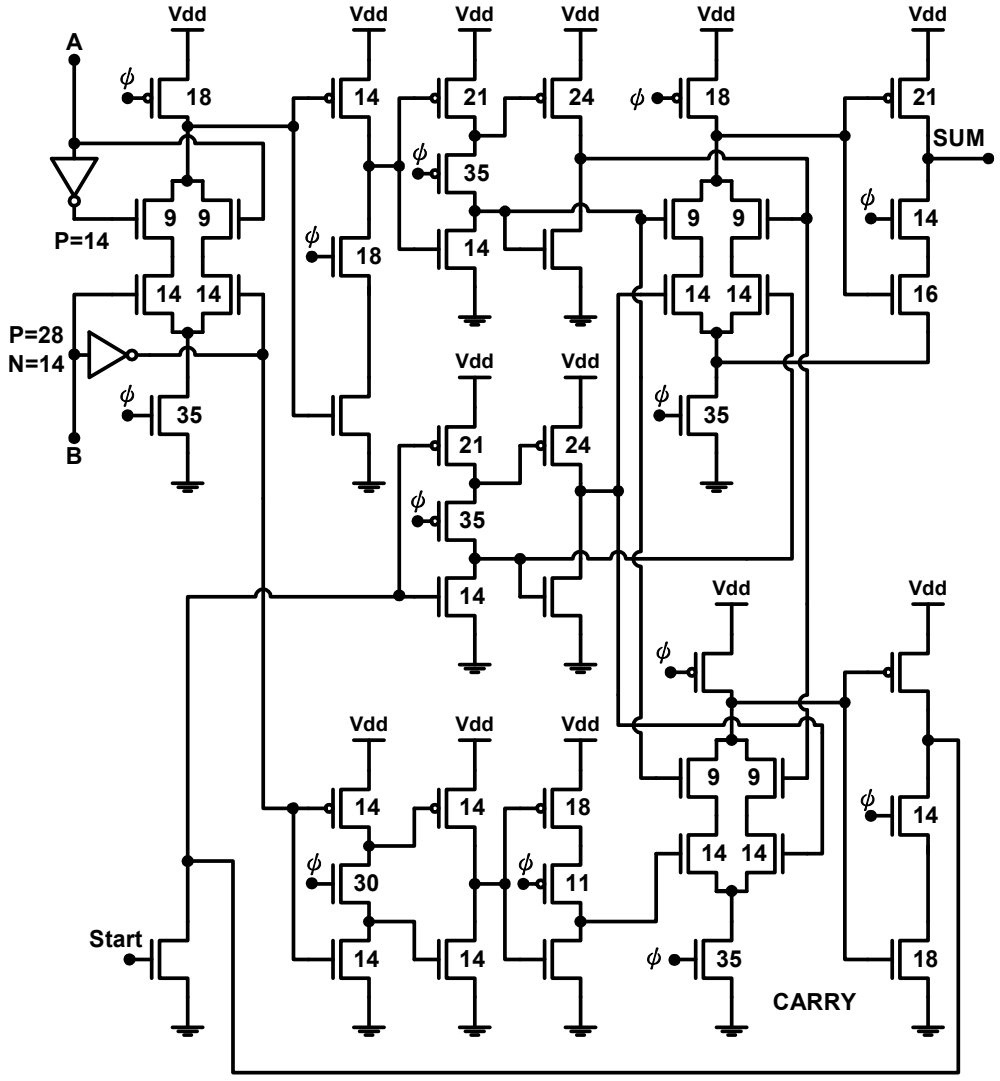
- D-type FF 'S (3/3)

(3) Positive ETDFF using split – output latch stages



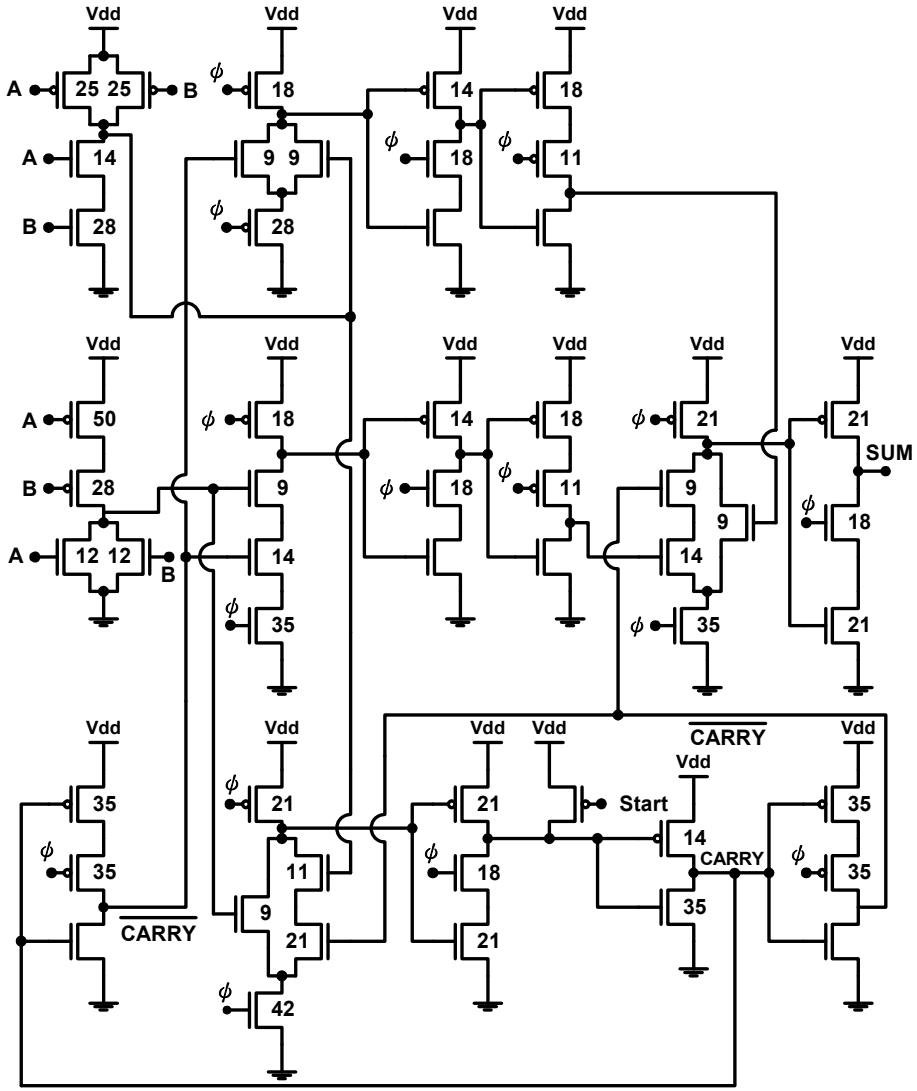
Non full swing

(三)、Circuit Examples：TSPC Serial Full Adder (1/2)：



Serial full adder with all logic in N-blocks (TSPC adder 2).

(三)、Circuit Examples：TSPC Serial Full Adder (2/2)：



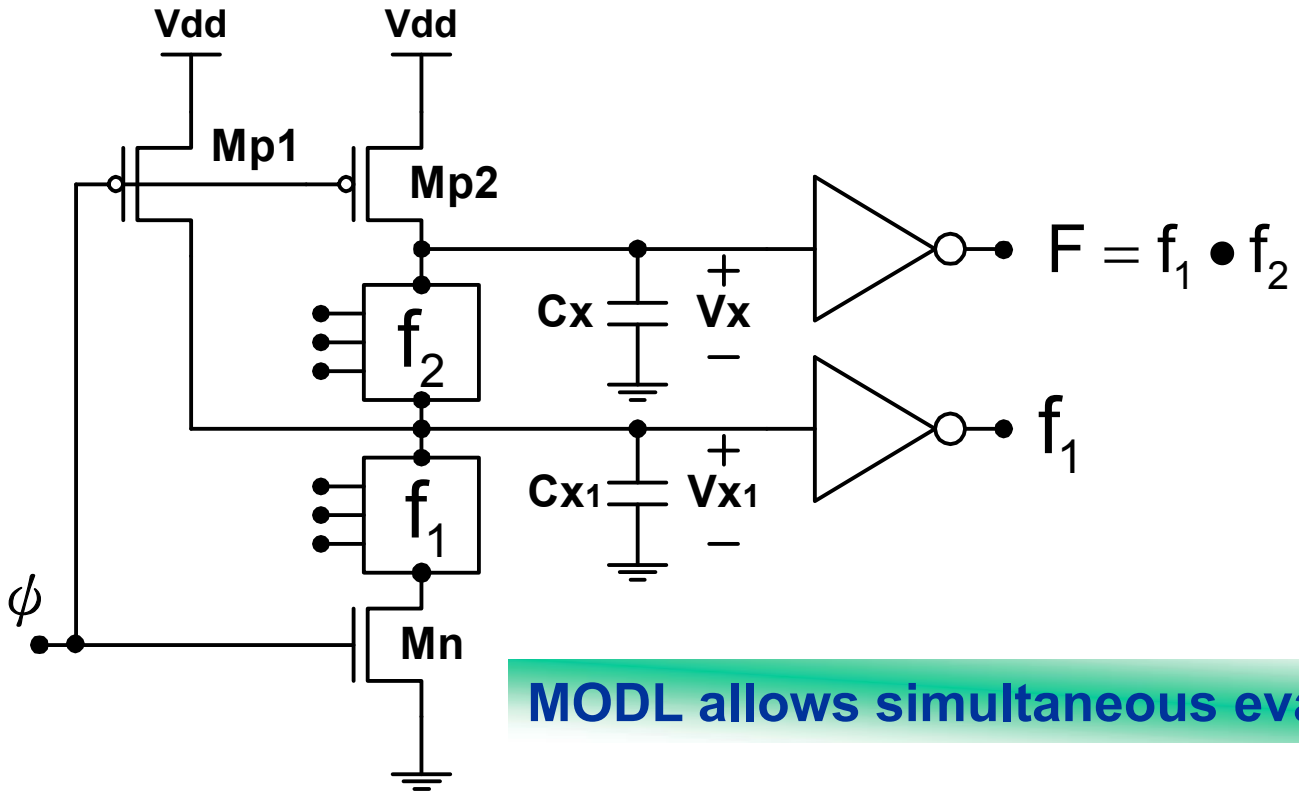
Fast serial full adder (TSPC adder 3).

5.2.12 Multiple – Output Domino Logic (MODL)

Ref: [1] IEEE JSSC vol. 24, PP.358-369, April, 1989

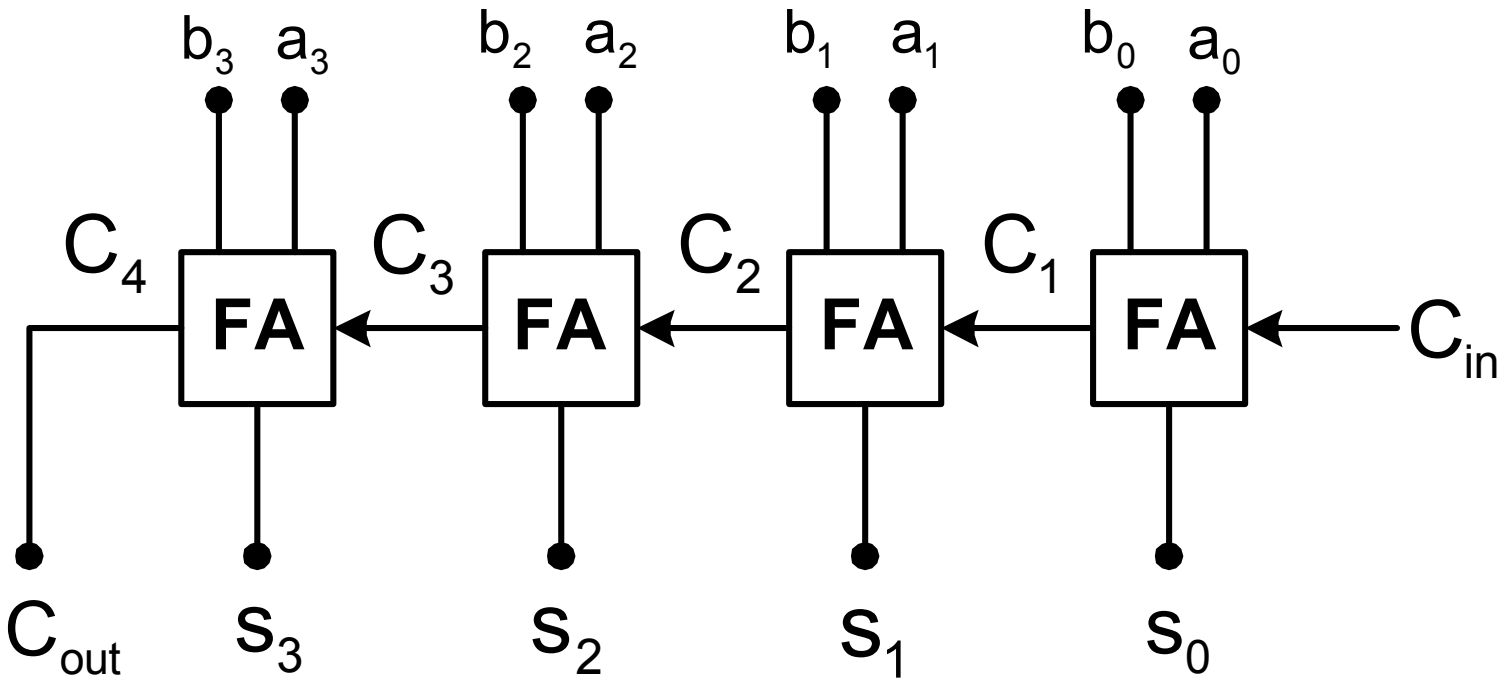
[2] IEEE JSSC vol. 25, PP.800-805, June, 1990

A. Concept :



MODL allows simultaneous evaluation of F and f_1

B. Design Example: MODL Carry-Look-Ahead Adder (1/2)



Basic 4-bit Parallel Adder

B. Design Example: MODL Carry-Look-Ahead Adder (2/2)

- **Basic addition algorithms**

$S_i = a_i \oplus b_i \oplus C_i$

$C_{i+1} = a_i b_i + (a_i \oplus b_i) C_i$

- **Carry-Look-Ahead Addition algorithms**

generate function $g_i = a_i\, b_i$

Propagate function $P_i = (a_i \oplus b_i)$

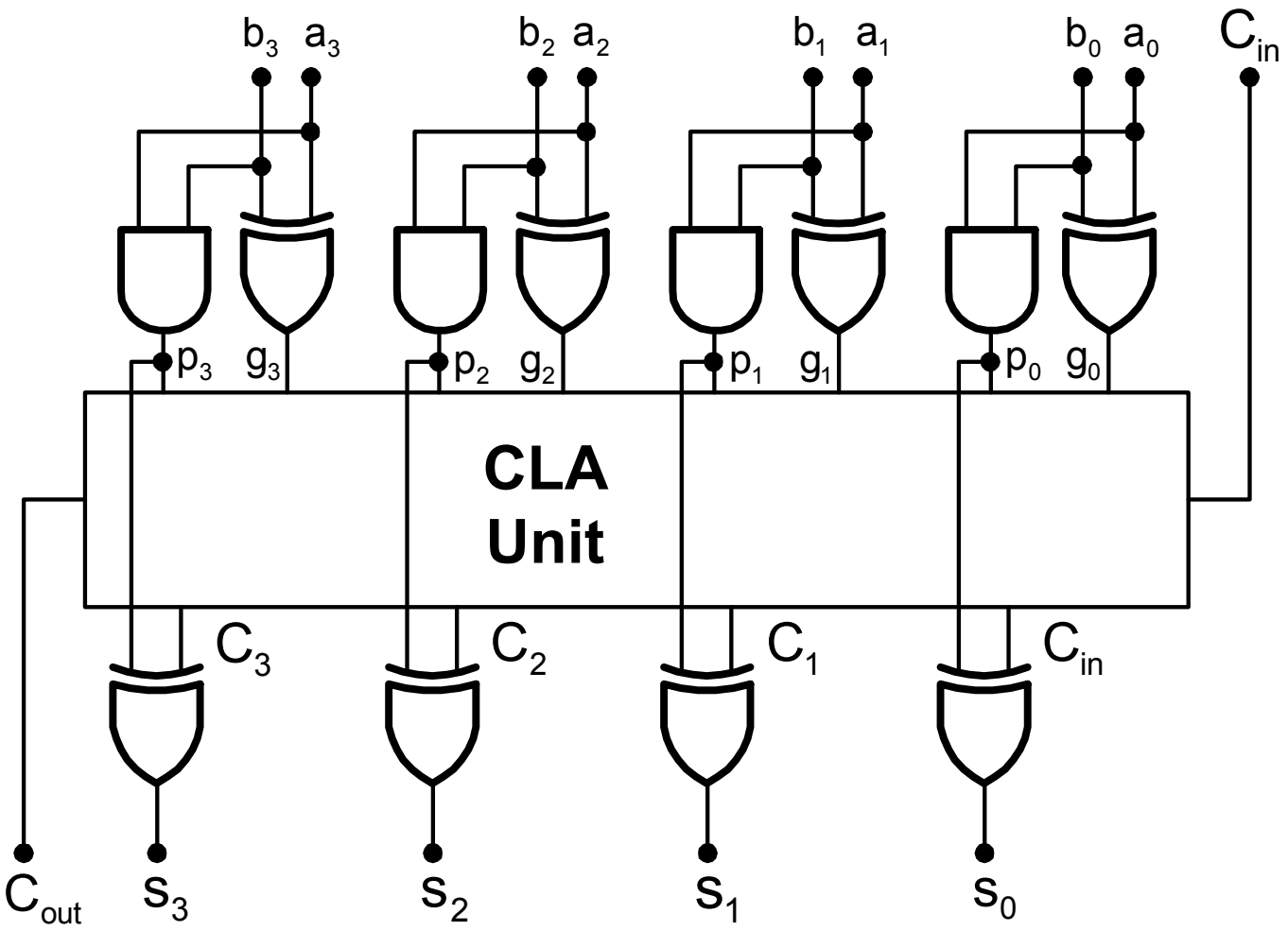
$C_0 = C_{in}$

$C_1 = g_0 + P_0\, C_{in}$

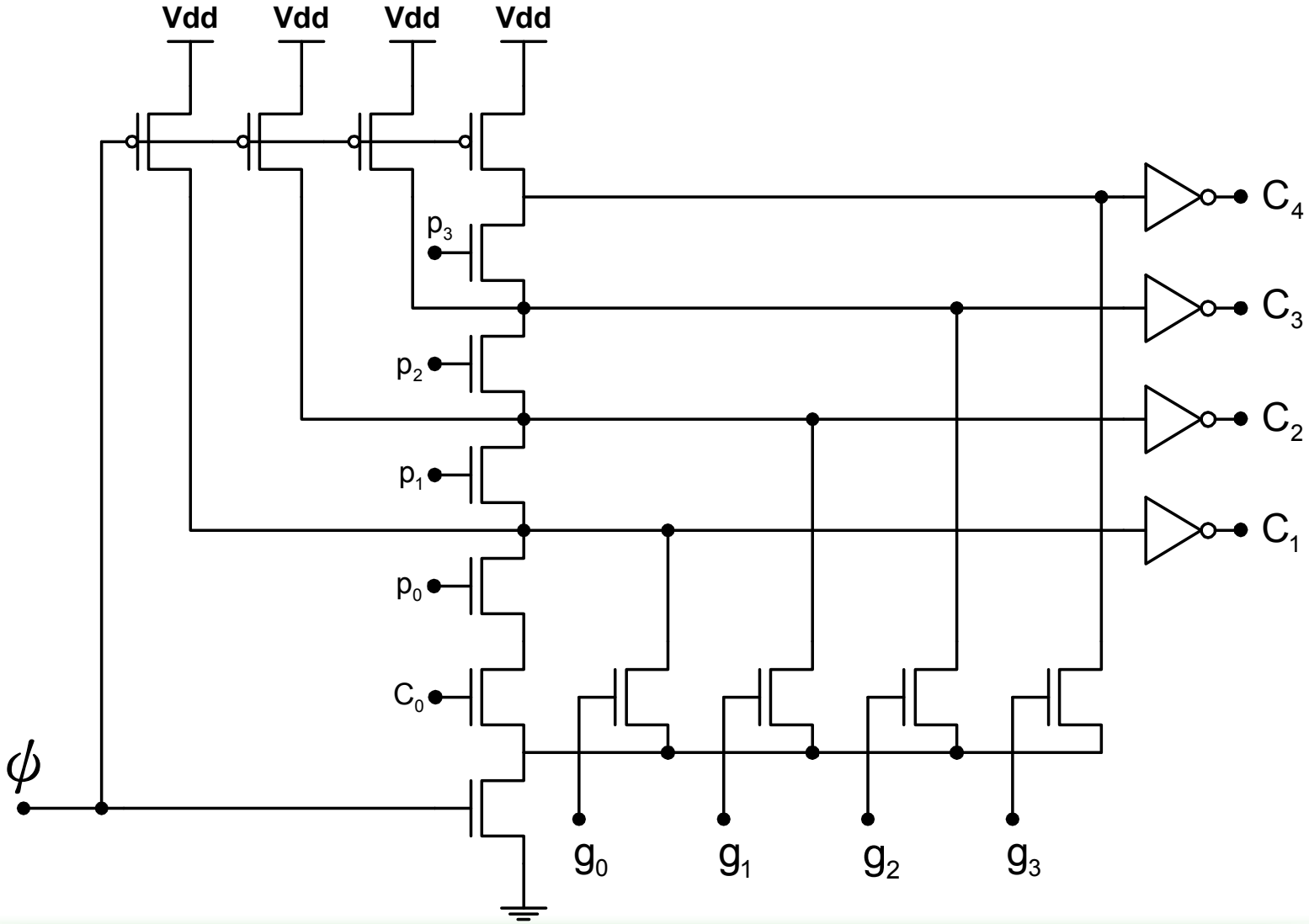
$C_2 = g_1 + P_1\, (g_0 + P_0\, C_{in})$

$C_3 = g_2 + P_2\, (g_1 + P_1\, (g_0 + P_0\, C_{in}))$

$C4 = g_3 + P_3\, (g_2 + P_2\, (g_1 + P_1 (g_0 + P_0 C_{in})))$



CLA 4-bit Parallel Adder



MODL CLA Carry-Generation Circuit

5.3 CMOS Differential Logic

- Cascode Voltage Switch Logic (CVSL)
- Invented by IBM
- *Ref: VLSI Design Oct. 1984, PP.78-86*

5.3.1 Differential Cascode Voltage Switch Logic (DCVS)

Ref : [1] ISSCC Dig. Tech. Papers, Feb 1984, PP 16-17

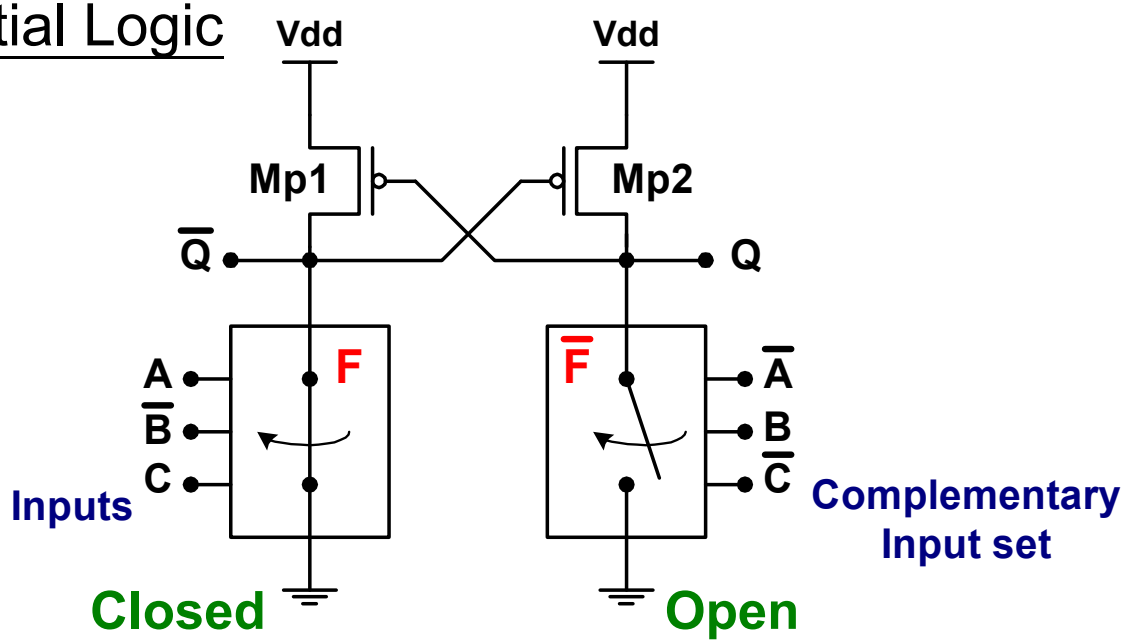
[2] VLSI Design Oct. 1984, PP. 78-86

[3] IEEE JSSC, vol. SC-22, PP. 528-532 ,Aug ,1987

[4] IEEE JSSC, vol. SC-21, PP. 1082-1087, Dec ,1986

(A) *Basic Structure* :

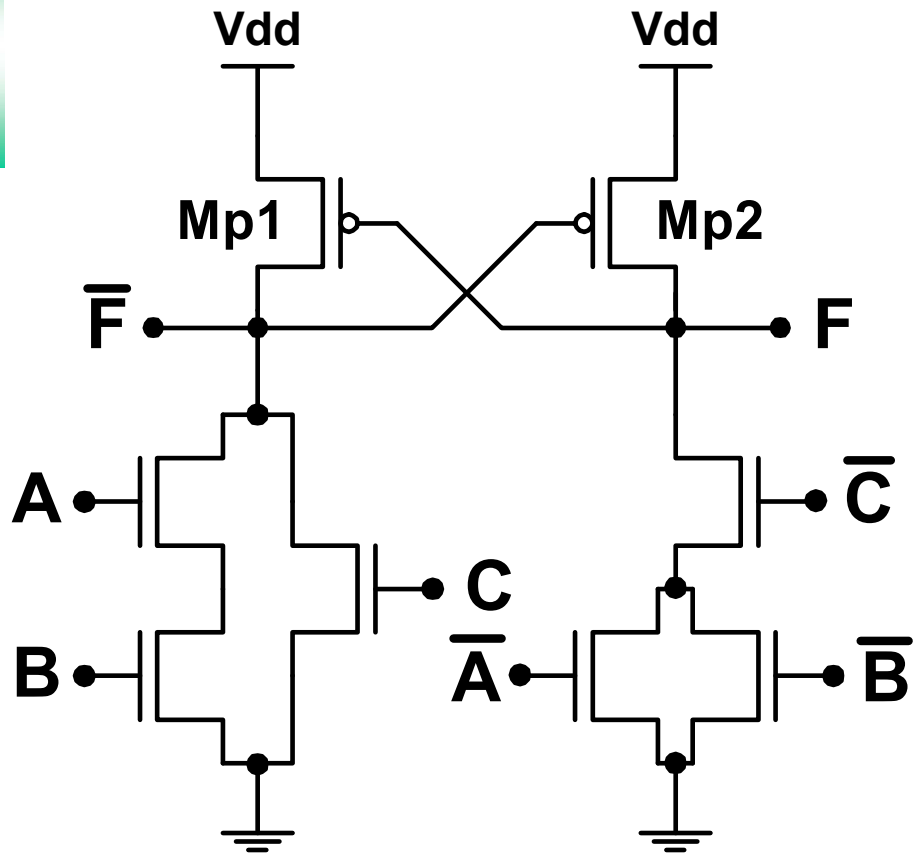
- Two cross-coupled PMOS transistors (Mp1 and Mp2) Form the latch
- The logic is performed by NMOS arrays
  ➔ Both true and complement inputs are necessary
- Cascode transistor circuits : called TREE
- dual-rail logic ➔ Differential Logic



Example - 1 :

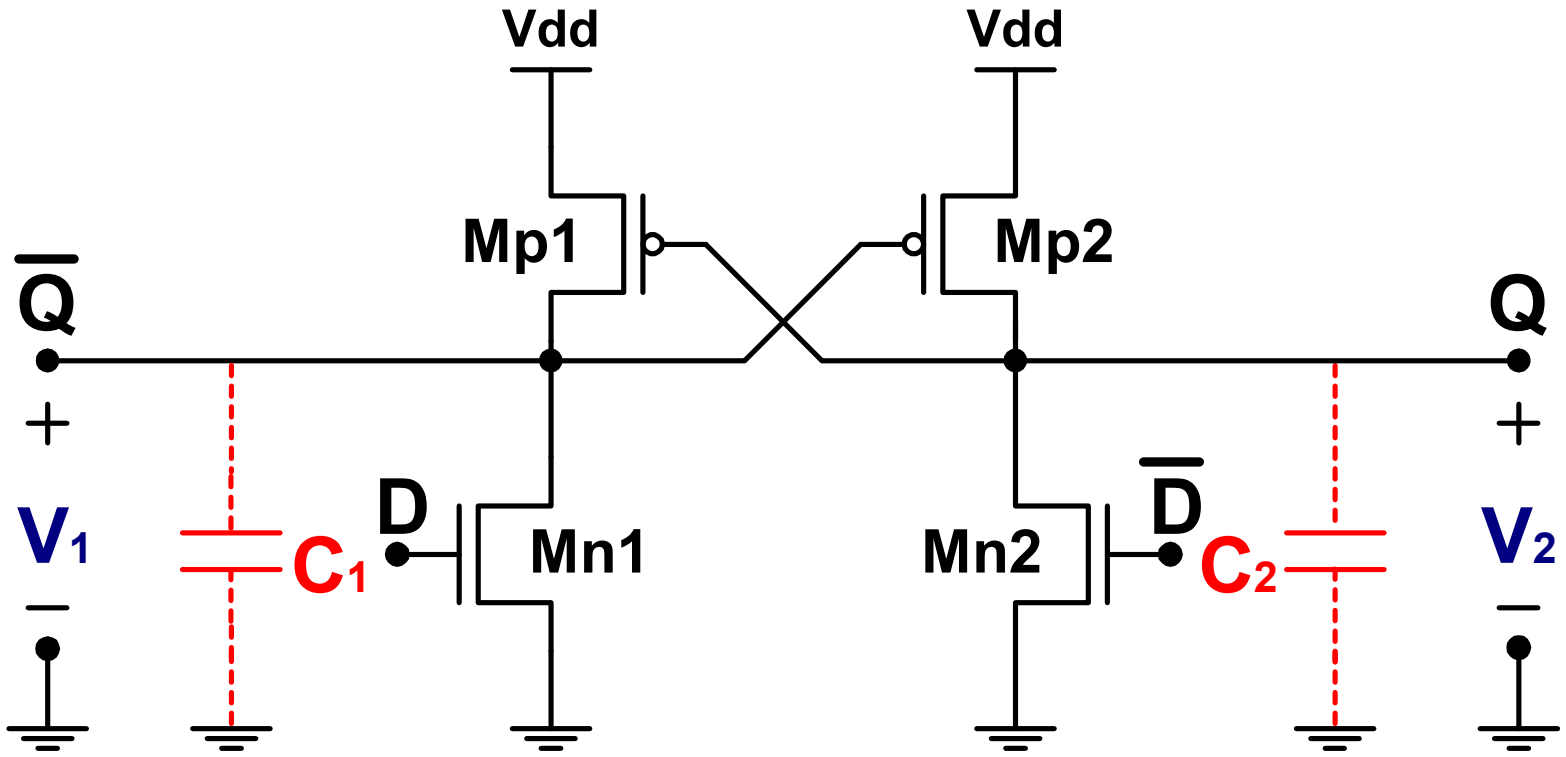
$$F = AB + C$$

$$\overline{F} = \overline{AB + C}$$



Example - 2 :

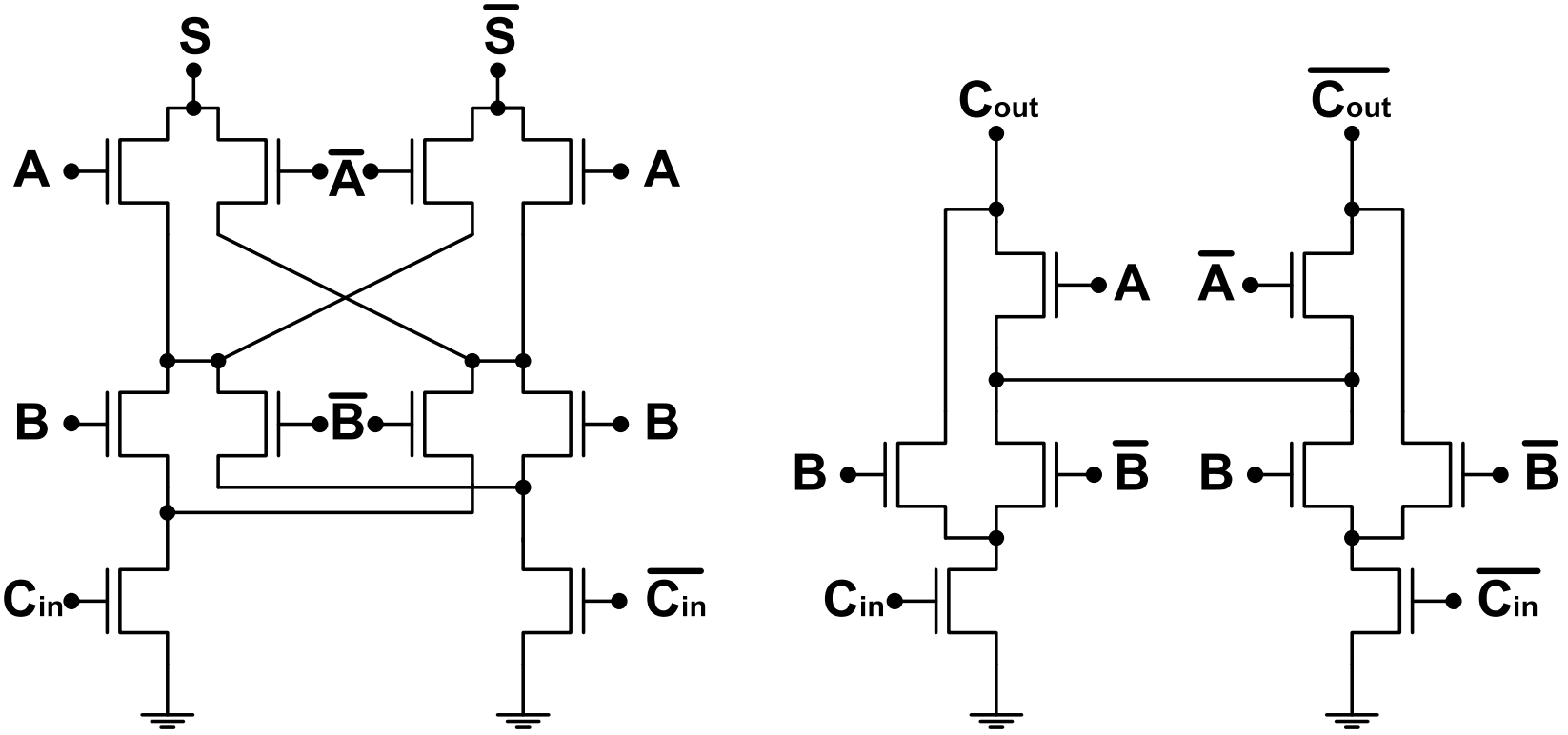
DCVSL Latch



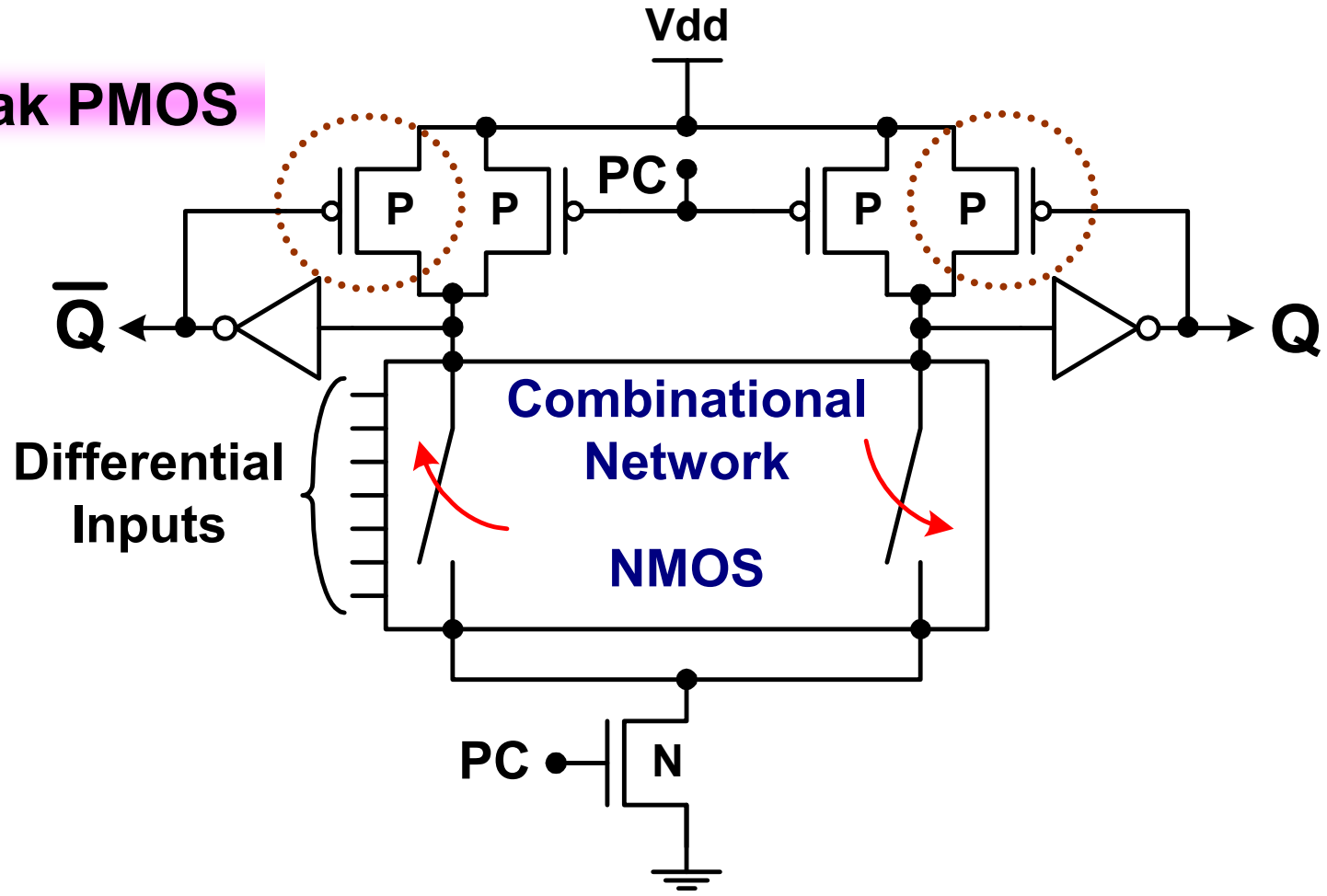
Example - 3 : DCVSL Full Adder

$S=A \oplus B \oplus C$

$C_{out}=AB+BC+AC$



(B) *Clocked DCVS (1/2)* :



(B) *Clocked DCVS (2/2)* :

- **Advantages** :

  Can implement any logic functions because of double rail logic (Differential outputs)

- **Disadvantages** :

  (1) Large current spikes during switching (static logic)

  (2) Large chip area

  (3) Slow

- **How to IMPROVE THE OPERATION SPEED OF THE DIFFERENTIAL LOGIC ?**

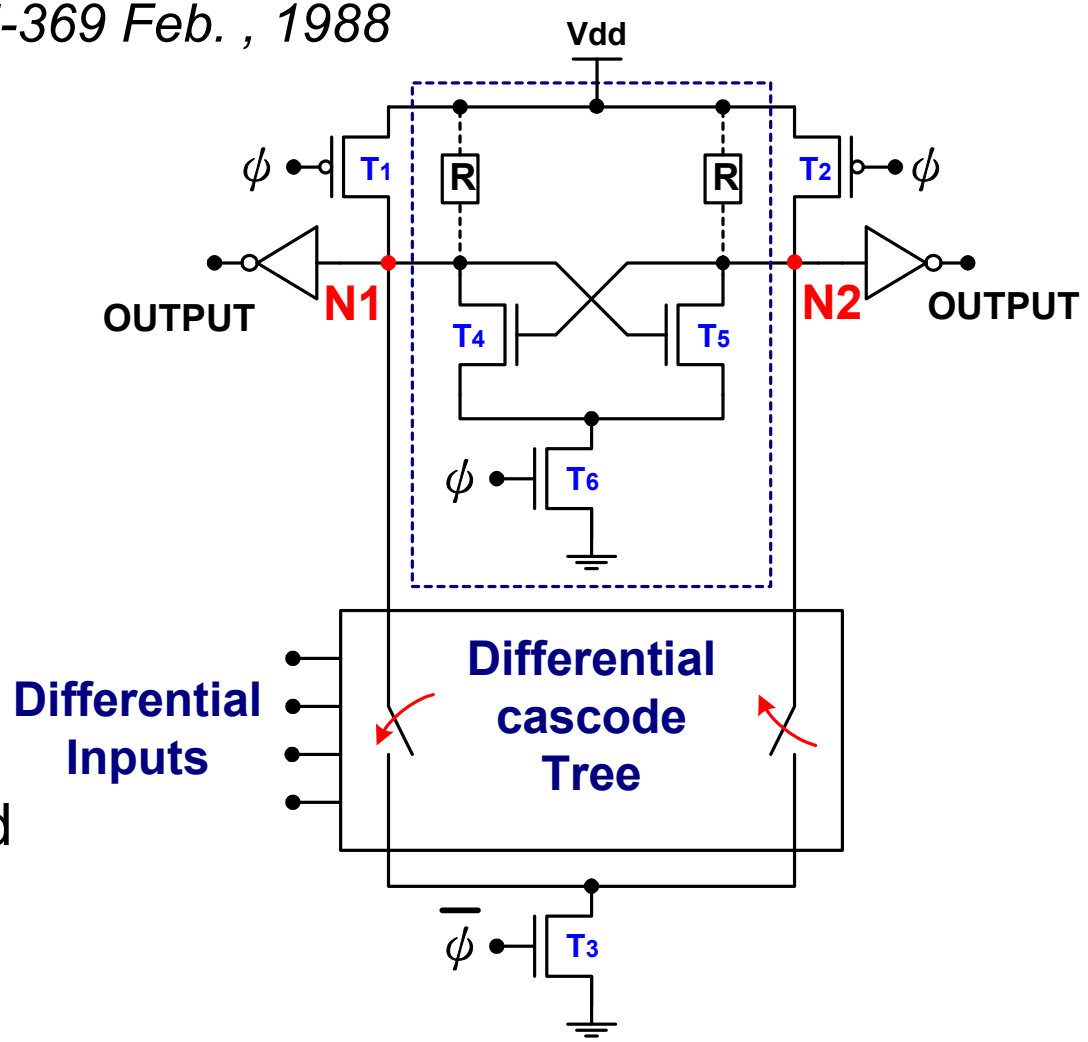
(1) Using SENSE AMP to improve the discharging speed .

(2) limited the Logic Voltage Swing .

5.3.2 Sample – Set Differential Logic (SSDL) (1/2)

Ref : IEEE JSSC Vol-21 PP.367-369 Feb. , 1988

- **ψ**= *0* (**$\bar{\psi}$**= *1*) : *Sample*
  T1, T2, and T3 are ON
  N1 and N2 have a Voltage difference
  (300 mV – 400 mV)
- **ψ**= *1* (**$\bar{\psi}$**= *0*) : *Set*
  T1, T2, and T3 are OFF
  T6 : ON
- Latch operation
  (N1 (or N2) is discharged quickly by sense amp.)



5.3.2 Sample – Set Differential Logic (SSDL) (2/2)

TABLE 1

Comparison results for the static CMOS , CVSL , and SSDL circuits (*) 20 Without Resistive load elements

Circuit Type	Device Count	Active Area (um X um	Time per Evaluatio (ns)	Power Dissipation (mW)
Static CMOS	28	420 X 3	10	1.75
CVSL	17	240 X 3	12	1.87
SSDL	22 *	250 X 3	8	3.72

TABLE 2

Speed of operation versus number of transistors in series in the NMOS tree network

Circuit Type	Transistor in Series	Relative Time to Change Buffer Output
Domino &CVSL	3	1. 0
	5	1. 3
	7	1.5
SSDL	3	1. 0
	5	1. 0
	7	1

- Allow the use of several transistors in series or parallel without significant speed degradation.
- Power dissipation↑.
- Relaxed dimension ratio and charge-redistribution design constraint .
- Stage number↓.
- true-two-phase clocking strategy.

5.3.3 Differential Split – Level (DSL) circuit (1/2)

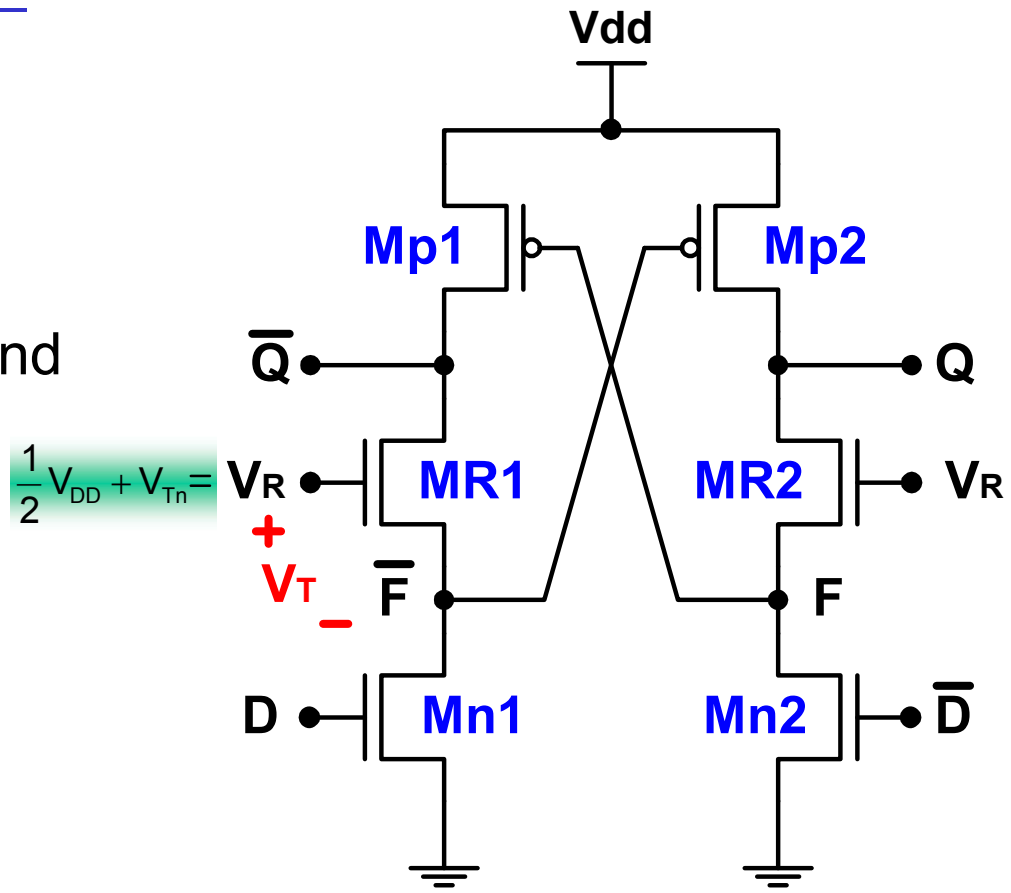
Ref : IEEE JSSC vol. SC-20 no.2 PP. 531-536, April, 1985

DSL Basic Structure

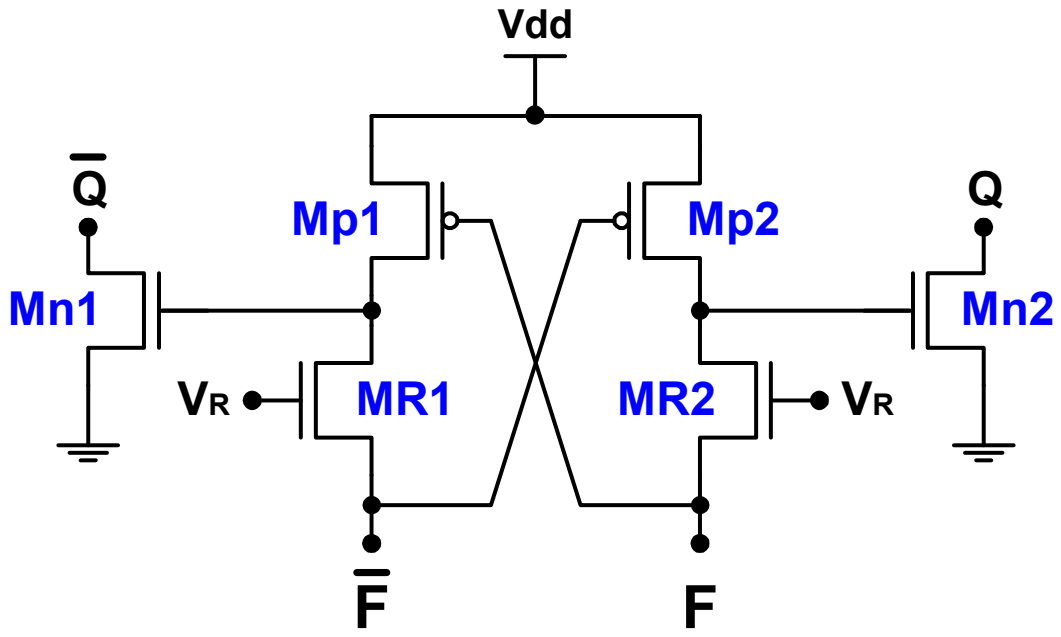
- Can be viewed as a modified DCVS logic circuit
- Two NMOSs MR1 and MR2 are added between Q ($\overline{Q}$) and F ($\overline{F}$)

➔ this clamps the voltage on nodes F and $\overline{F}$ to (V_{DD} / 2)

➔higher operation speed (∵ quickly discharge)



5.3.3 Differential Split – Level (DSL) circuit (2/2)



Advantages	Disadvantages
(1) high Speed (due to small swing) (2) Complementary output (3) No large current spikes	(1) Large chip area (2) Small swing (3) dc power dissipation (4) skew problem

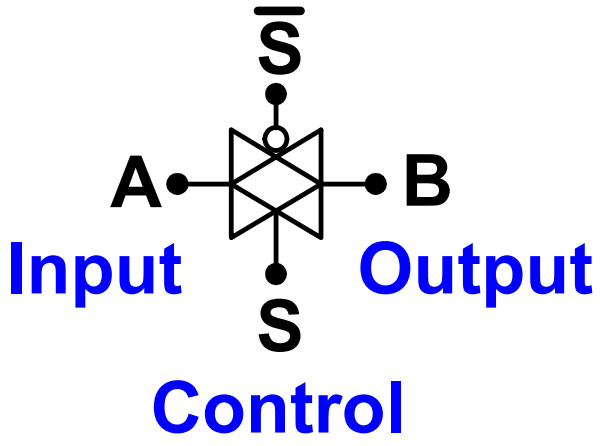
5.4 Pass - Transistor Logic Families

- Several pass - transistor logic families , for logic design , have been proposed for improving the speed of CMOS circuits , Such families are：

1) The conventional CMOS pass - transistor logic
2) The complementary pass - transistor logic (**CPL**)
3) The Dual Pass - transistor Logic (**DPL**)
4) The Swing Restored Pass - Transistor Logic (**SRPL**)

5.4.1 TG – Based Switch Logic Gates (1/6)

- Transmission gate logic is based on data path control



S=1: B ⟵ A
S=0: TG is OFF
(High - imp.)

Transmission - Gate Logic

5.4.1 TG – Based Switch Logic Gates (2/6)

- **Path Select**

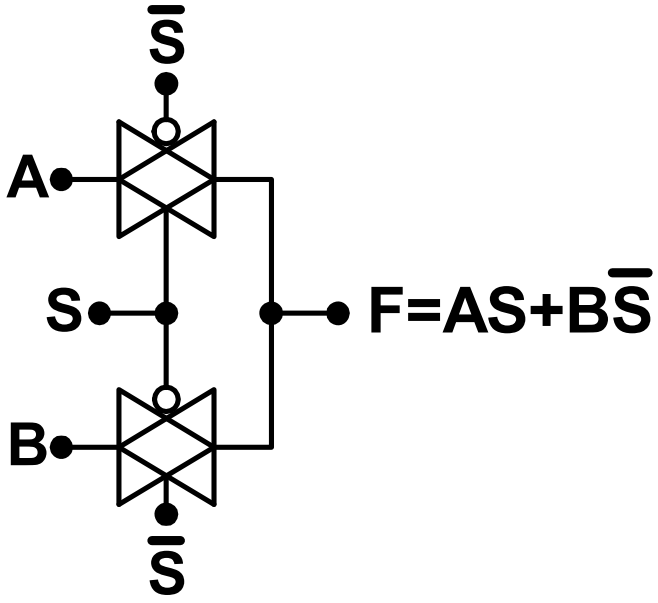
$$F = AS + B\overline{S}$$

This can be used as a data path selector as seen by writing.

if S = 1 : F = A

if S = 0 : F = B

This is a special case of a multiplexer network

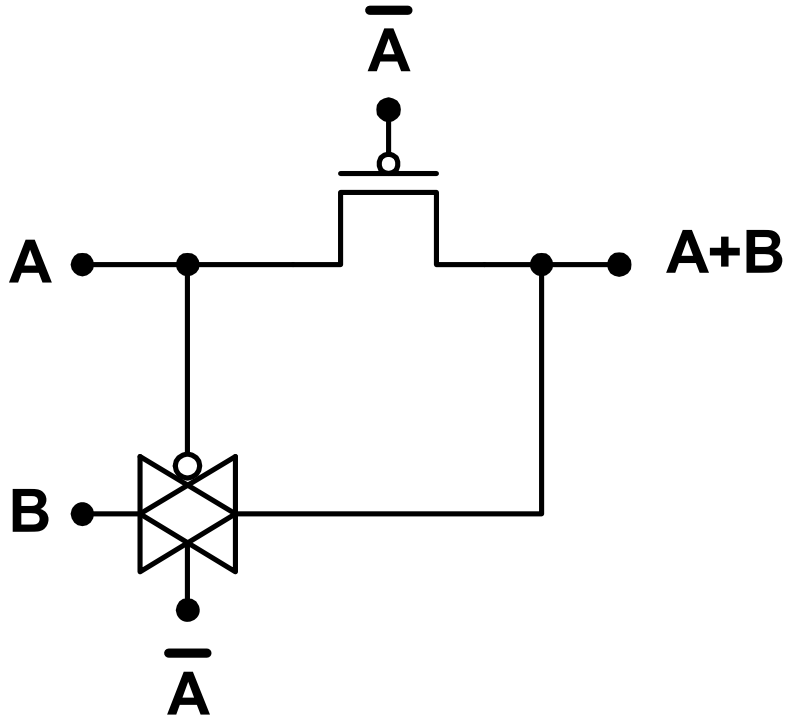


2 - Input TG Path Select

5.4.1 TG – Based Switch Logic Gates (3/6)

- OR Gate

$$F = A + \overline{A}B$$
$$= A + B$$

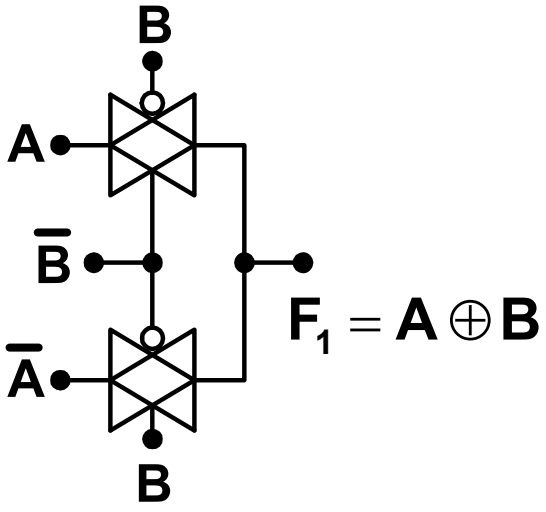


TG - Logic OR Gate

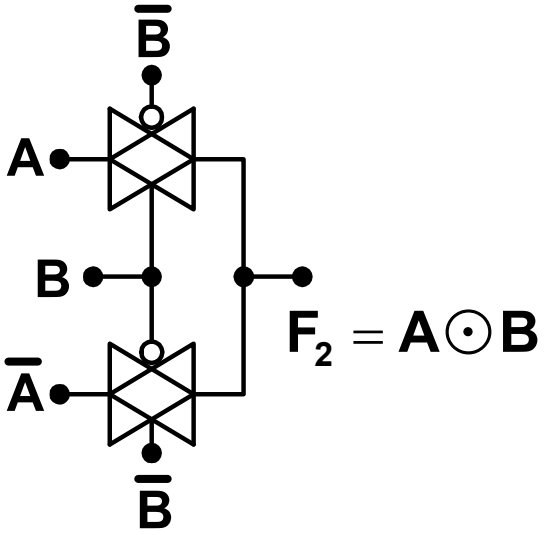
5.4.1 TG – Based Switch Logic Gates (4/6)

- **XOR , XNOR Gate**

$$F_1 = A \oplus B$$
$$= A\overline{B} + \overline{A}B$$
$$F_2 = A \odot B$$
$$= AB + \overline{A}\overline{B}$$



(a) XOR



(b) Equivalence

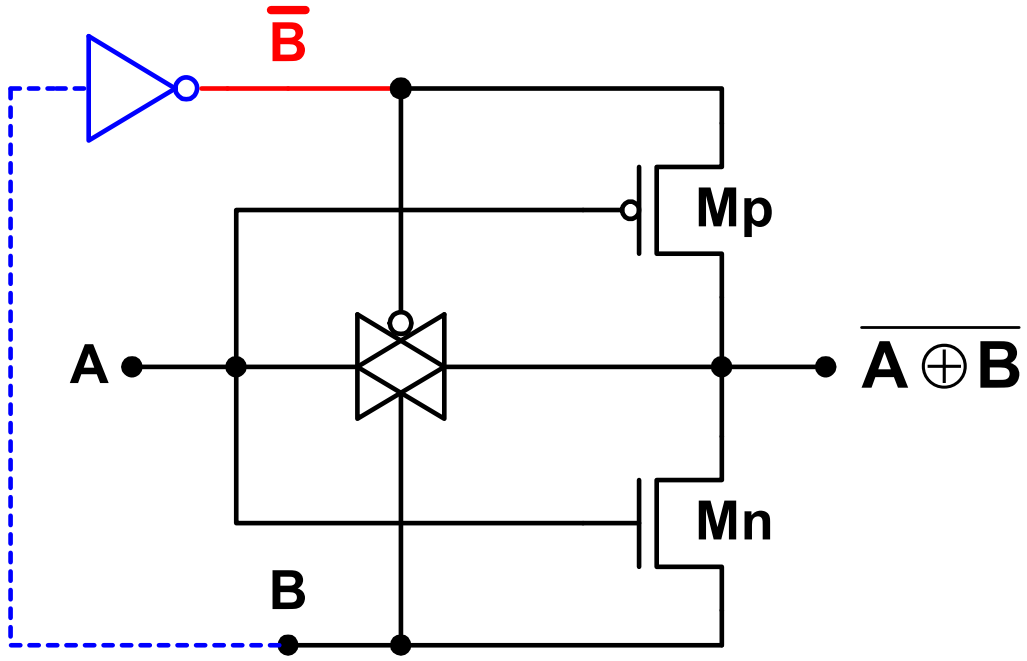
* Both A , $\overline{A}$, B , $\overline{B}$ are used

5.4.1 TG – Based Switch Logic Gates (5/6)

- **Alternate XNOR Gate**

  * Only A , B , $\overline{B}$ are used

  * Fewer MOS transistors

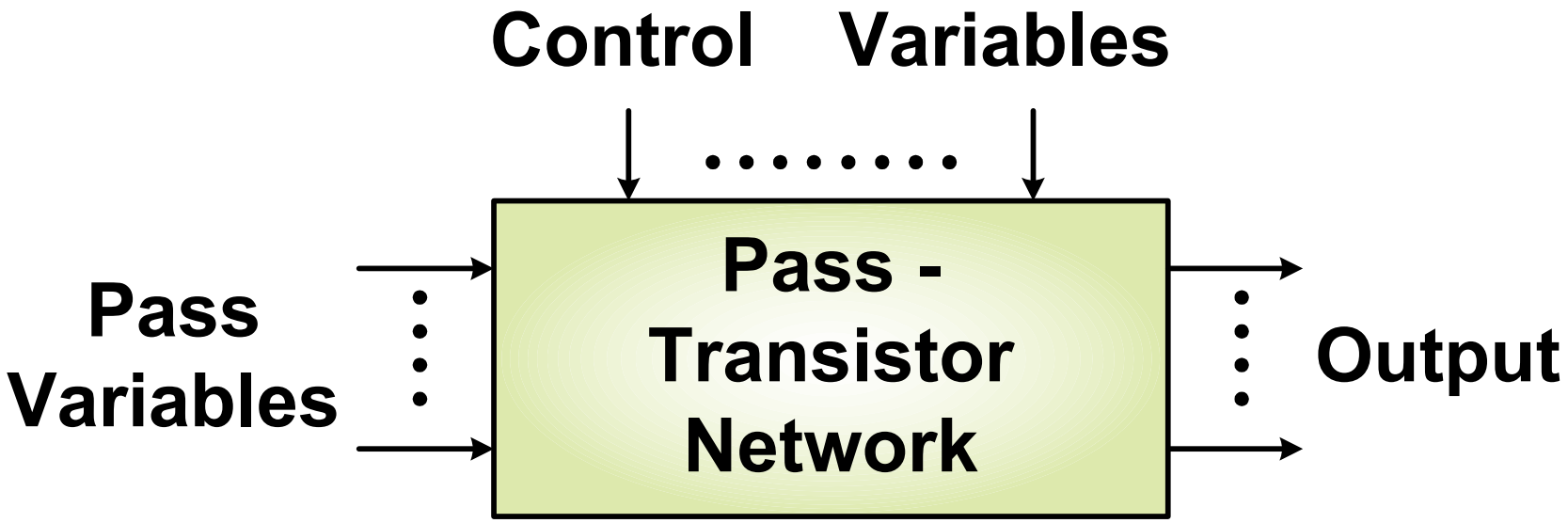


5.4.1 TG – Based Switch Logic Gates (6/6)

- Compared with the traditional logic design method , pass transistor networks may achieve high density , low power , and in some case high performances.
- Designing pass transistor networks has been a Craft than a Science.

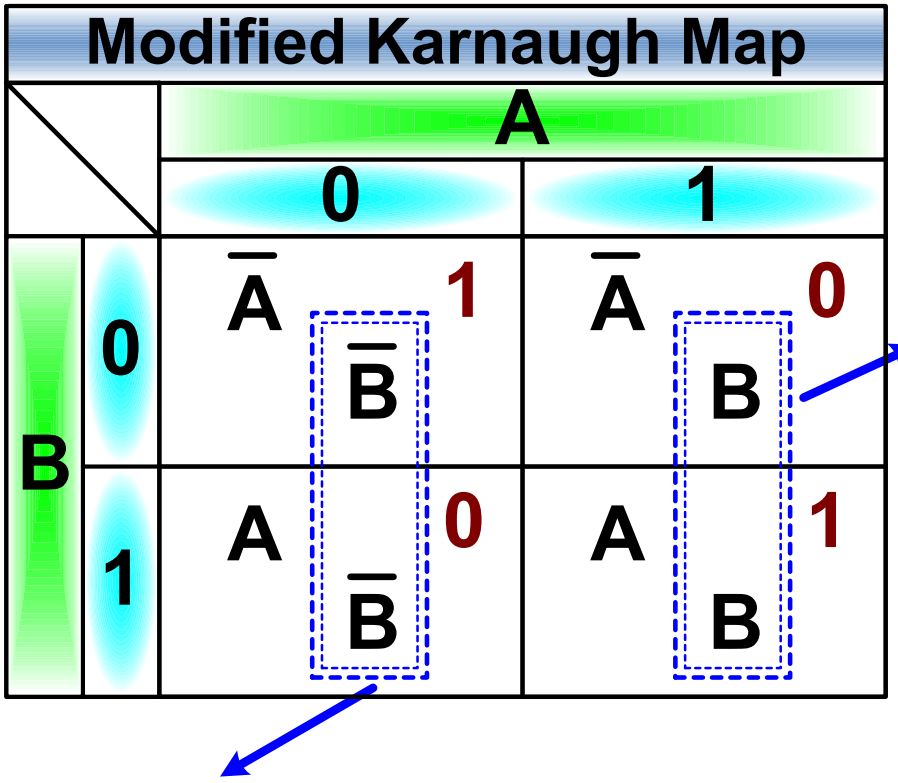
5.4.2 TG – Based Switch Logic Gates Design Method

- **Model for pass transistor logic**



5.4.2 TG – Based Switch Logic Gates Design Method

- **Example**：**A ⊙ B** (1/3)

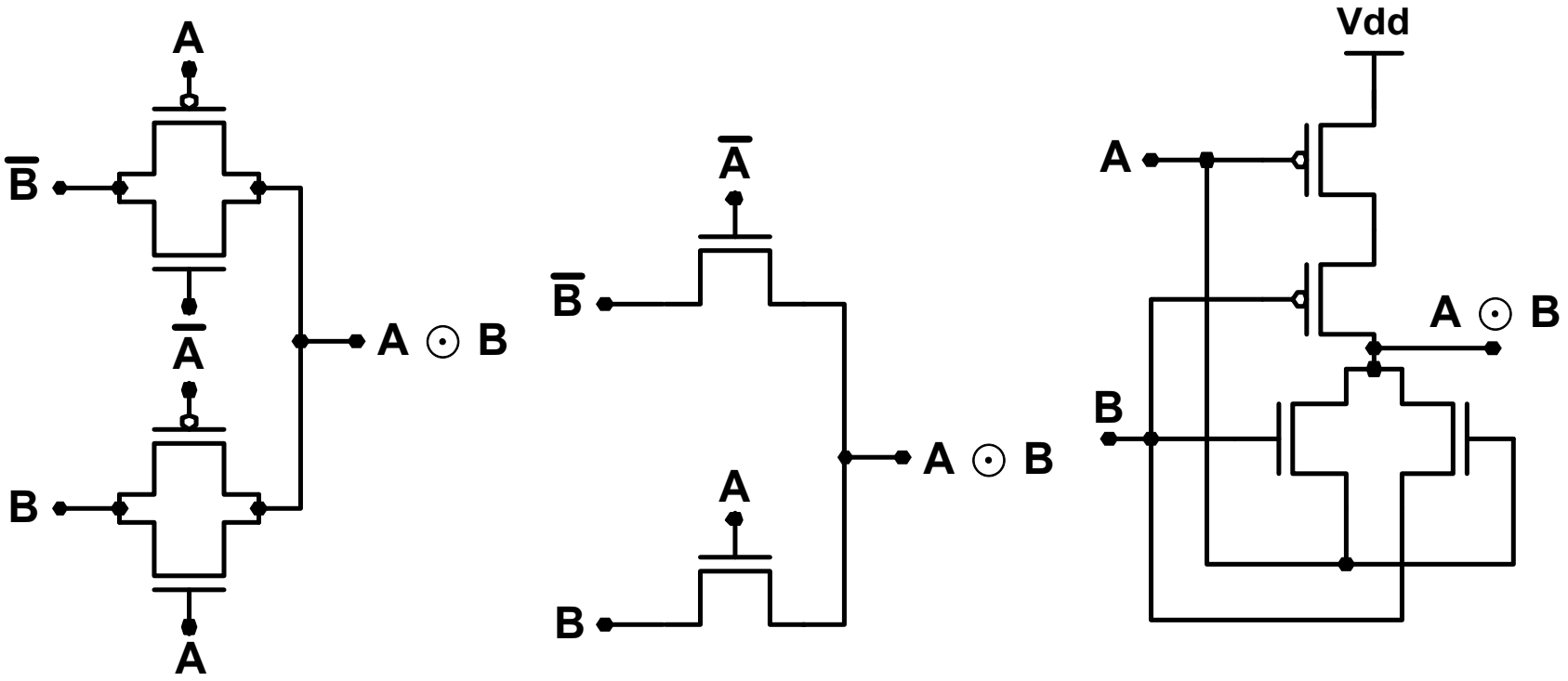


XNOR Turth Table

A	B	$\overline{A \oplus B}$	Pass Function
0	0	1	$\overline{A}+\overline{B}$
0	1	0	$A+\overline{B}$
1	0	0	$\overline{A}+B$
1	1	1	$A+B$

5.4.2 TG – Based Switch Logic Gates Design Method

- **Example**：**A ⊙ B** (2/3)

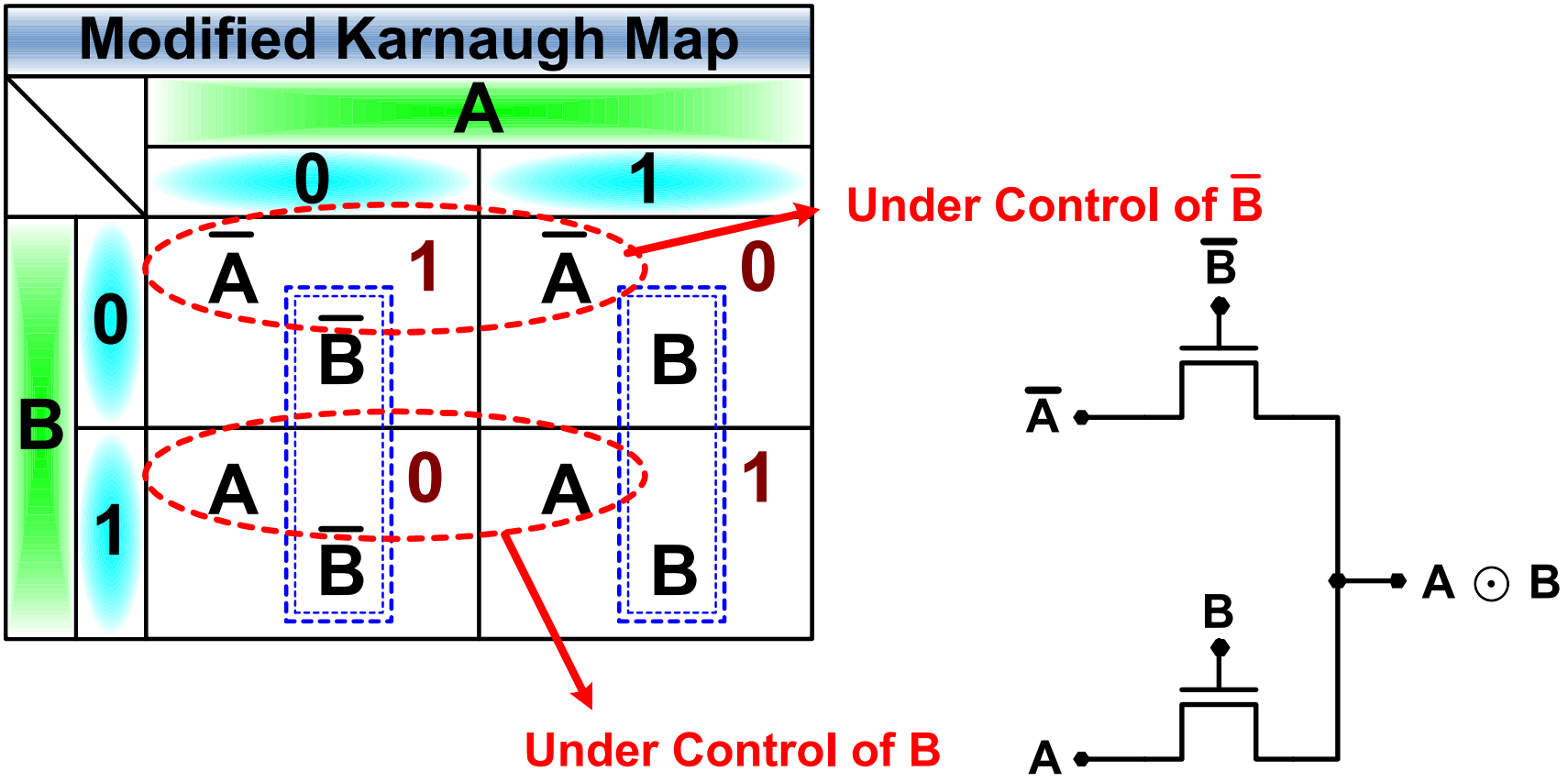


(a) complementary　　(b) signal - polarity　　(c) cross - coupled

Two - input XNOR gate implemented in pass - transistor logic

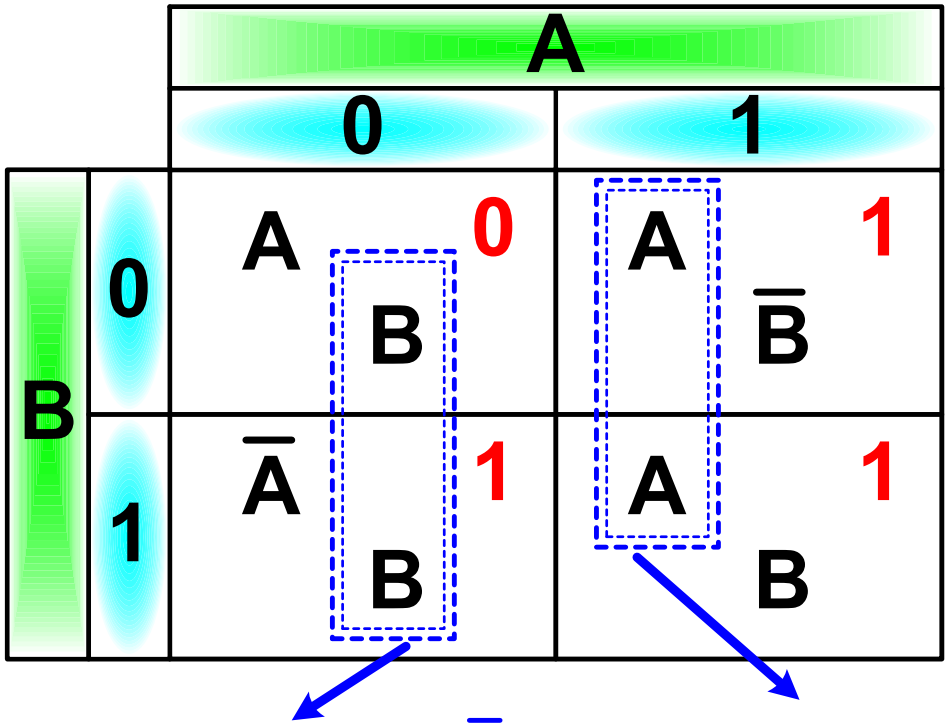
5.4.2 TG – Based Switch Logic Gates Design Method

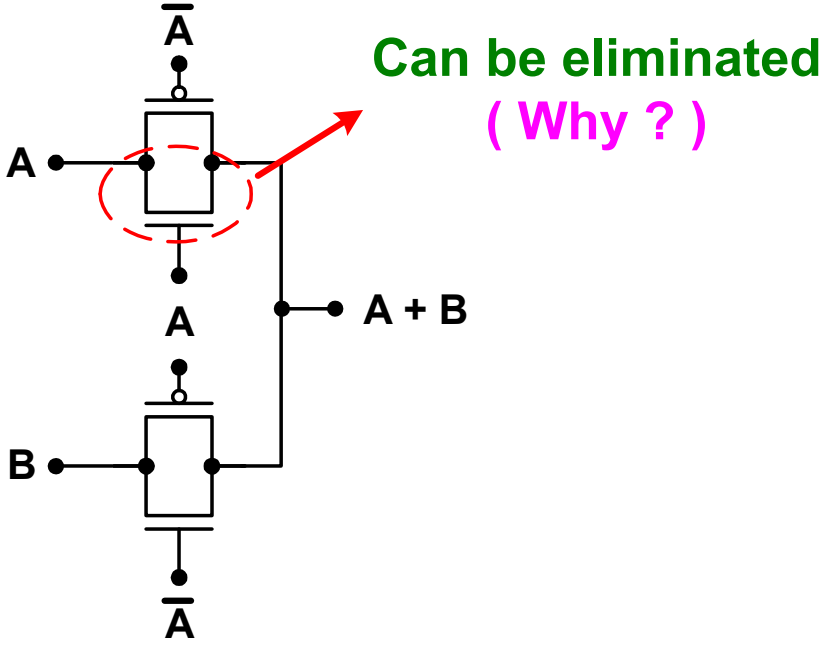
- **Example**：**A ⊙ B** (3/3)



5.4.2 TG – Based Switch Logic Gates Design Method

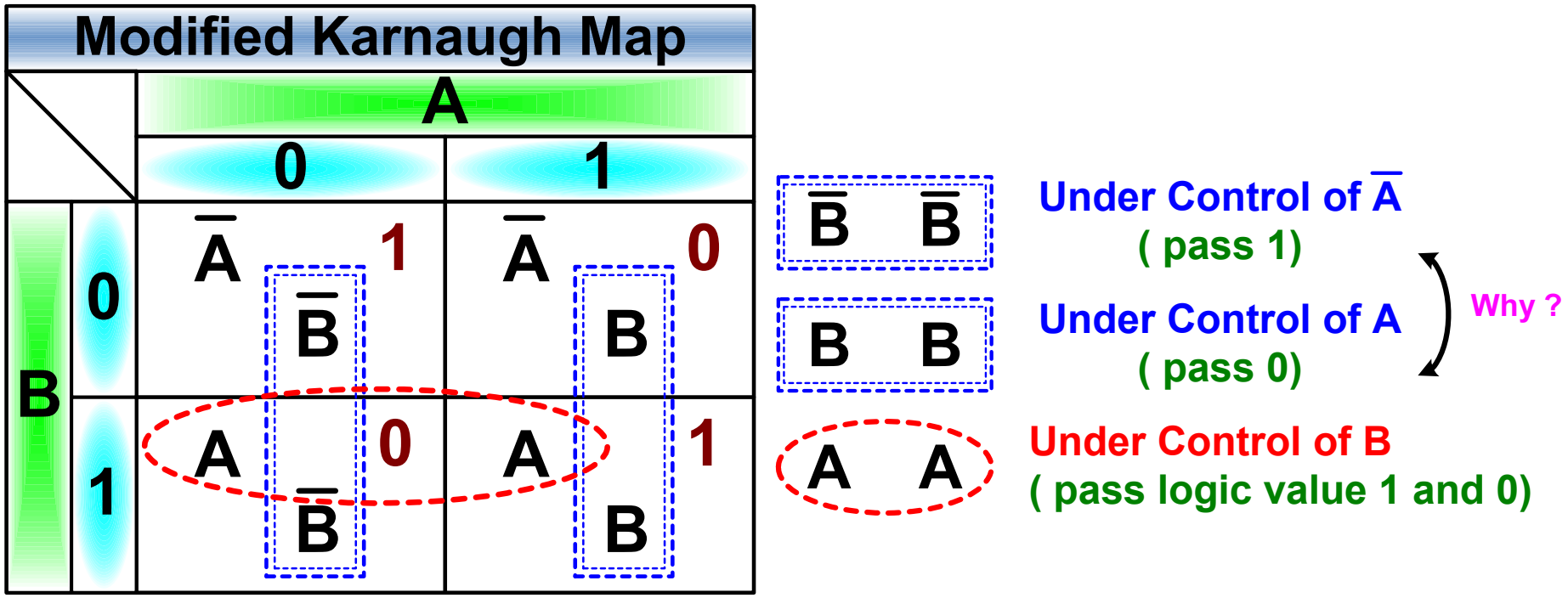
- **Example：OR Gate**





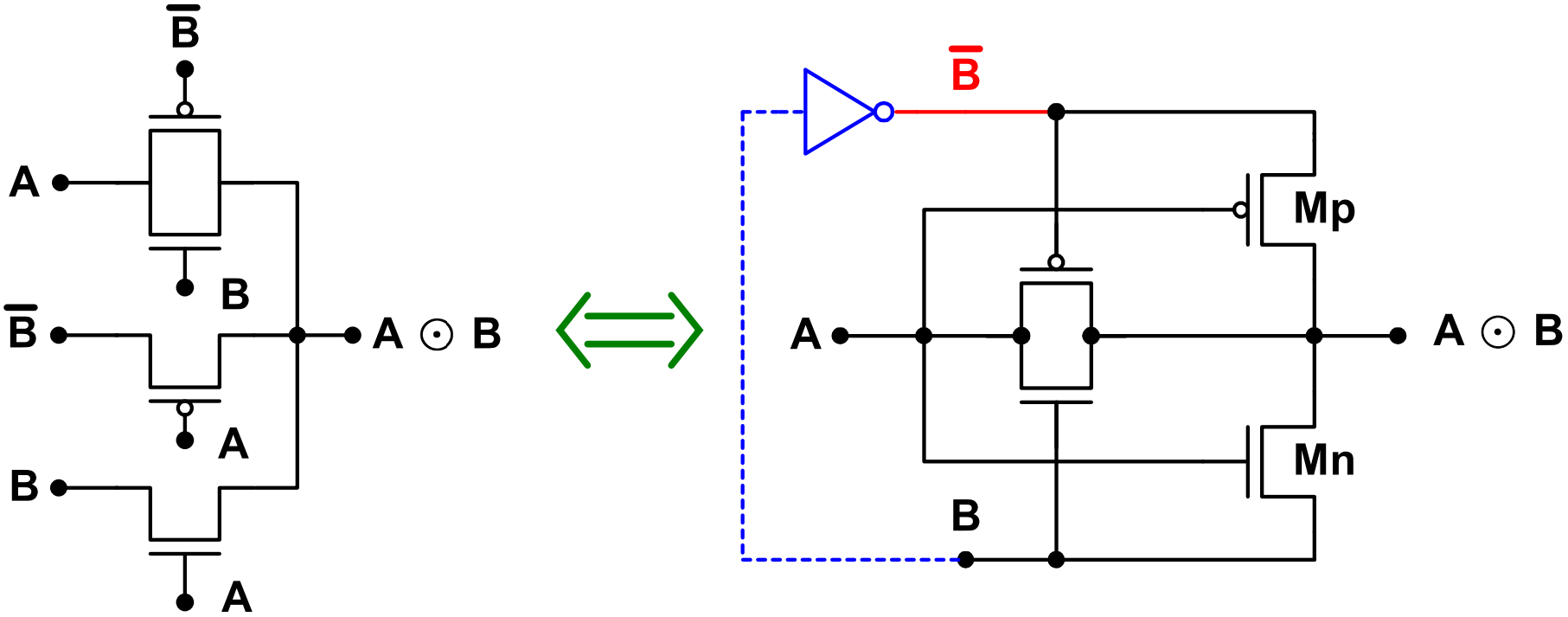
5.4.2 TG – Based Switch Logic Gates Design Method

- **Alternate XNOR Gate**：$A \odot B = AB + \bar{A}\bar{B}$ (1/2)



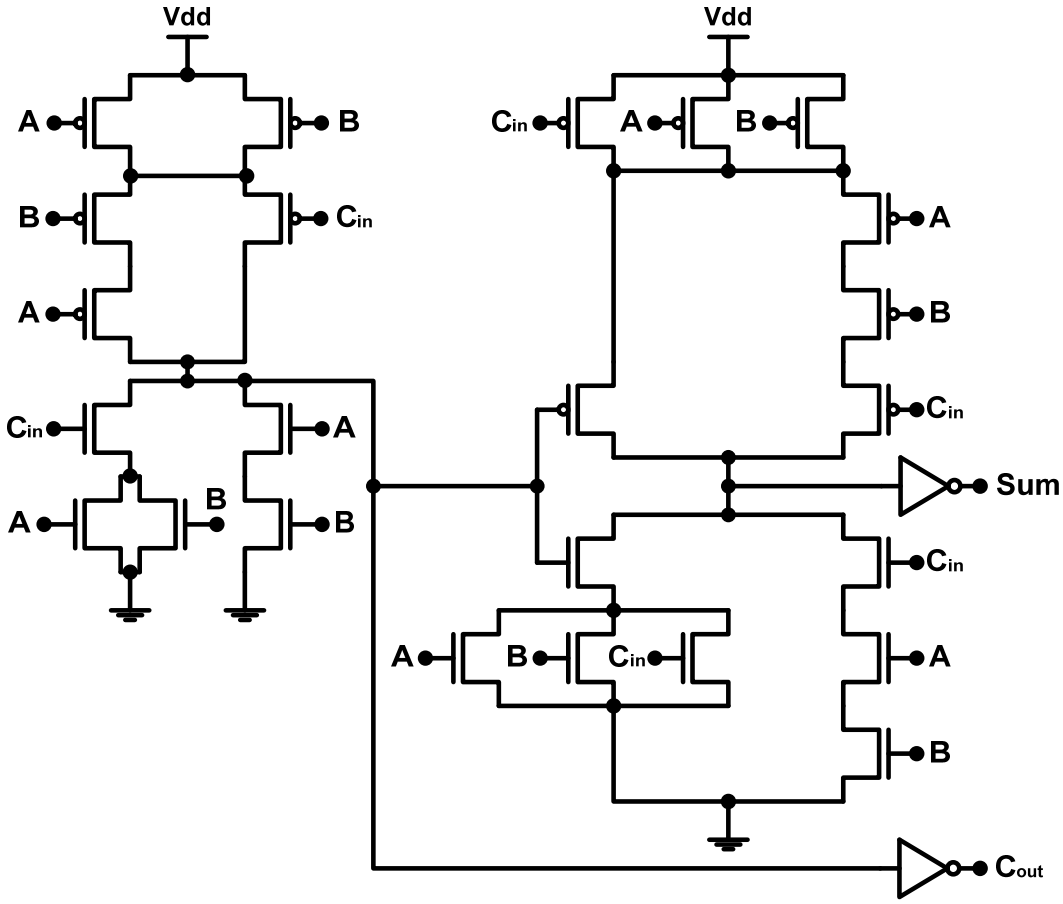
5.4.2 TG – Based Switch Logic Gates Design Method

- **Alternate XNOR Gate**：$A \odot B = AB + \bar{A}\bar{B}$ (2/2)



5.4.2 TG – Based Switch Logic Gates Design Method

- **Full Adder (1/2) :**

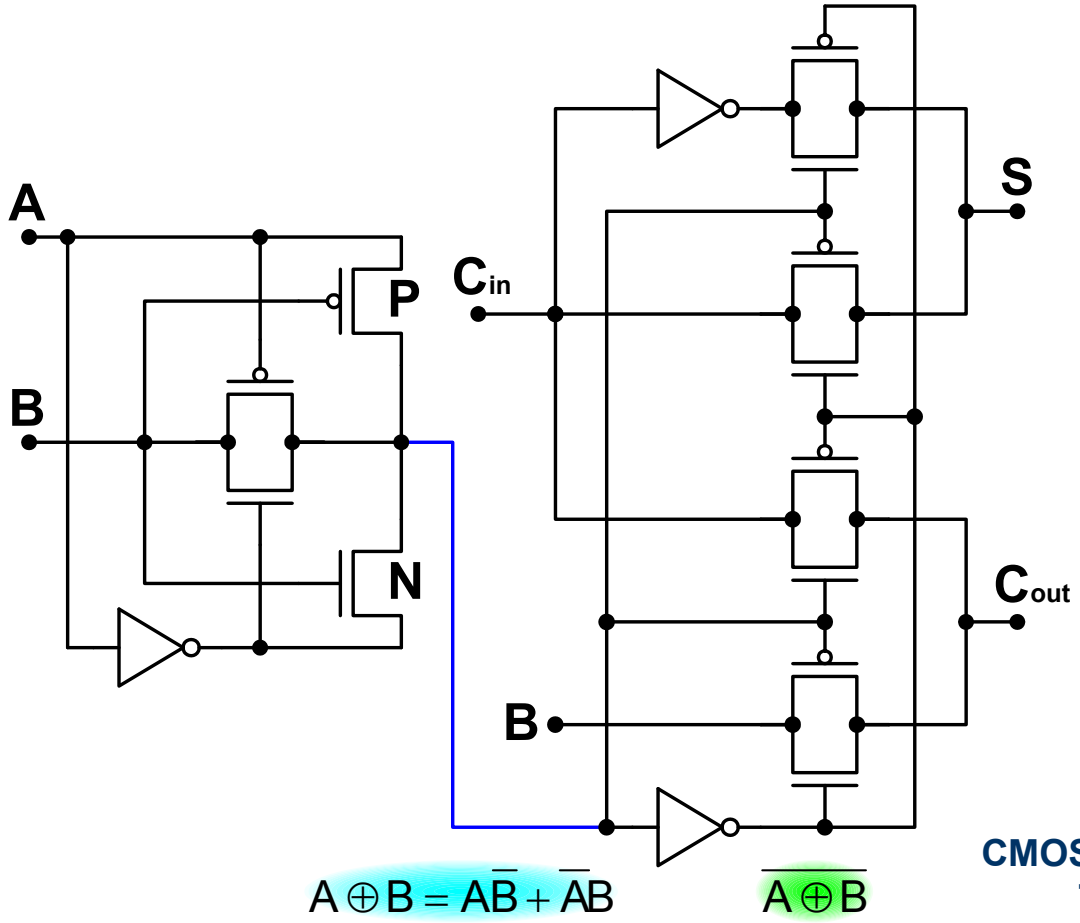


Adder Turth Table				
A	B	Cin	Sout	Cout
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1

CMOS static full adders using complementary style

5.4.2 TG – Based Switch Logic Gates Design Method

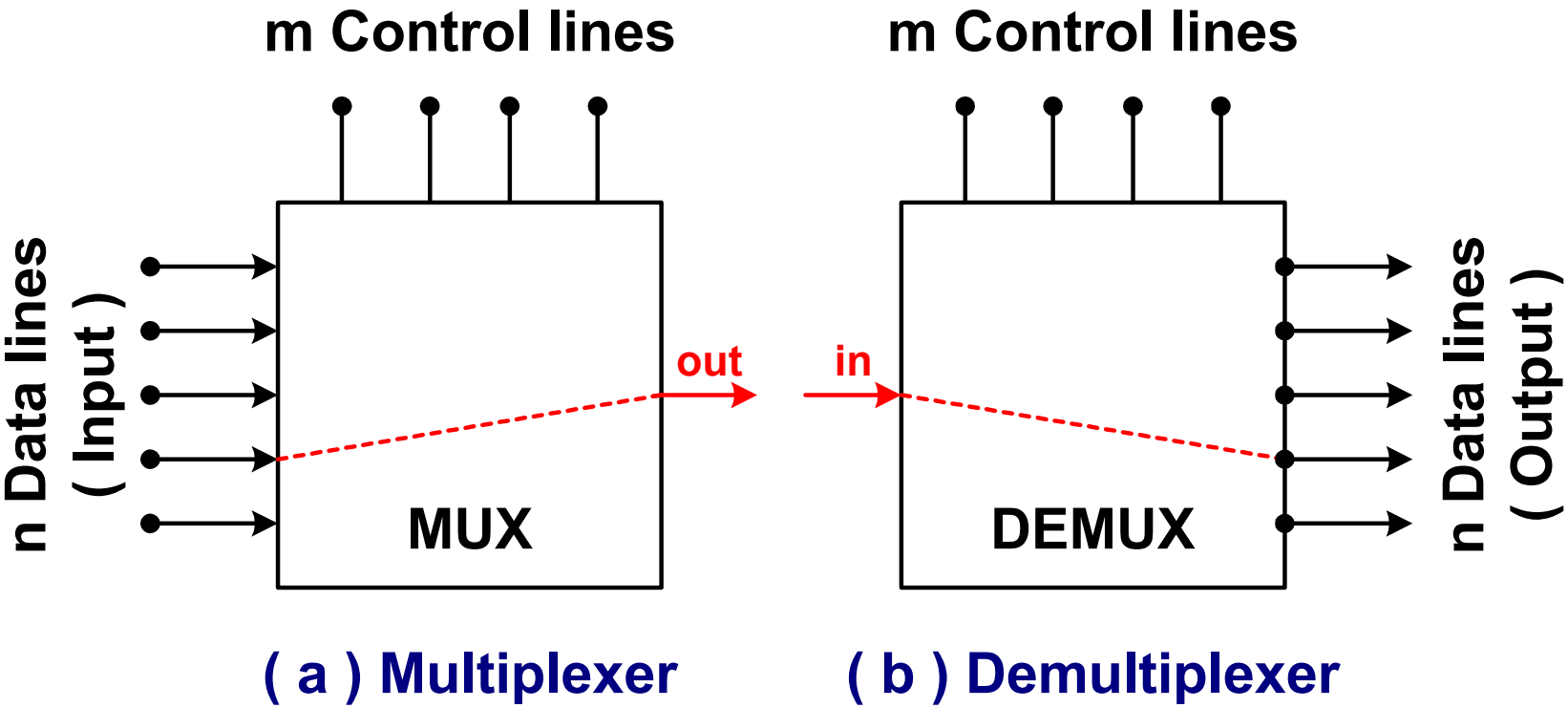
- **Full Adder** (2/2) :



CMOS static full adders using Transmission Gate

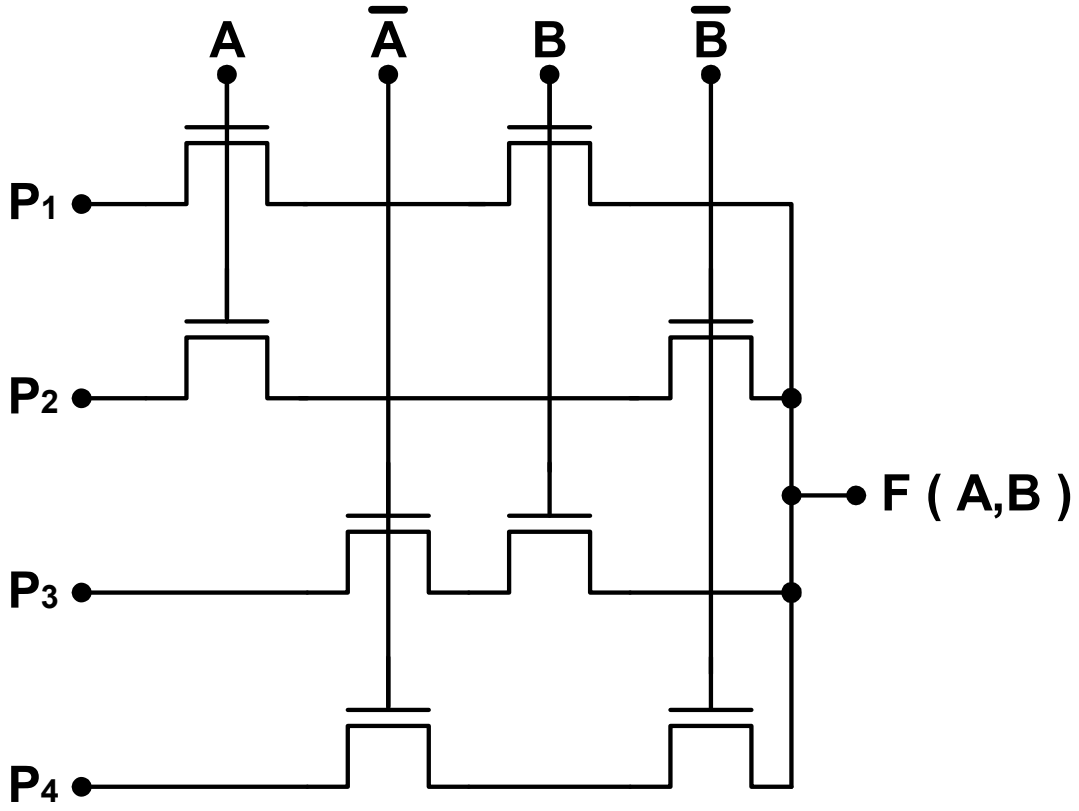
5.4.3 Pass – Transistor Array Logic (1/7)

- Multiplexer and Demultiplexer can be directly implemented using PG arrays



5.4.3 Pass – Transistor Array Logic (2/7)

- **(a) NMOS PTL**
  - $V_o \fallingdotseq V_{DD} - V_{TN}$
  - Multiplexer

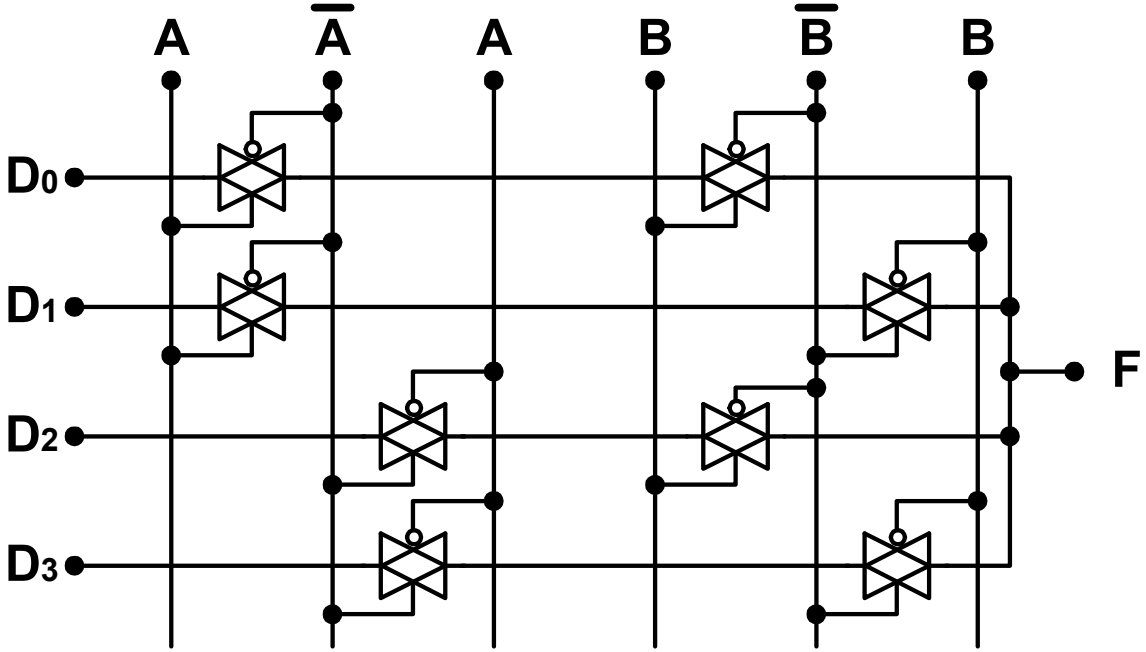


$$F = P_1 AB + P_2 A\overline{B} + P_3 \overline{A}B + P_4 \overline{A}\,\overline{B}$$

5.4.3 Pass – Transistor Array Logic (3/7)

(b) CMOS PTL (Ⅰ)

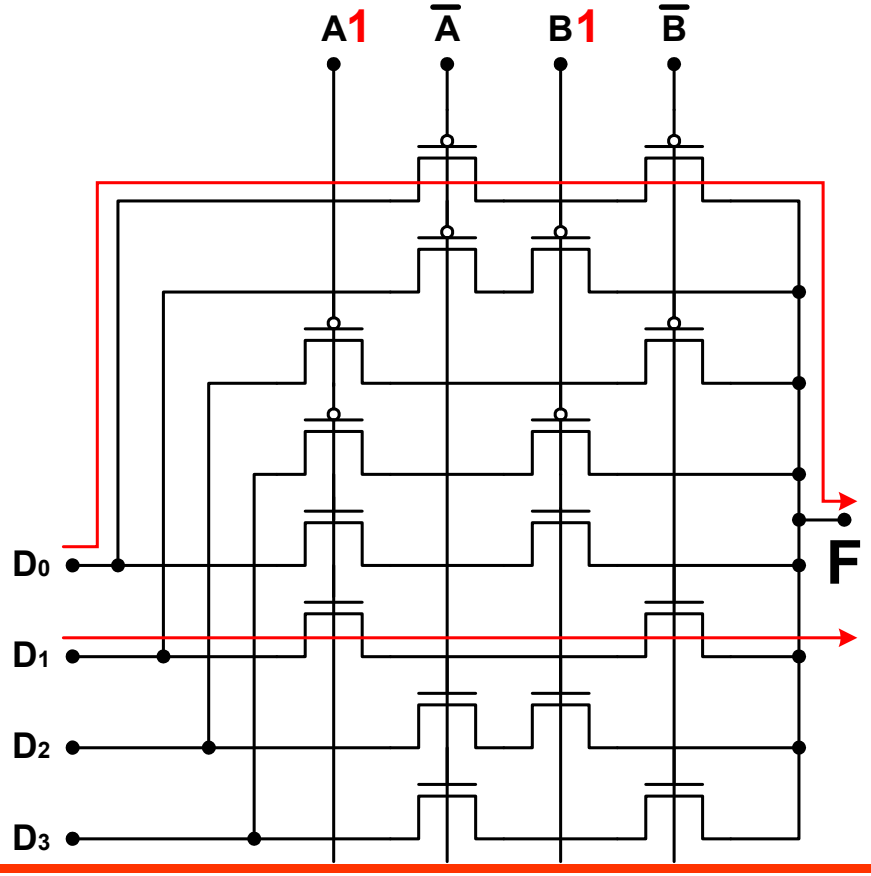
- $V_o = V_{DD}$
- Chip area ↑
- Faster than NMOS PTL



5.4.3 Pass – Transistor Array Logic (4/7)

(c) Split CMOS PTL (Ⅱ)

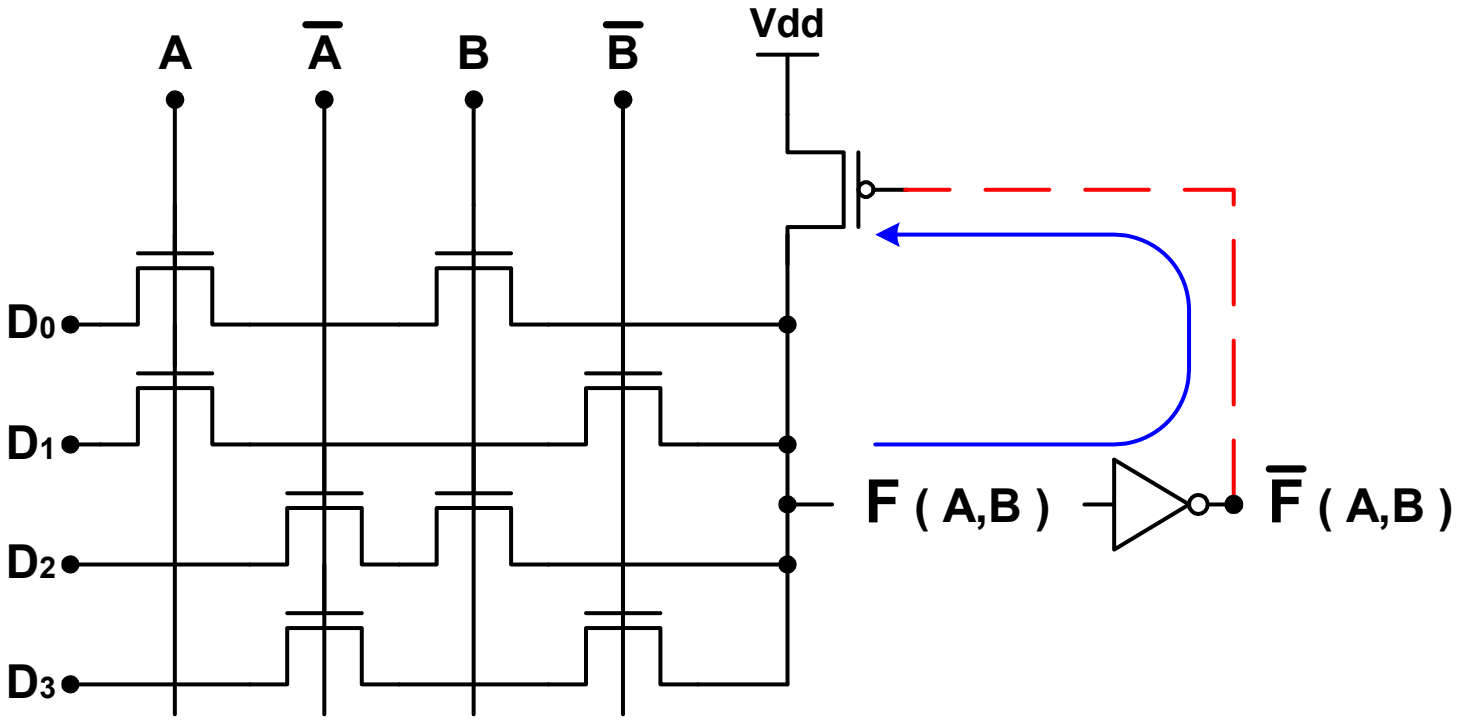
- Chip area is smaller than that of (b) CMOS PTL (Ⅰ)
  ∵ $n^+ - p^+$ contacts ↓
- $V_o = V_{DD}$



5.4.3 Pass – Transistor Array Logic (5/7)

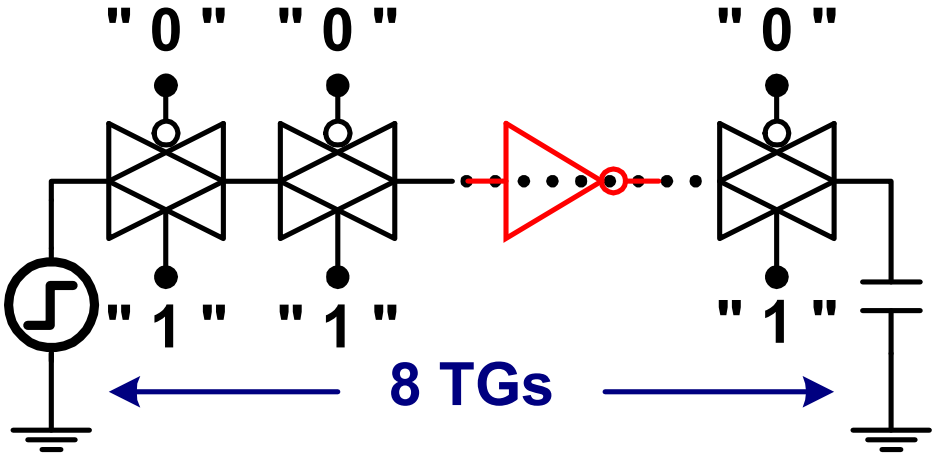
(d) CMOS PTL (Ⅲ)

- V_o restoring circuit ➔ $V_o \fallingdotseq V_{DD}$
- Speed can be improved



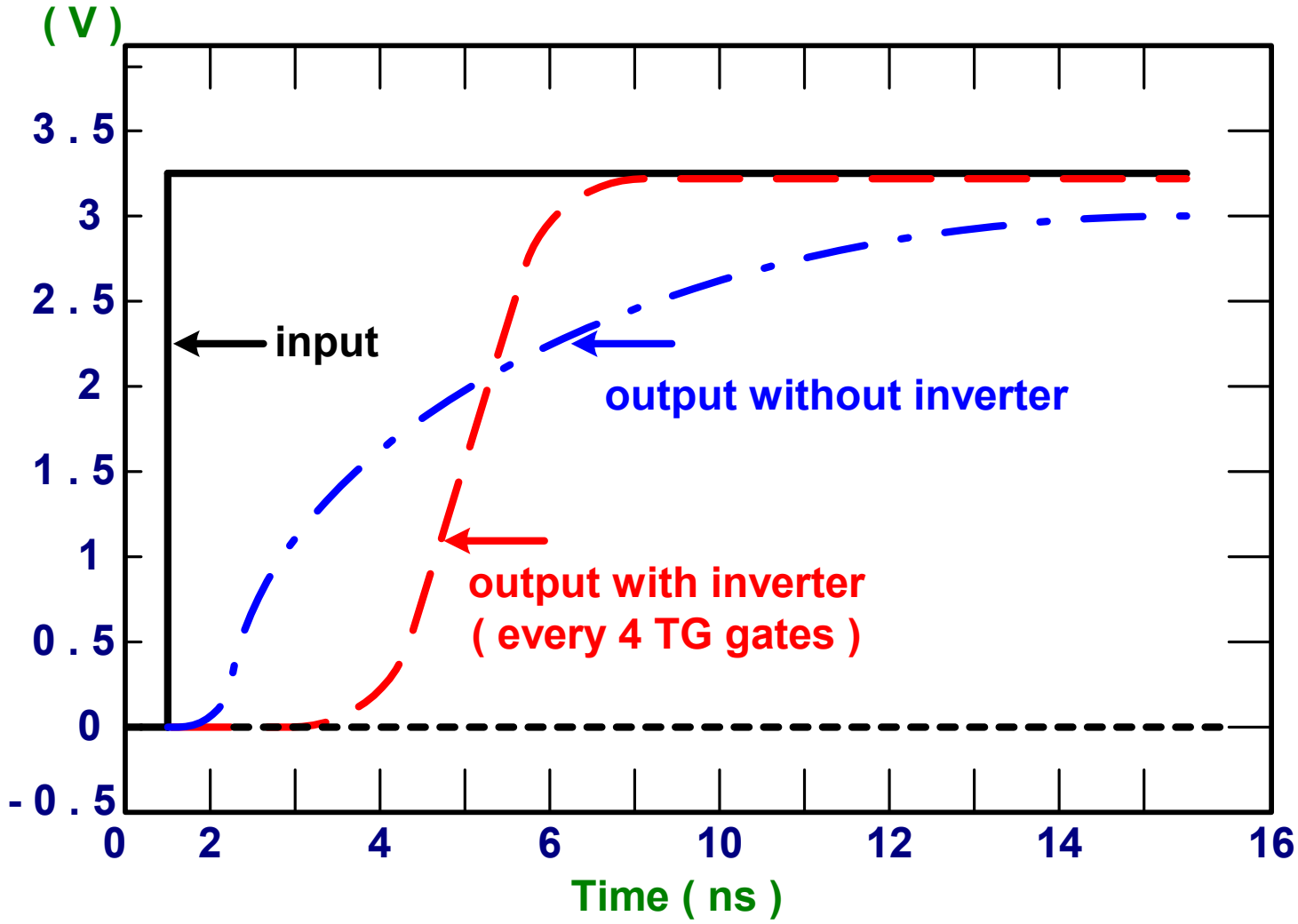
5.4.3 Pass – Transistor Array Logic (6/7)

- However , the Pass – Transistor (TG) logic circuit does not provide a restoration of the logic level , i.e. the logic gates are passive with no gain elements .
- The driving capability of the circuit is limited and delay increases with long TG chains .



An inverter stage is added every 4 TG stages

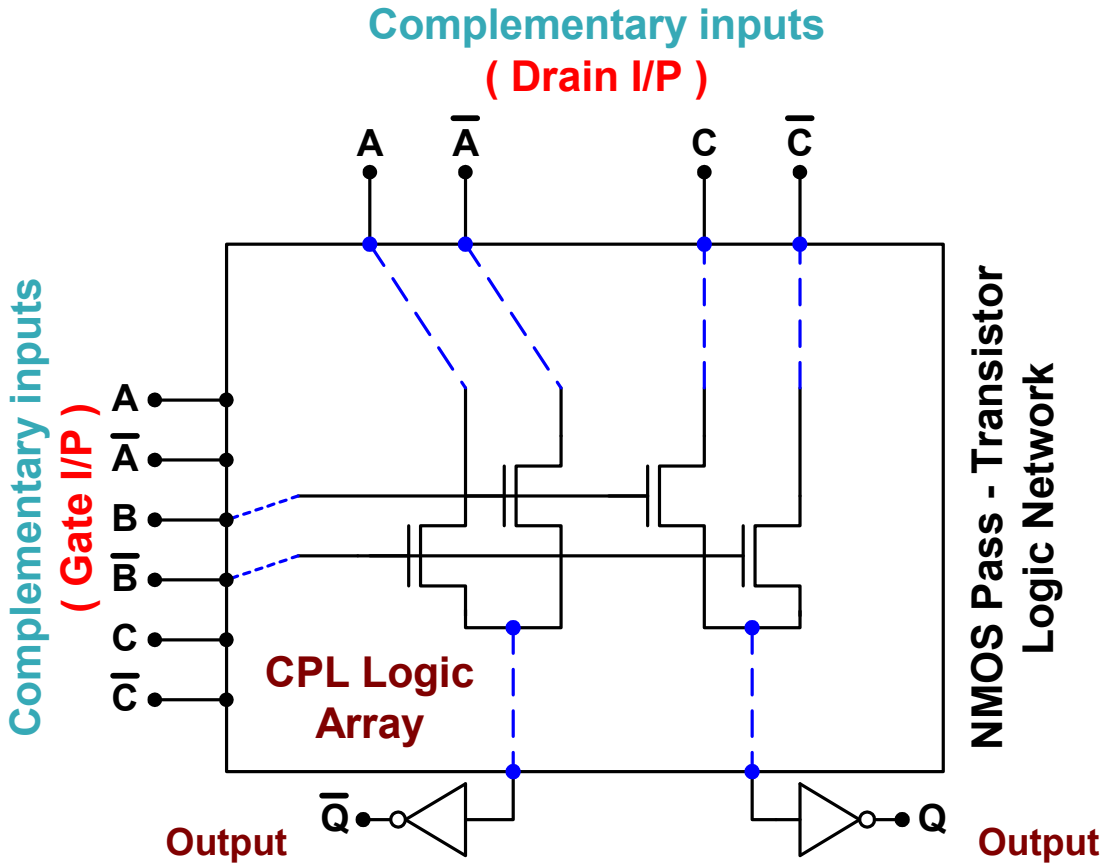
5.4.3 Pass – Transistor Array Logic (7/7)



5.4.4 Complementary Pass – Transistor Logic (CPL) (1/7)

Ref : IEEE JSSC , Vol. sc-25 , No.2 P.P.388 ~ 394 , April 1990 (Hitachi VLSI Eng. Corp.)

- CPL consists of NMOS pass transistor logic network driven by two sets of complementary inputs and two CMOS inverters used as buffers.
- For 1.2 V applications



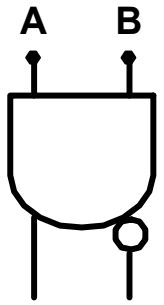
5.4.4 Complementary Pass – Transistor Logic (CPL) (2/7)

2-input NAND gate (1/2)

- The NMOS network from pull - up and pull - down function

  ➔ the voltage of the node Q is

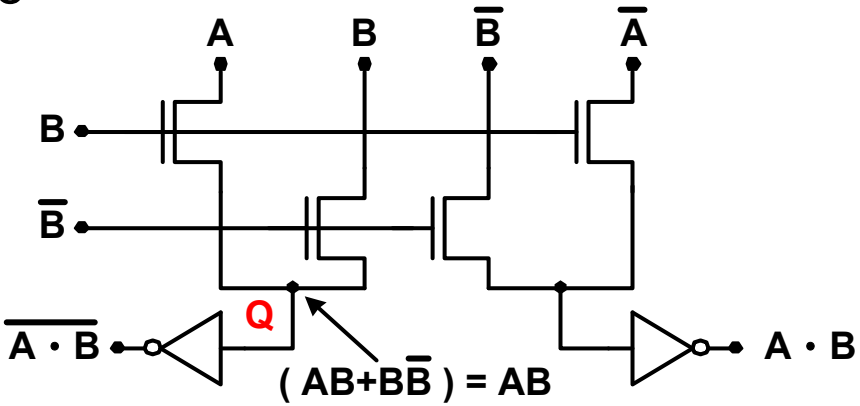
  $$V_Q = V_{DD} - V_{TN} \quad (V_Q)$$



- Sol：The logic threshold of the inverter buffer should be shifted to lower voltage than ½ VDD

  ➔ β ratio of Inv. = $\dfrac{\mu_n \dfrac{W_n}{L_n}}{\mu_p \dfrac{W_p}{L_p}} > 1$

  ➔ Inv. output quickly to pull - down



5.4.4 Complementary Pass – Transistor Logic (CPL) (3/7)

2-input NAND gate (2/2)

- ∵V_Q (High) = $V_{DD} - V_{TN}$

  ➔ the V_{GS} of the PMOS is $-V_{Tn}$ as shown

  in Fig (1)

  ➔ the PMOS is not completely OFF

  ➔DC power dissipation

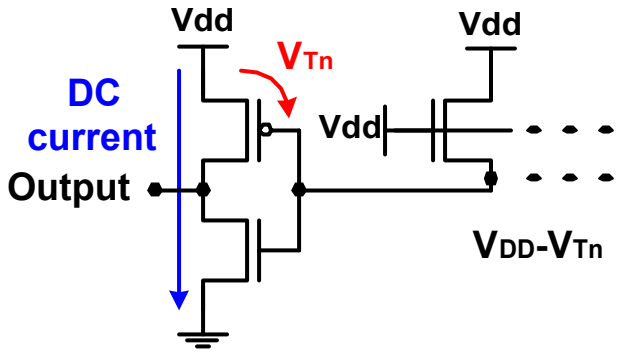


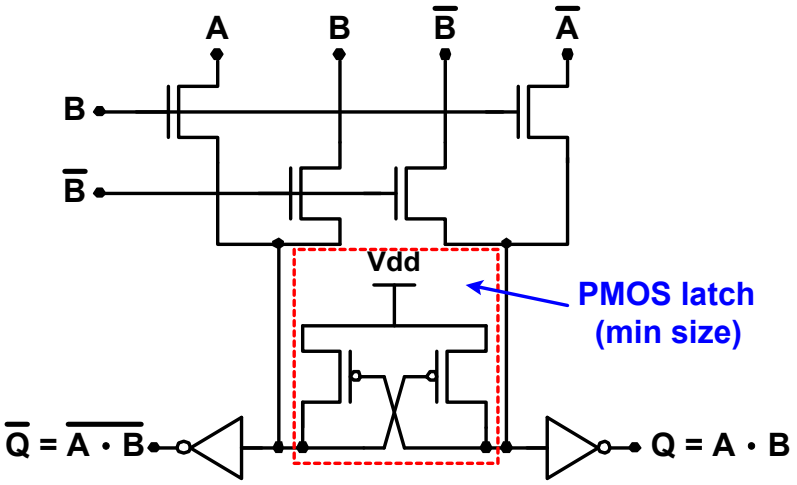
Fig (1) High level degradation in CPL .

- Sol :

(1) ※ Can design MOS threshold such that $V_{TN} < |V_{TP}|$

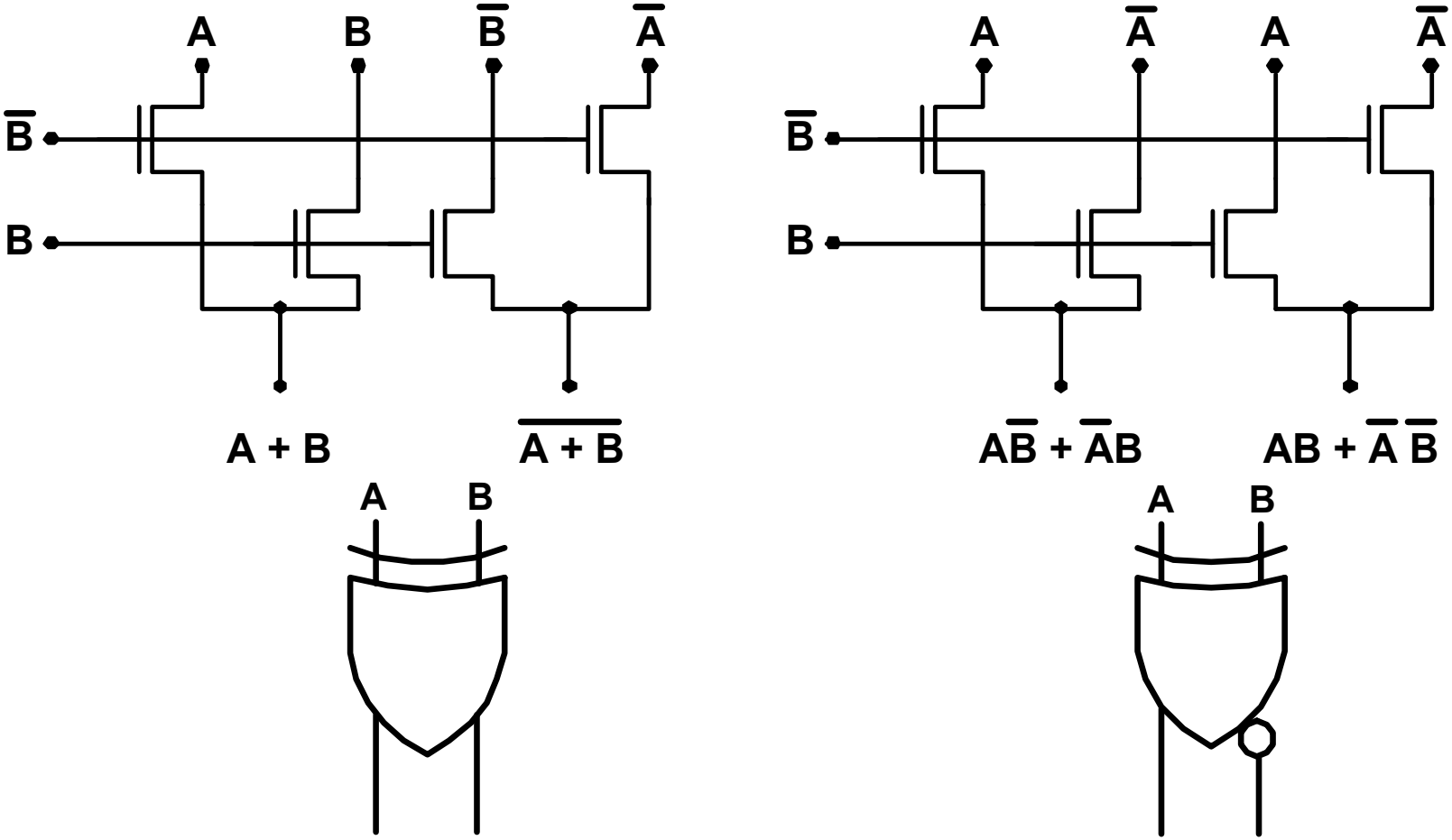
(2) ※ Using PMOS latch to restore the high – level voltage of the node Q .

※ Careful design should be considered , otherwise the high – level stored in the latch can not be discharged .



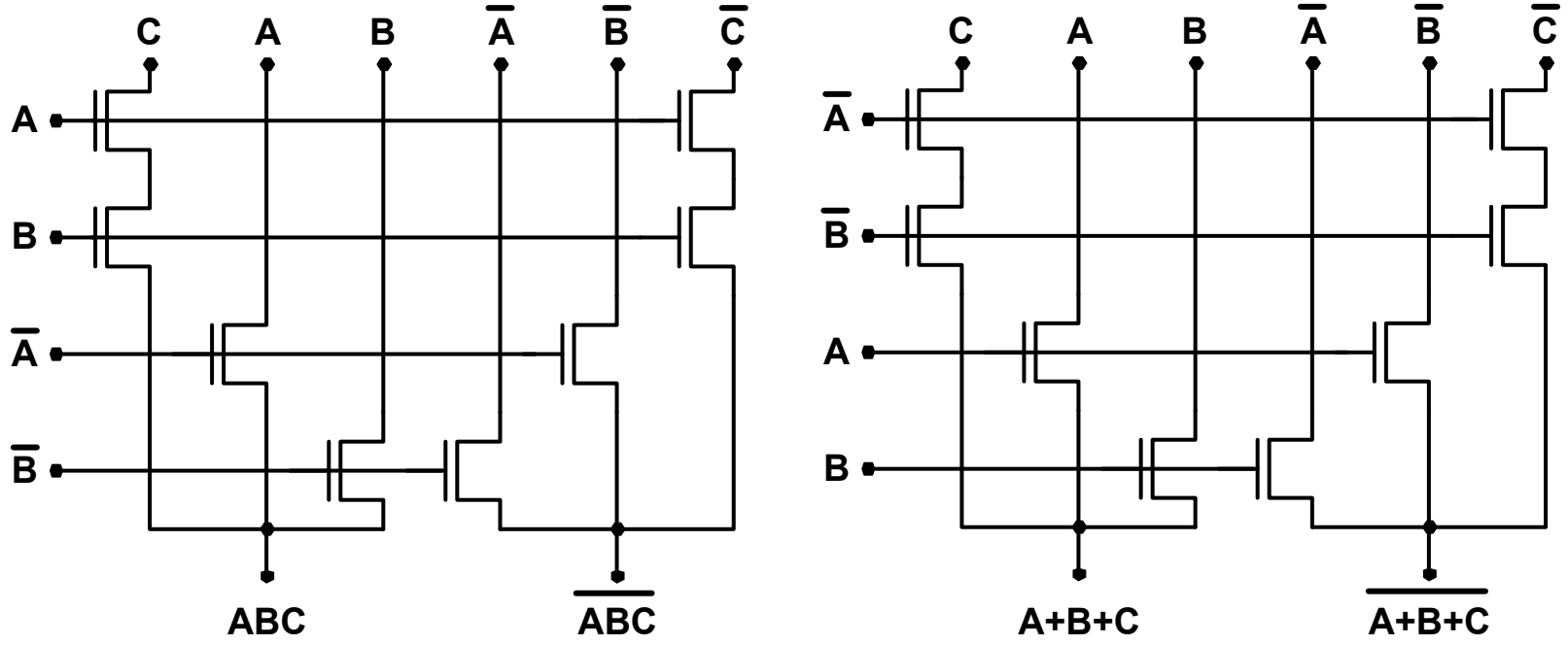
5.4.4 Complementary Pass – Transistor Logic (CPL) (4/7)

- 2-input OR , XOR gate



5.4.4 Complementary Pass – Transistor Logic (CPL) (5/7)

- 3-input AND / NAND , OR / NOR logic array

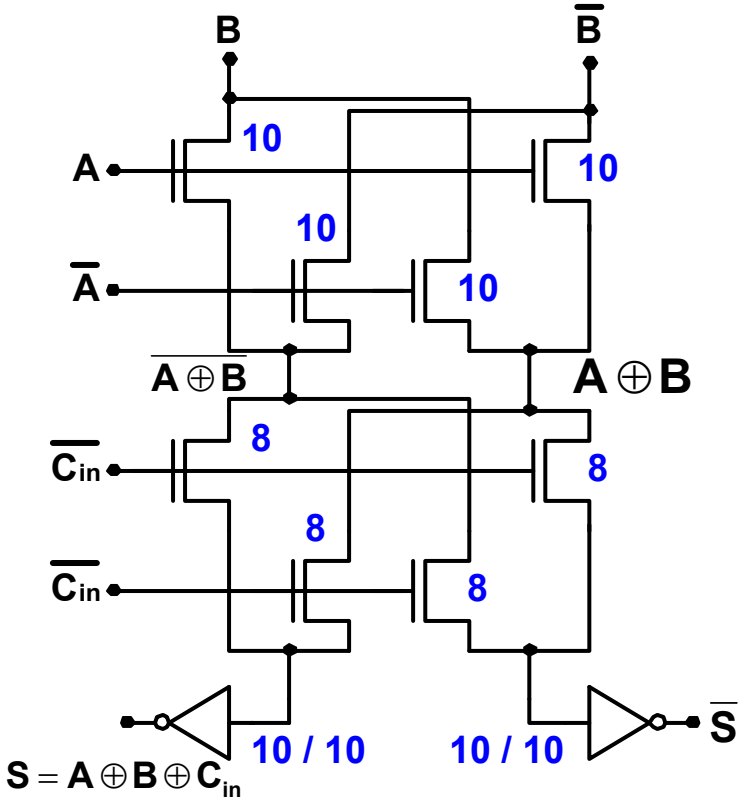


- Any complex logic function can be constructed easily using this principle of the NMOS network transistors .

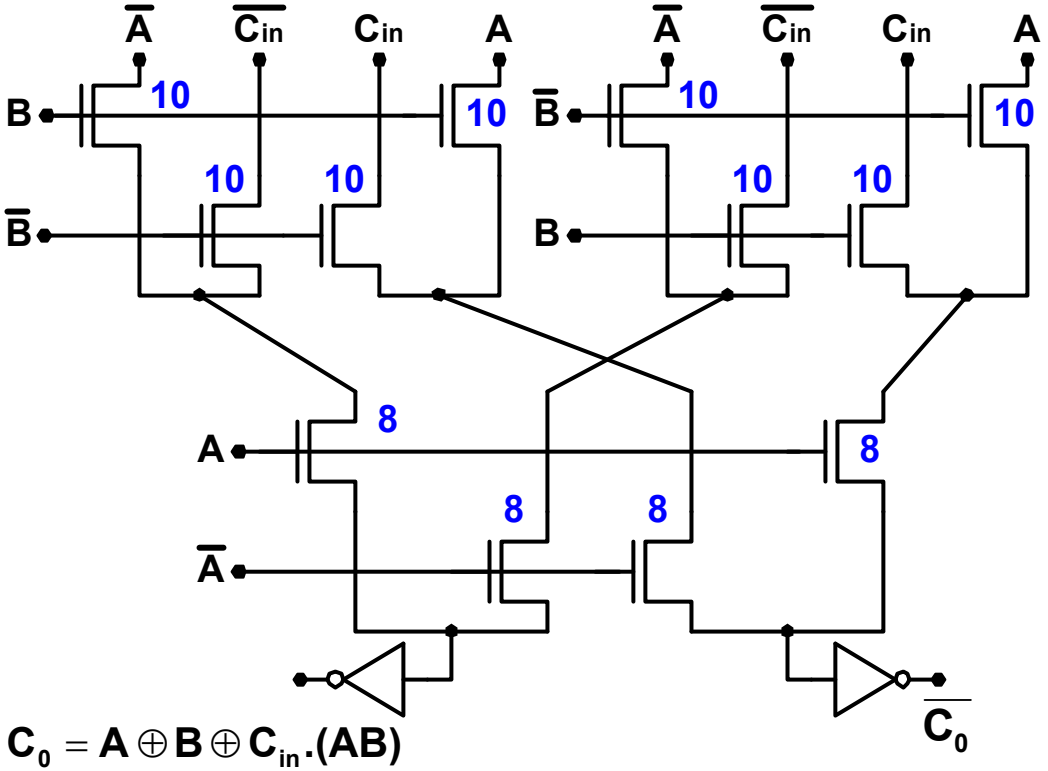
5.4.4 Complementary Pass – Transistor Logic (CPL) (6/7)

CPL Full Adder (1/2)

(a) Sum block



(b) Carry block



5.4.4 Complementary Pass – Transistor Logic (CPL) (7/7)

CPL Full Adder (2/2)

- For fast operation , the transistors of the NMOS network , far from the output , have large size than those closer to the output . (This is because the NMOS devices , closer to the output , pass a reduced swing .)

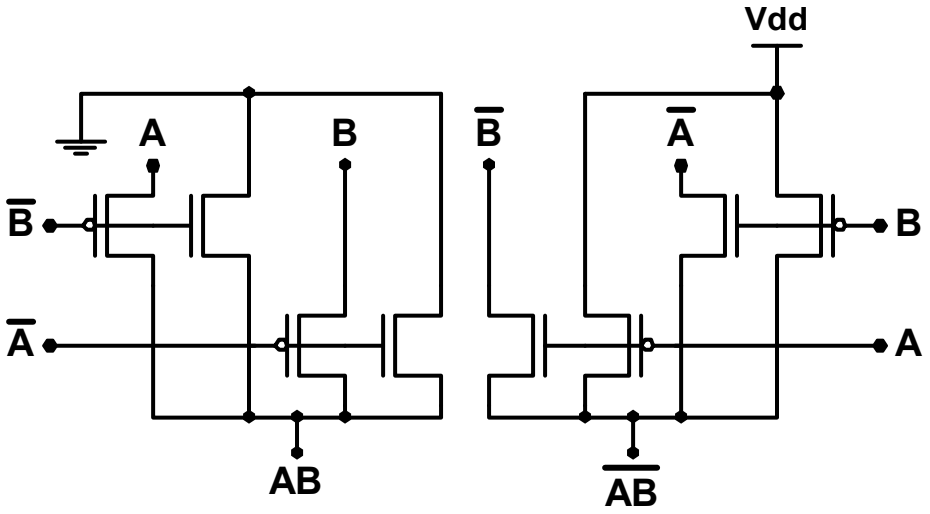
5.4.5 Dual Pass – Transistor Logic (DPL) (1/5)

- *Ref : IEEE JSSC , vol. sc-28 , no.11 , PP.1145~1151 , Nov. 1993.*

The DPL is a modified version of CPL suitable for low – voltage applications.

DPL NAND / AND gate

- Consists of NMOS and PMOS pass transistors .
- Output is full rail – to – rail swing



DPL AND / NAND gate

5.4.5 Dual Pass – Transistor Logic (DPL) (2/5)

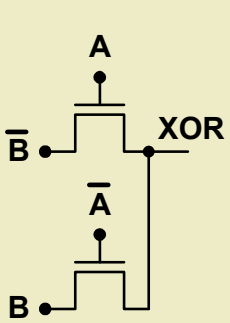
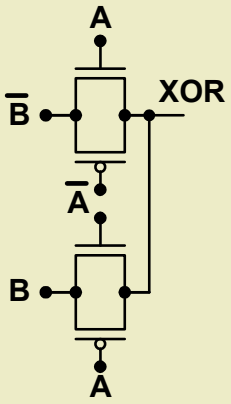
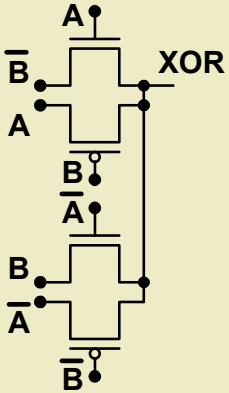
Features of DPL

- The DPL has a balanced input capacitance . (reduces the delay dependence of the input data)
- In DPL , for any input combination , these are always two current paths driving the output .

5.4.5 Dual Pass – Transistor Logic (DPL) (3/5)

Comparison of CPL , conventional CMOS TG and DPL

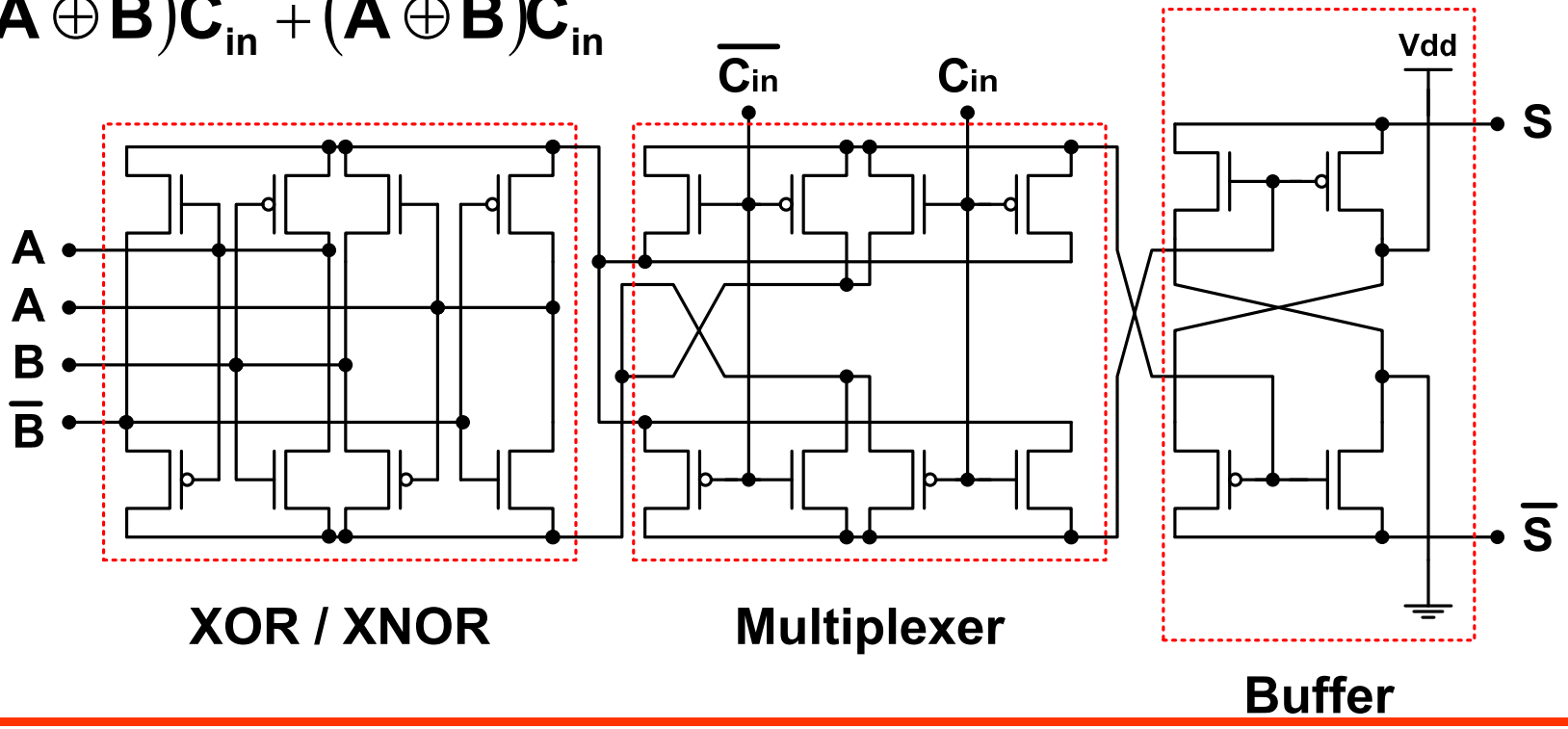
- In the true tables , the column labeled " pass " shows which signals are passed and perform the XOR function .

	CPL	CMOS	DPL
circuit			
Turth Table & Operation	A B XOR Pass 0 0 0 B 0 1 1 1 0 1 $\bar{B}$ 1 1 0	A B XOR Pass 0 0 0 B 0 1 1 1 0 1 $\bar{B}$ 1 1 0	A B XOR Pass 0 0 0 B A 0 1 1 $\bar{A}$ 1 0 1 $\bar{B}$ A 1 1 0 $\bar{A}$
Swing	$0 \leftrightarrow V_{DD}- V_{Tn}$	$0 \leftrightarrow V_{DD}$	$0 \leftrightarrow V_{DD}$

5.4.5 Dual Pass – Transistor Logic (DPL) (4/5)

- CPL Full – Adder (1/2)

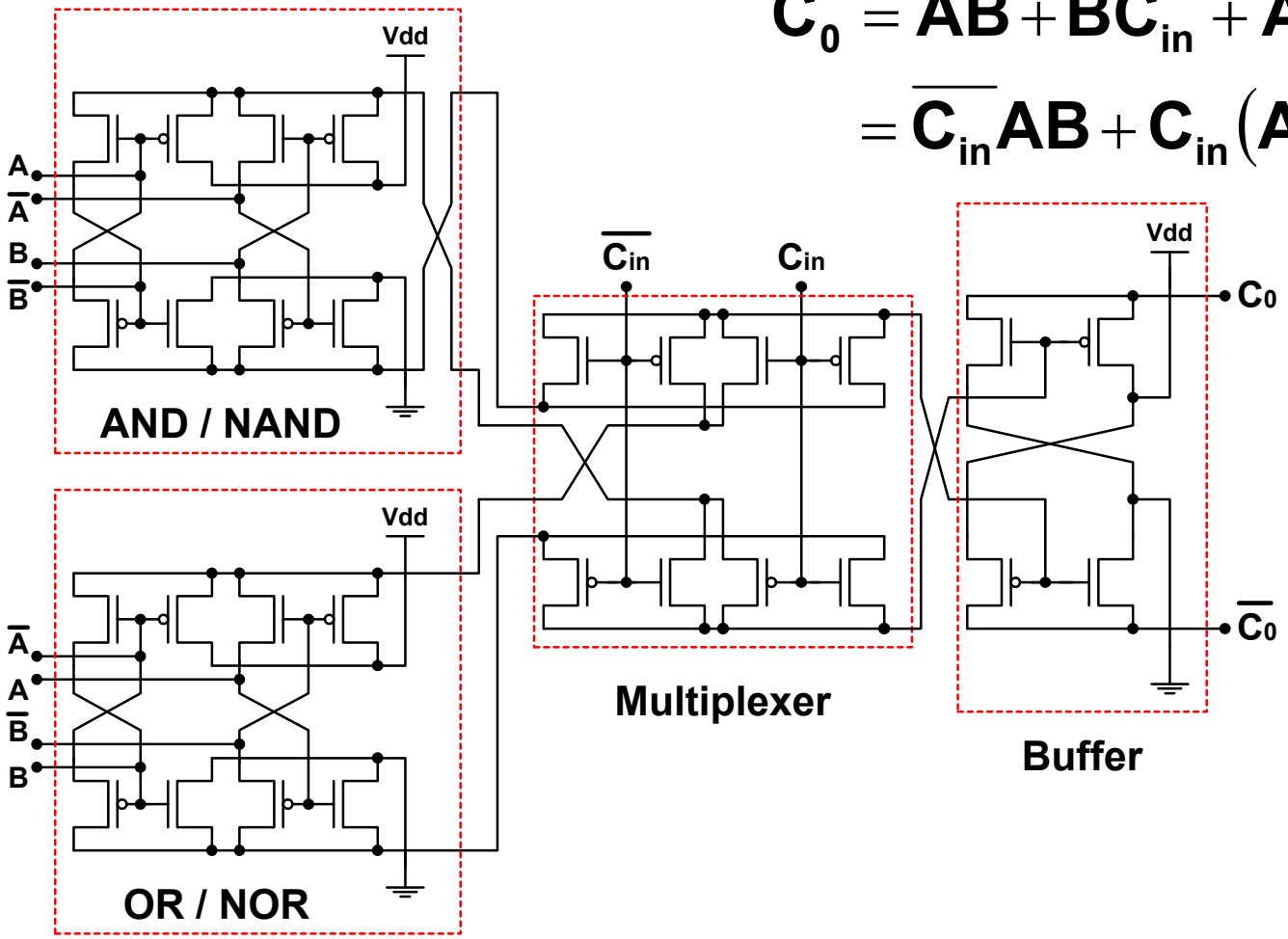
$$S = A \oplus B \oplus C_{in}$$
$$= (A \oplus B) \oplus C_{in}$$
$$= (A \oplus B)\overline{C_{in}} + \overline{(A \oplus B)}C_{in}$$



5.4.5 Dual Pass – Transistor Logic (DPL) (5/5)

CPL Full – Adder (2/2)

$$C_0 = AB + BC_{in} + AC_{in}$$
$$= \overline{C_{in}}AB + C_{in}(A + B)$$

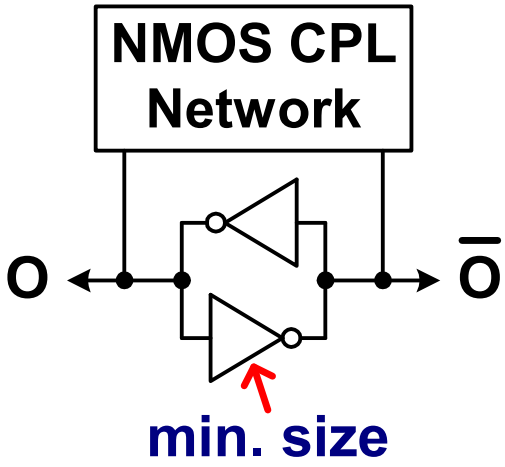


5.4.6 Swing Restored Pass - Transistor Logic (SRPL) (1/2)

- *Ref : IEEE Custom Integrated Circuits Conference Tech Dig. PP.278~281 , May. 1994.*

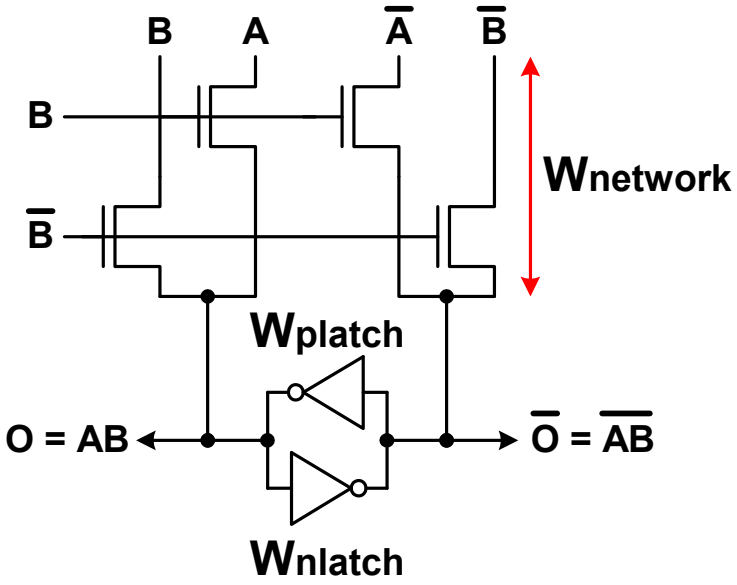
CPL like style suitable for low – power / low – voltage application .

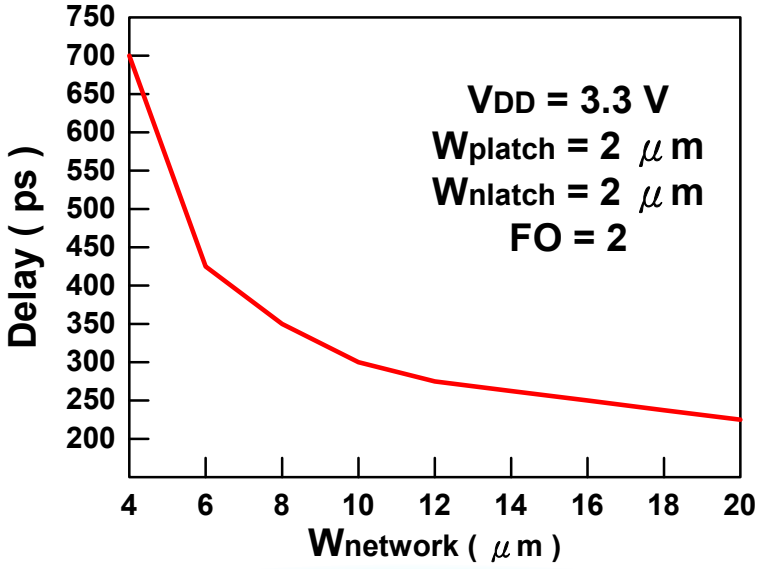
- The cross – coupled CMOS inverters (latch) permit to restore the logic levels .
- The size of the latch should be minimum



5.4.6 Swing Restored Pass - Transistor Logic (SRPL) (2/2)

- The size of such a logic is critical for speed and power dissipation .





Delay of AND / NAND gate versus NMOS size

- *If the size of the NMOS transistors in the network is small , the output of the SRPL gate fail to swing to ground , because the equivalent impedance of the network is lower than the one seen by the output to VDD .*

5.4.7 Pass - Transistor Logic Comparison

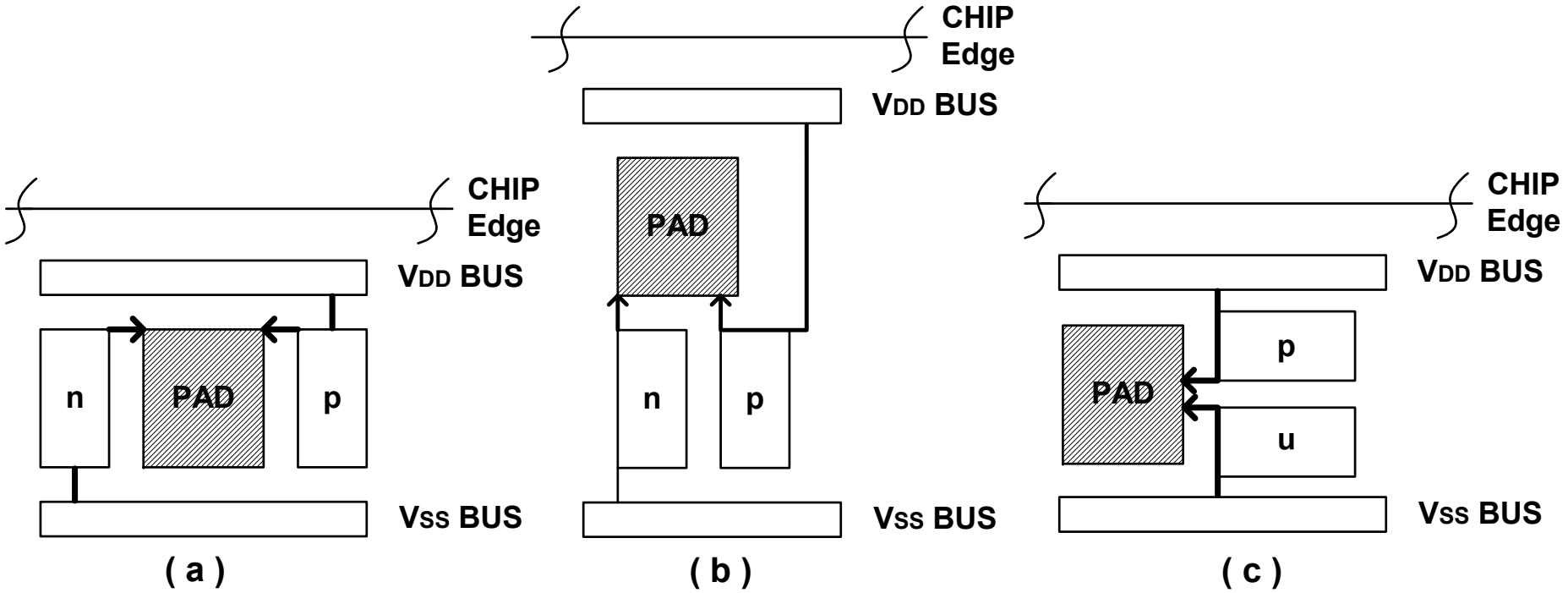
- The speed and power dissipation of the different pass – logic styles , so far presented , depend on the circuit type and the application of the circuit (cascaded gates , driving a fixed load , etc .) . For the case of a full – adder , used in a multiplier array , a comparison is given . In general , SRPL has the lowest power dissipation but careful design is needed when small device sizes are used . The DPL consumes more power than SRPL and PMOS latch CPL , because of the higher transistor count . Both CPL and SRPL circuits have the smallest area and the fastest speed . In summary , CPL – like styles are promising , for low – power and high – speed applications .

5.5 Input / Output Circuits

- I/O circuits connect the on – chip logic circuitry to the external world .
- Play an important role in the limitation of speed and power dissipation of the whole chip

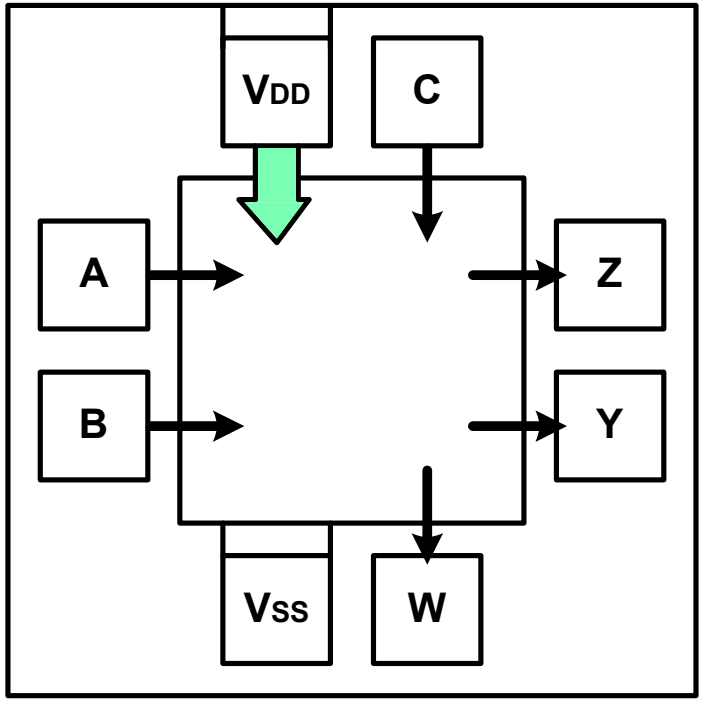
5.5.1 Overall Organization (1/2)

- Pad structure

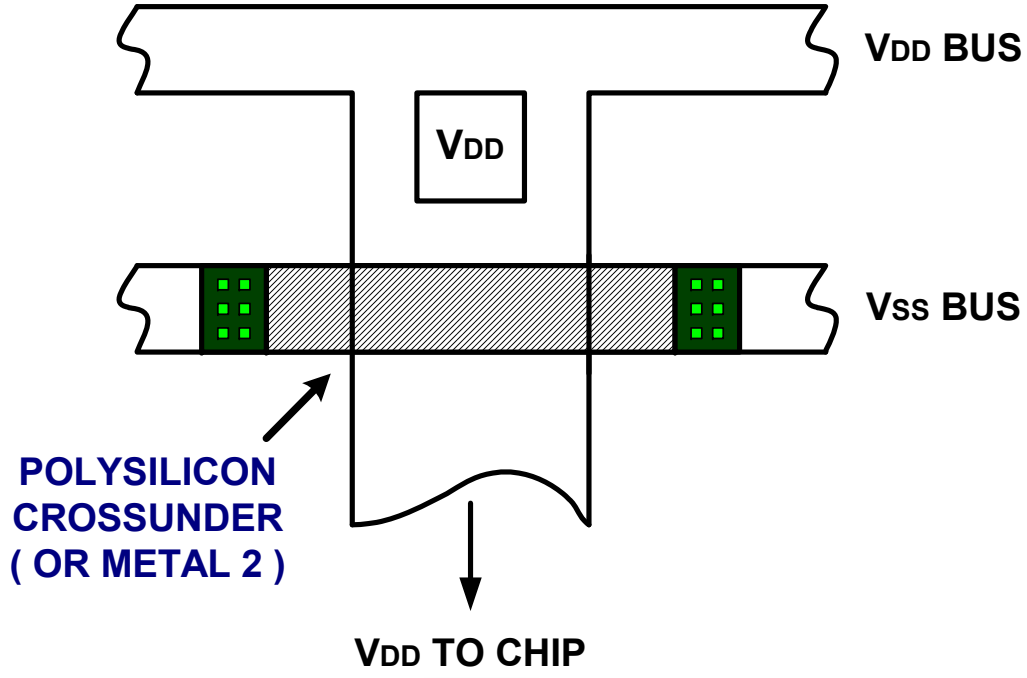


5.5.1 Overall Organization (2/2)

- I/O Frame

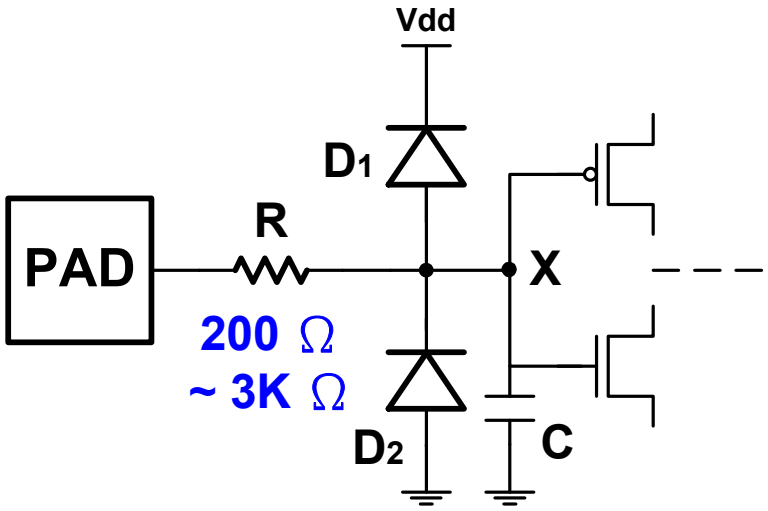


(a) I/O frame generation



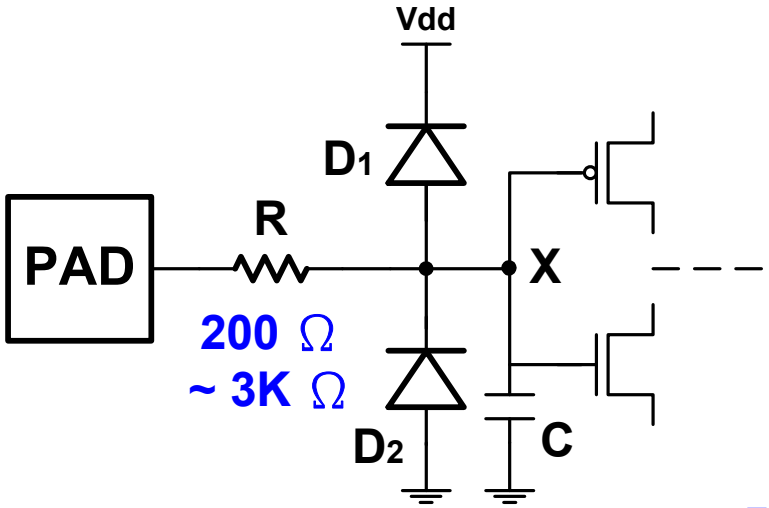
(b) VDD pad design

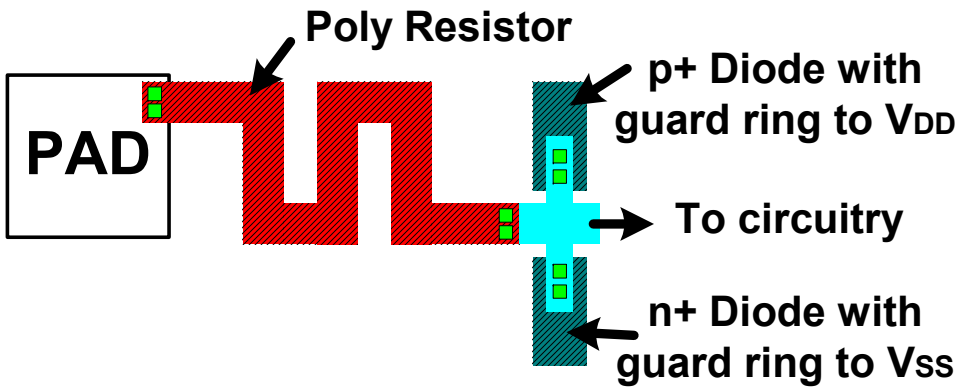
5.5.2 Input – pad Protection Circuit (1/2)

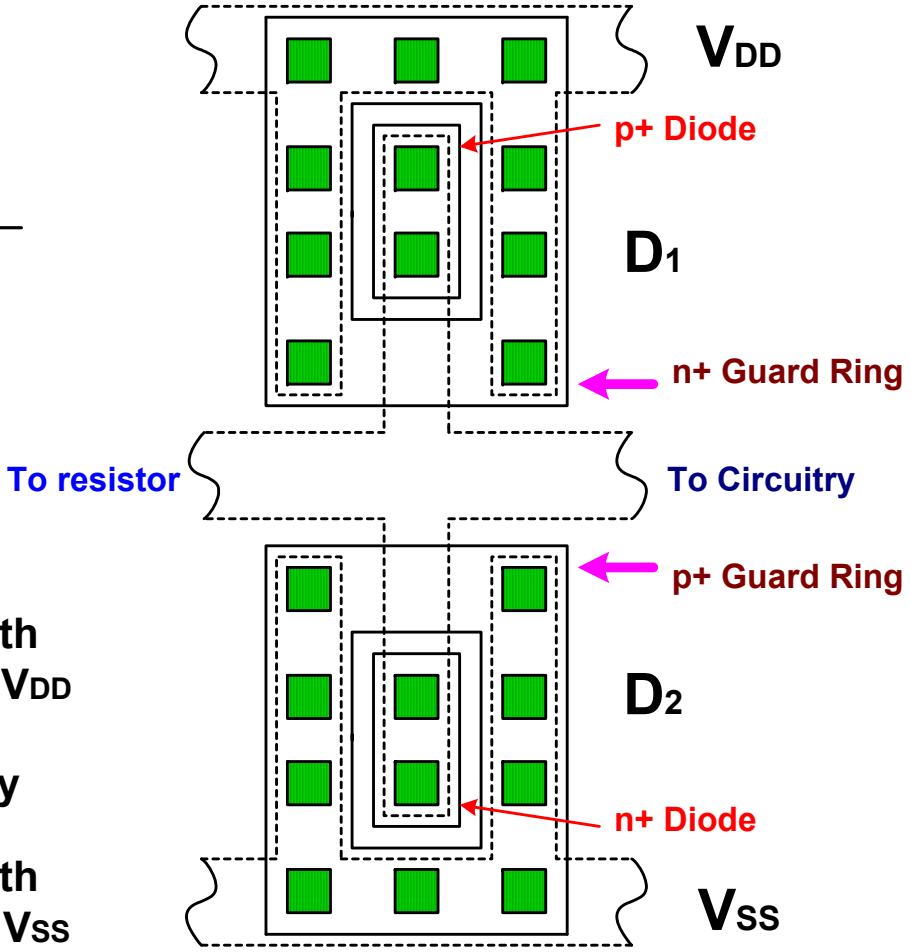


- D1 and D2 protection diode：limit the voltage excursion of the node N within ($-V_{BE}$) and ($V_{DD}+V_{BE}$)
- Input protection circuit has a parasitic RC limits the high speed operation .

5.5.2 Input – pad Protection Circuit (2/2)







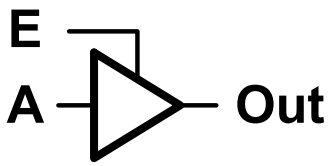
5.5.3 Tri – state Output Buffer (1/2)

- Non – inverting tri – state buffer

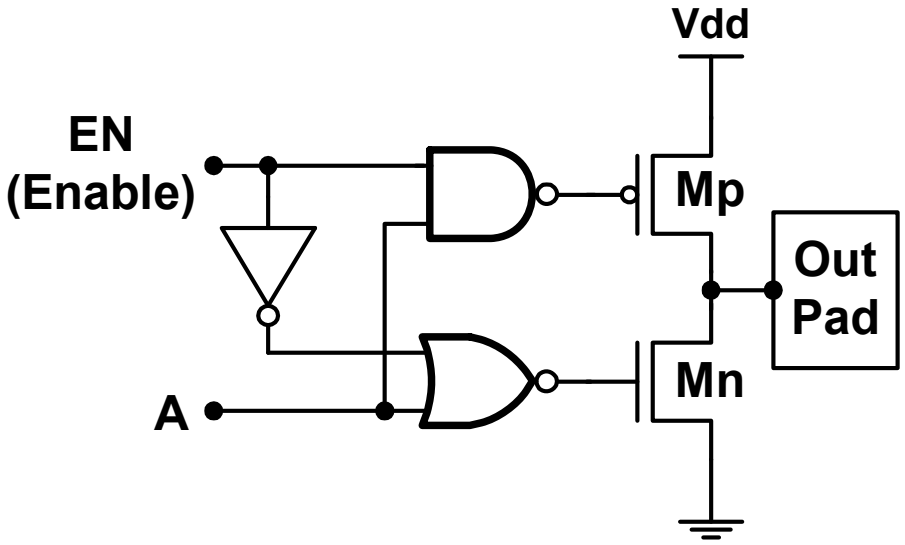
※ For large output cap.

Truth table

En	A	Output	
0	X	Z	(High Impedance)
1	0	0	
1	1	1	



(a) symbol

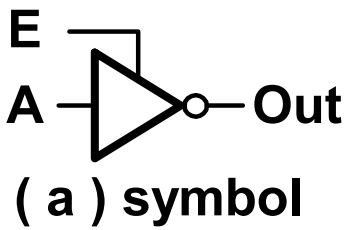


(b) Circuit

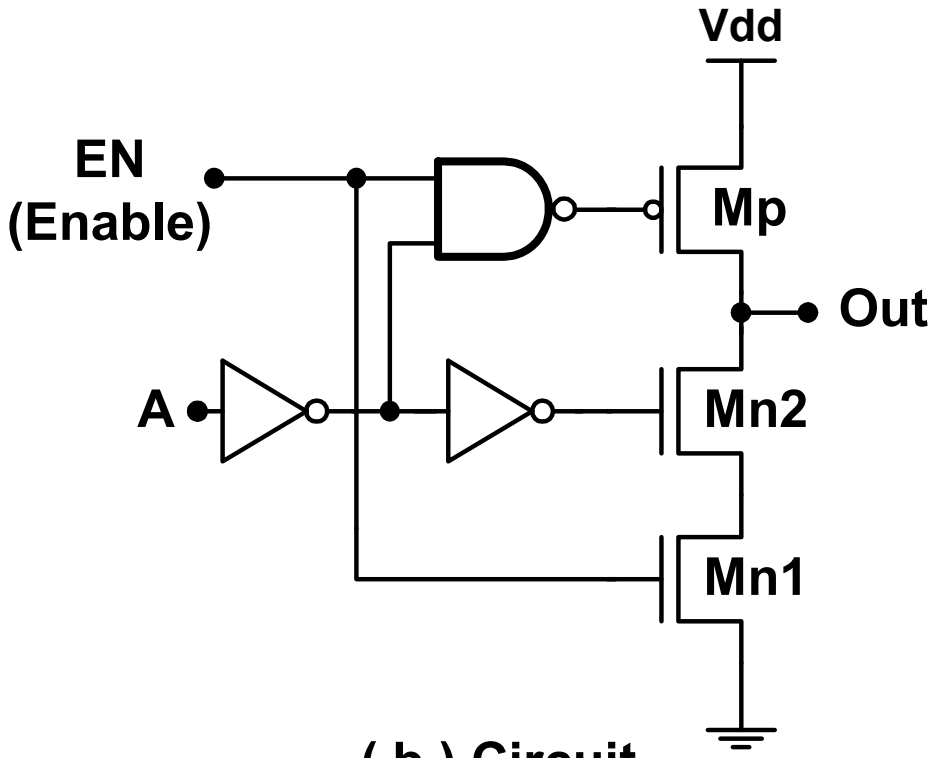
※ C^2MOS is not suitable in this case (Why ?)

5.5.3 Tri – state Output Buffer (2/2)

- Inverting tri – state buffer

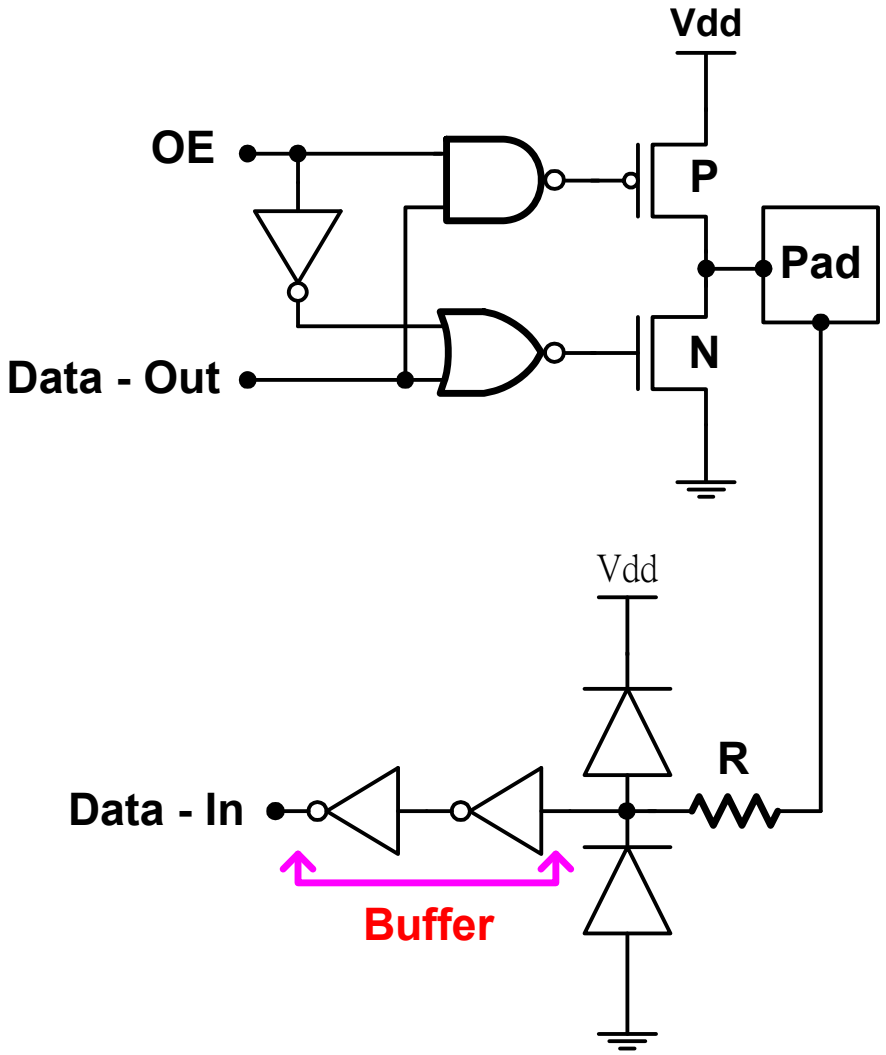


(a) symbol



(b) Circuit

5.5.4 Bi – Directional Pad

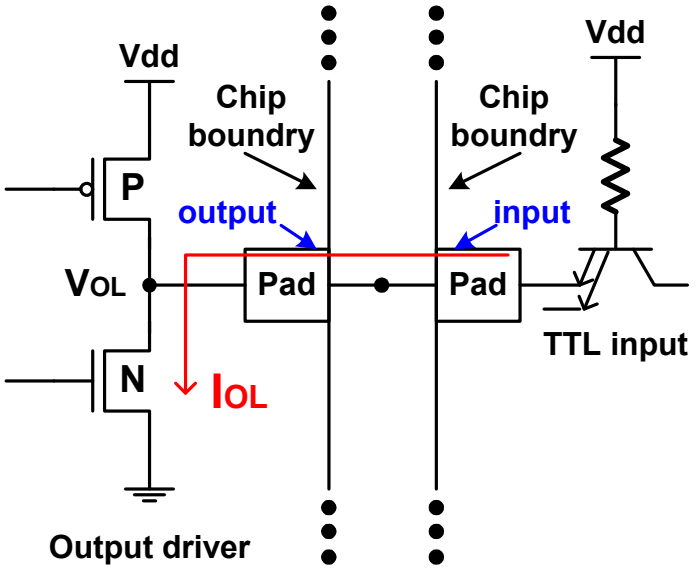


5.5.5 CMOS To TTL Output Buffer

- If the CMOS Output buffer is intended to drive bipolar TTL inputs , then important current is SUNK .

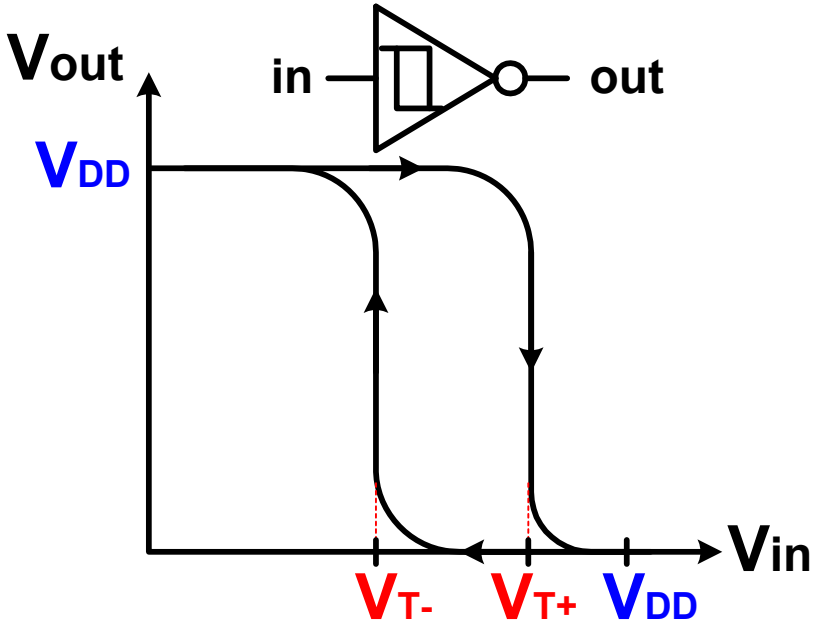
➔ A CMOS output buffer must sink this current .

- For 3.3V supply , this current can be 1mA to 12mA (depending on driver)

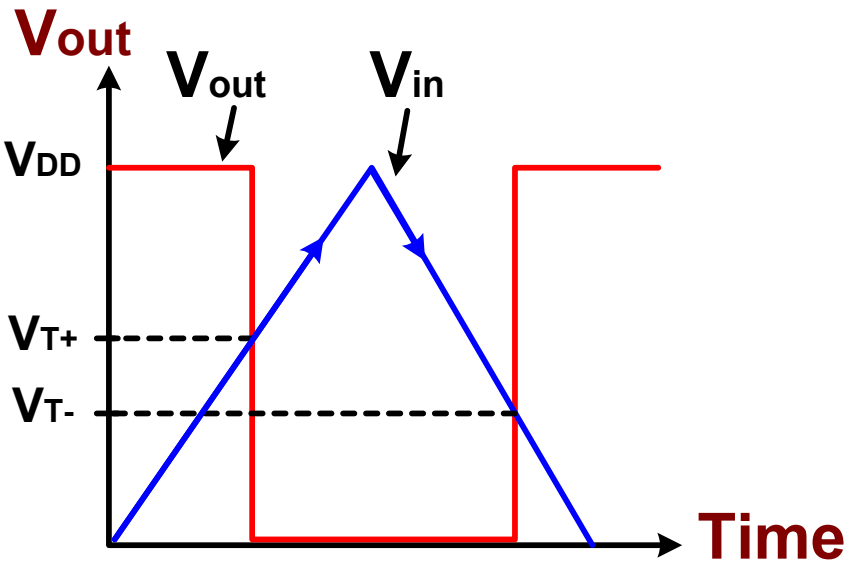


TTL output buffer

5.5.6 Schmitt Trigger (1/4)



Transfer characteristic of Schmitt trigger



Voltage waveforms for slow input

5.5.6 Schmitt Trigger (2/4)

- When $V_{in} = V_{T+}$, N_2 enters in conduction mode which means $V_{GS2} = V_{Tn}$ then

$$V_{FN} = V_{T+} - V_{Tn}$$

- We neglect the body effect of N_2
- The voltage V_{FN} is controlled by N_1 and N_3 . These transistors operate in saturation because

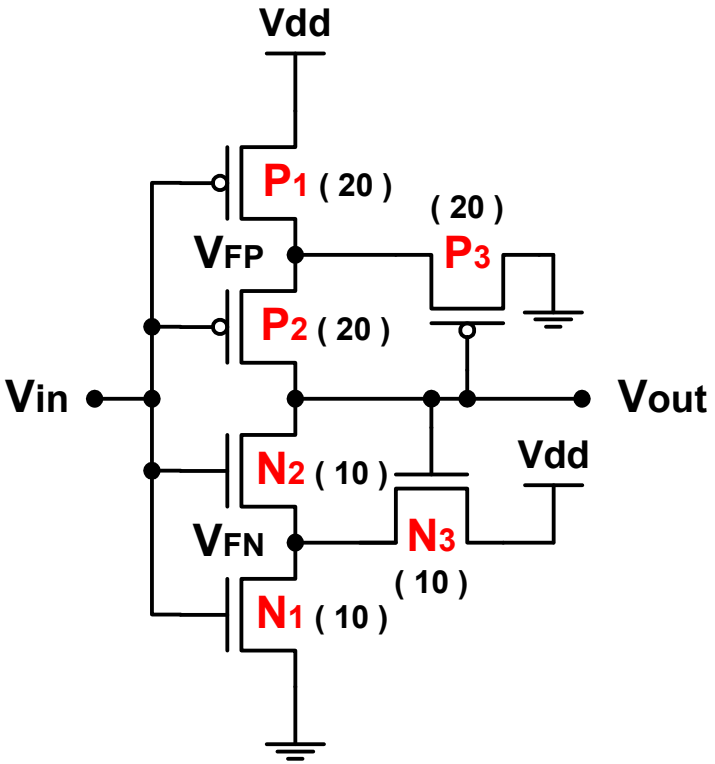
$$V_{GS1} = V_{T+}$$

$$V_{DS1} = V_{FN} = V_{FN} = V_{T+} - V_{Tn}$$

And

$$V_{GS3} = V_{DD} - V_{FN}$$

$$V_{DS3} = V_{DD} - V_{FN}$$



The CMOS Schmitt trigger schematic

5.5.6 Schmitt Trigger (3/4)

The drain currents flowing in N_1 and N_3 are equal . Then using a simple MOS model we have

$$\frac{\beta_{n1}}{2}(V_{T+}-V_{Tn})^2=\frac{\beta_{n3}}{2}(V_{DD}-V_{T+})^2$$

We have

$$V_{T+}=\frac{V_{DD}+\sqrt{\beta_n}V_{Tn}}{1+\sqrt{\beta_n}}$$

Where

$$\beta_n=\frac{\beta_{n1}}{\beta_{n3}}=\left\{\frac{W_{n1}/L_{n1}}{W_{n3}/L_{n3}}\right\}_{eff}$$

This equation shows that the trigger point is independent of the process parameters except for V_{Tn} . By symmetry , the trigger point for falling transition , can be deduced from the pull – up section . We have

$$V_{T-}=\frac{\sqrt{\beta_p}(V_{DD}+V_{Tp})}{1+\sqrt{\beta_p}}$$

Where

$$\beta_p=\frac{\beta_{p1}}{\beta_{p3}}=\left\{\frac{W_{p1}/L_{p1}}{W_{p3}/L_{p3}}\right\}_{eff}$$

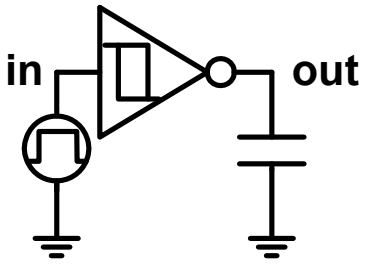
5.5.6 Schmitt Trigger (4/4)

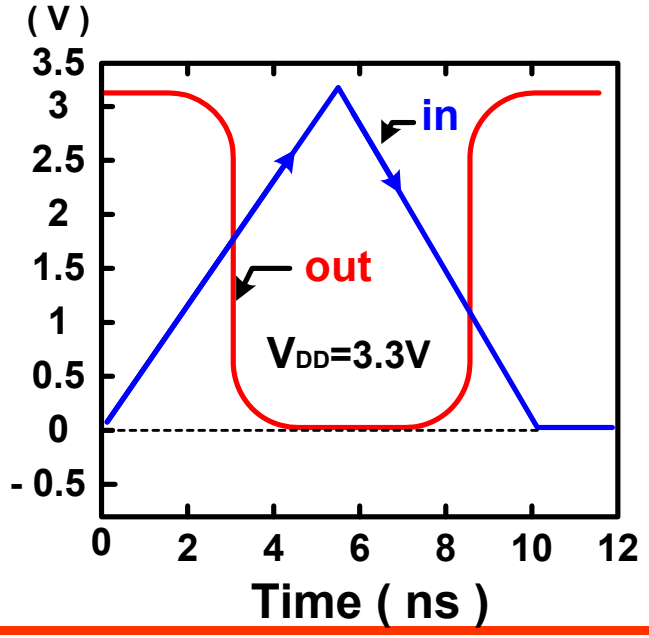
If n=p and $V_{Tn} = -V_{Tp} = V_T$, then

$$V_{T+} = \frac{V_{DD}}{2} + \frac{V_T}{2} \; ; \; V_{T-} = \frac{V_{DD}}{2} - \frac{V_T}{2} \; ; \; V_H = V_{T+} - V_{T-} = V_T$$

In this case the hysteresis voltage can be made equal to V_T .

The short – circuit power dissipation of the Schmitt trigger can be very important since the rise / fall times of the input signal is very long .

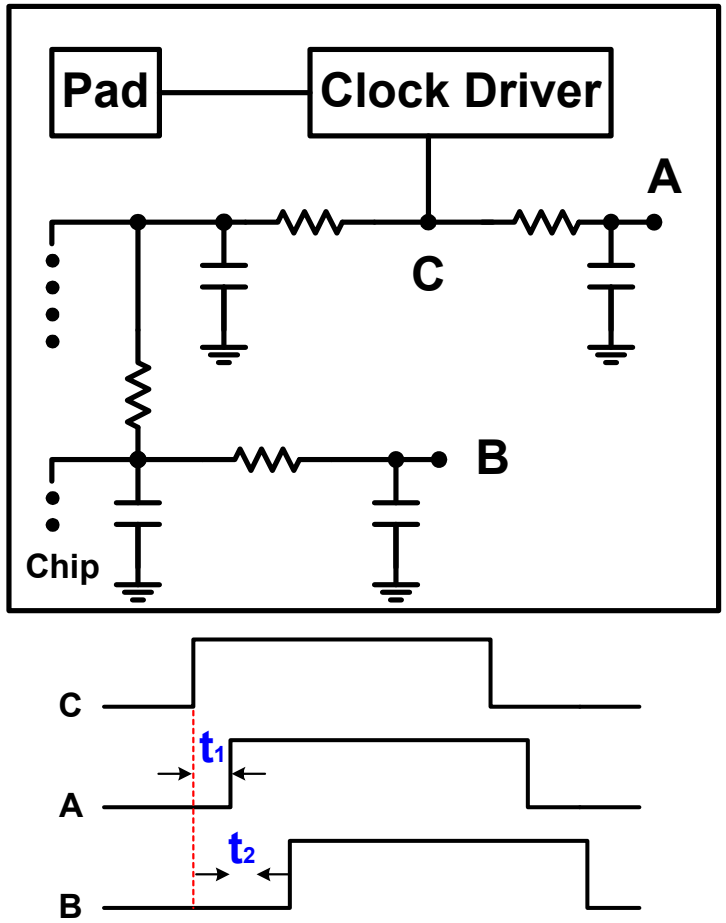




5.6 Clock Drivers and Clock Distribution

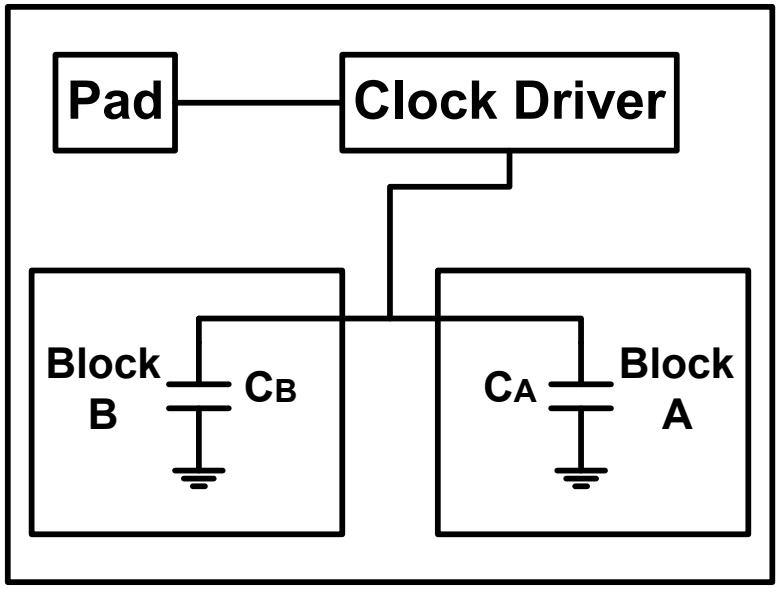
5.6.1 Clock Skew Problem (1/2)

- The clock skew problem is due mainly to two issues：
- **(1)、The difference in RC interconnect time constants：**
- Example：node A and node B have different branch lengths to node C ➔ at node A and node B are different

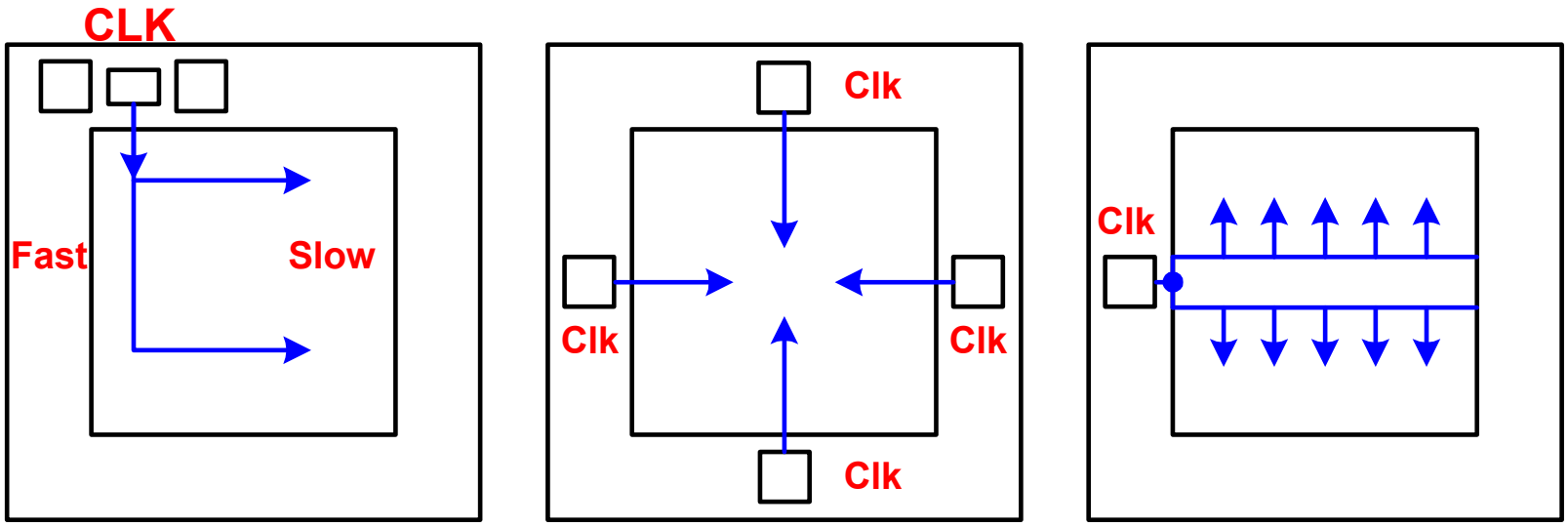


5.6.1 Clock Skew Problem (2/2)

- **(2)、Unbalanced loads at different nodes：**
  - Example：load at node A and B (C_A , C_B) are different ➔ clock skew .



5.6.2 Clock Driver Options (1/3)



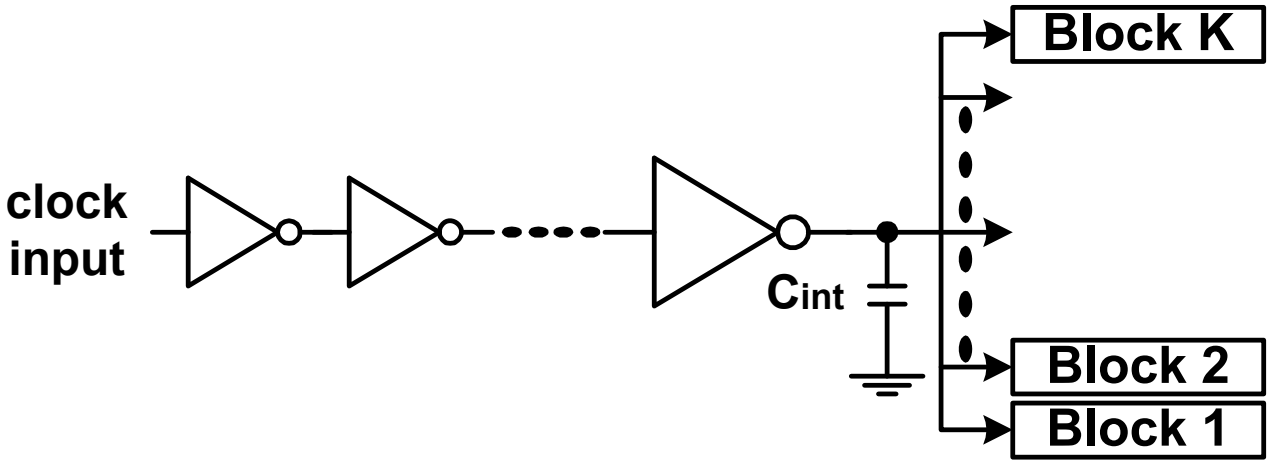
(a) Single driver

(b) Four side approach

(c) " down the center " approach

5.6.2 Clock Driver Options (2/3)

- Clock distribution
  - Cascaded inverters for global clock distribution



5.6.2 Clock Driver Options (3/3)

- Tree distribution