VLSI Digital Design

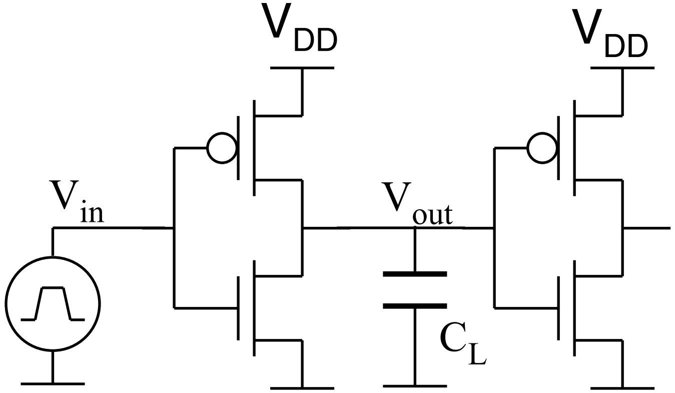
MODULE II
CMOS LOGIC DESIGN

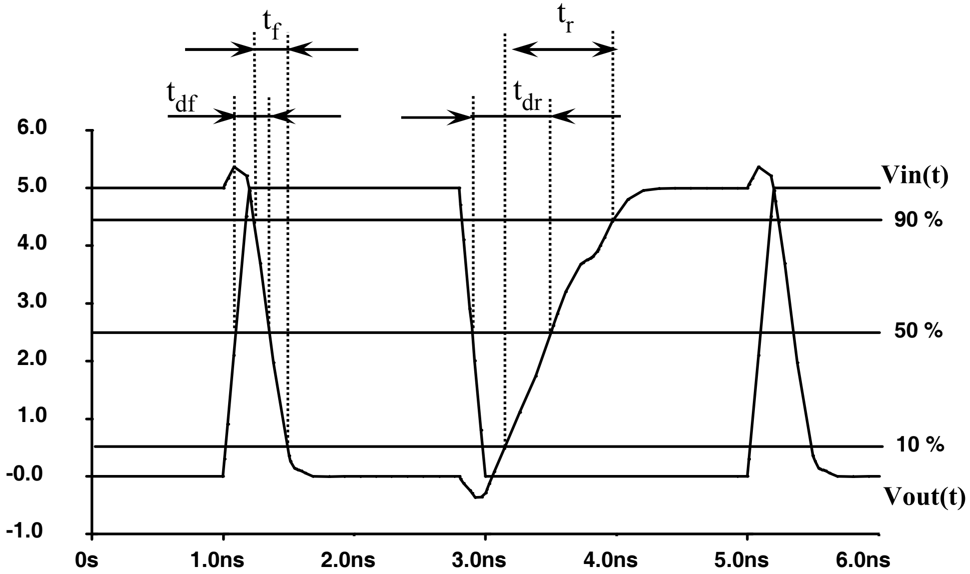
2.1. Switching characteristics review
2.2. Delay, logical effort and buffering
2.3. Logic structures
2.4. Clock strategies

2.1 Switching characteristics review

Time definitions

- *Rising time* $\mathbf{t_r}$: Transition time between 10% to 90% of the steady-state levels.
- *Falling time* $\mathbf{t_f}$: Transition time between 90% to 10% of the steady-state levels.
- *Delay* $\mathbf{t_d}$: Time between crossing the 50% of input signal transition and output signal transition.

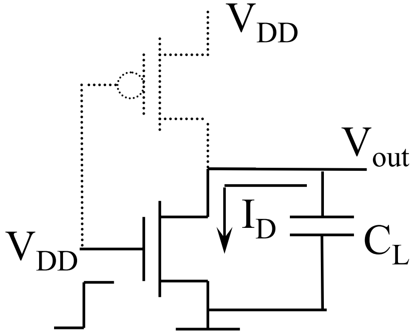


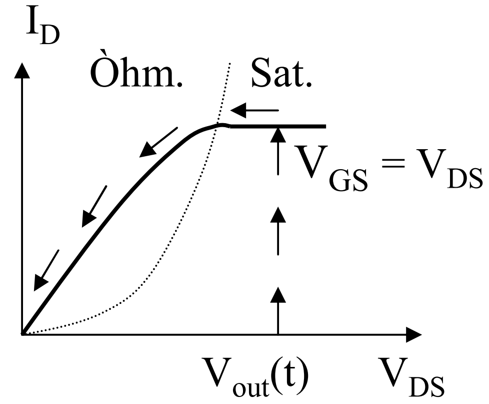


2.1 Switching characteristics review

First-order analytic delay model

Hypothesis: fast ramp (PMOS does not operate)





$$I_{Dn} = \frac{\beta_n}{2}(V_{GS} - V_{TN})^2 \quad (SATURATION : V_{DS} \geq V_{GS} - V_{TN})$$

$$I_{Dn} = \beta_n\left[(V_{GS} - V_{TN})V_{DS} - \frac{V_{DS}^2}{2}\right] \quad (OHMIC : V_{DS} \leq V_{GS} - V_{TN})$$

Falling time:

$$C_L \frac{dV_{out}}{dt} + i_D = 0; \quad t_f = \int_{0.1V_{DD}}^{0.9V_{DD}} \frac{C_L}{i_D} dV_{out}$$

- Saturation region

$$t_{f1} = \frac{2C_L}{\beta_n(V_{DD} - V_{TN})^2} \int_{V_{DD}-V_{TN}}^{0.9V_{DD}} dV_{out} = \frac{2C_L(V_{TN} - 0.1V_{DD})}{\beta_n(V_{DD} - V_{TN})^2}$$

- Ohmic region

$$t_{f2} = \frac{C_L}{\beta_n(V_{DD} - V_{TN})} \int_{0.1V_{DD}}^{V_{DD}-V_{TN}} \frac{dV_{out}}{V_{out} - \frac{V_{out}^2}{2(V_{DD} - V_{TN})}}$$

$$t_{f2} = \frac{C_L}{\beta_n(V_{DD} - V_{TN})} \ln\left(\frac{19V_{DD} - 20V_{TN}}{V_{DD}}\right)$$

2.1 Switching characteristics review

Equating $\mathrm{n} = \frac{\mathrm{V_{TN}}}{\mathrm{V_{DD}}}$ and adding $\mathrm{t_{f1}}$ and $\mathrm{t_{f2}}$,

$$t_f = t_{f1} + t_{f2} = \frac{2C_L}{\beta_n V_{DD}}\left[\left(\frac{n-0.1}{(1-n)^2}\right) + \frac{1}{2(1-n)}\ln(19-20n)\right]$$

we obtain:

$$\boxed{t_f = k_n \frac{C_L}{\beta_n V_{DD}}}$$

where $k_n = f(n) = \frac{1}{(1-n)}\left[\frac{2(n-0.1)}{1-n} + \ln(19-20n)\right]$

Typically takes values between 3 and 5.

Rising time (dual analysis):

$$\boxed{t_r = k_p \frac{C_L}{\beta_p V_{DD}}}$$

Delay: $\mathrm{t_d} \equiv \frac{\mathrm{t_{dr}} + \mathrm{t_{df}}}{2} \cong \frac{\frac{\mathrm{t_r}}{2} + \frac{\mathrm{t_f}}{2}}{2} = \frac{\mathrm{t_r} + \mathrm{t_f}}{4}$

for $\beta_p = \beta_n$ i $\mathrm{V_{TN}} = |\mathrm{V_{TP}}|$,

$$\boxed{\mathrm{t_d} = \mathrm{k}\frac{\mathrm{C_L}}{\beta \mathrm{V_{DD}}}}$$

Assuming a single inverter load identical to the driver:

$\mathrm{C_L} = 2\mathrm{C'_{OX}} \cdot \mathrm{W} \cdot \mathrm{L}$

Taking into account that $\beta = \mu C'_{OX}\frac{W}{L}$

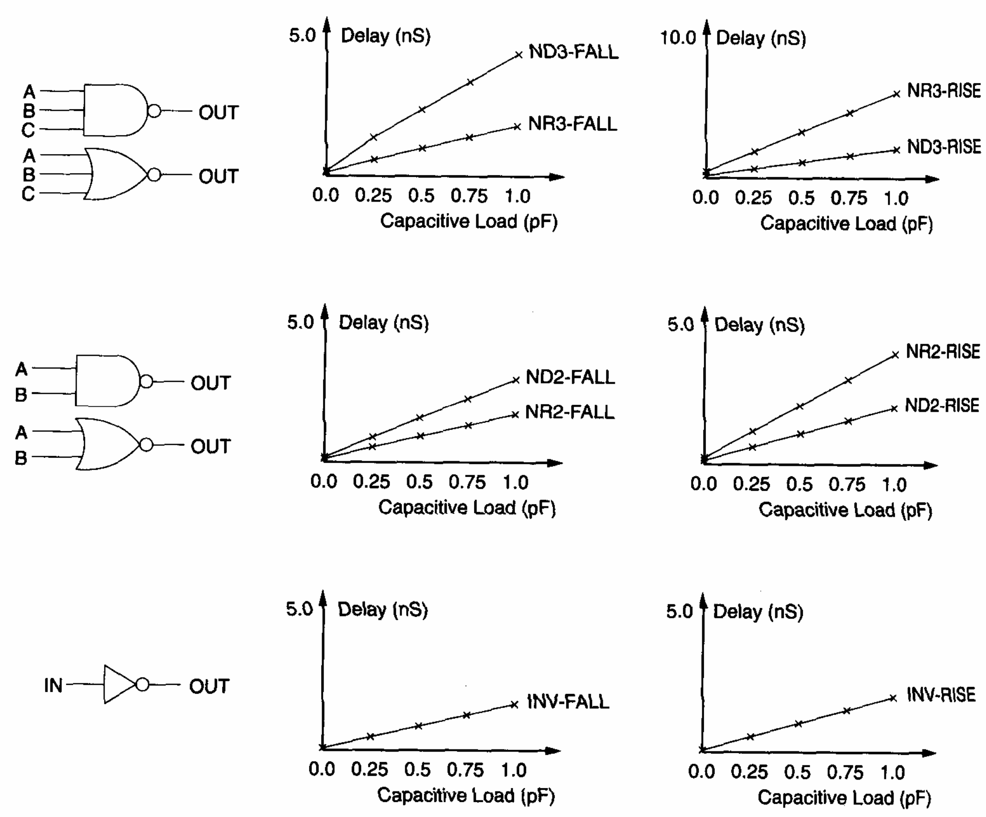
It results:

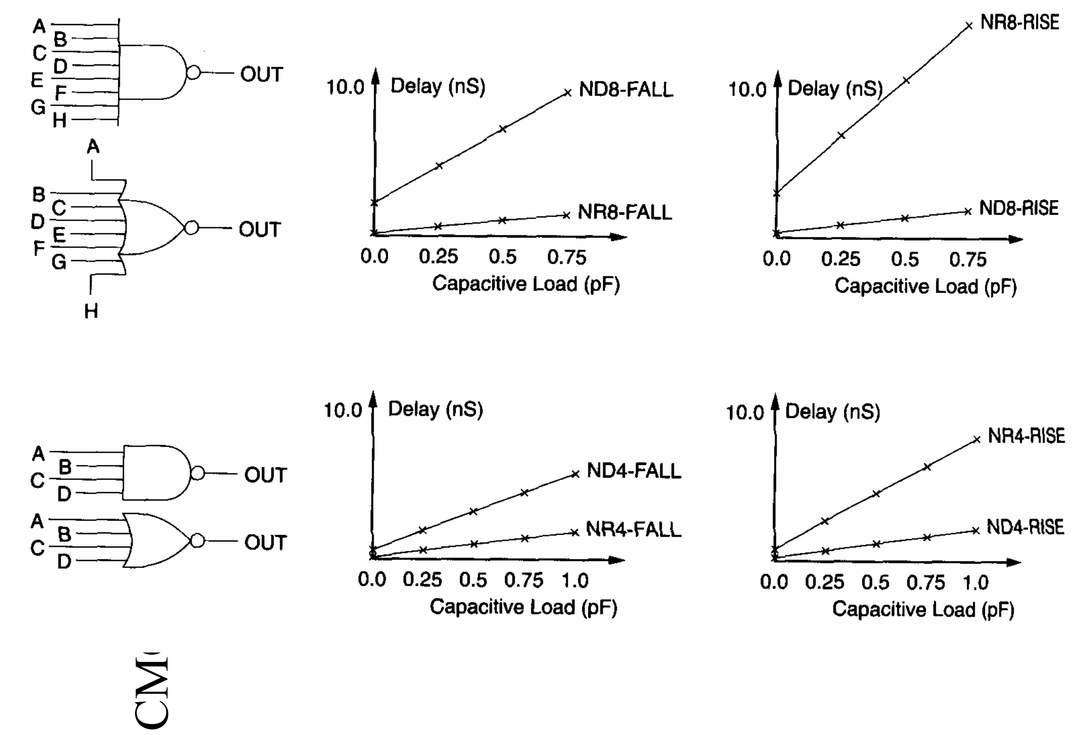
$$\boxed{t_{d1} = 2k\frac{L^2}{\mu \cdot V_{DD}}}$$

2.1 Switching characteristics review

Delay in CMOS gates

	NOR	NAND
PMOS	SERIES	PARALLEL
NMOS	PARALLEL	SERIES



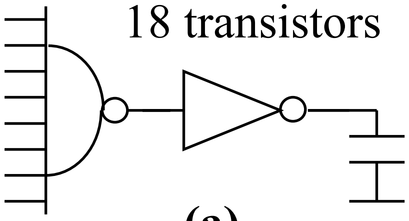


NAND gates are preferred than NOR ones

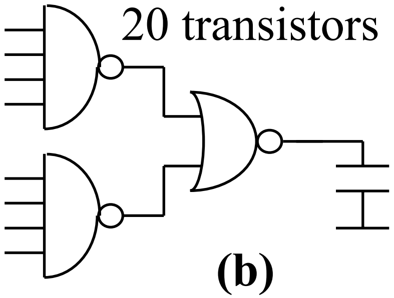
2.1 Switching characteristics review

Example:
8-input AND
loaded with
$C_L = 1$ pF

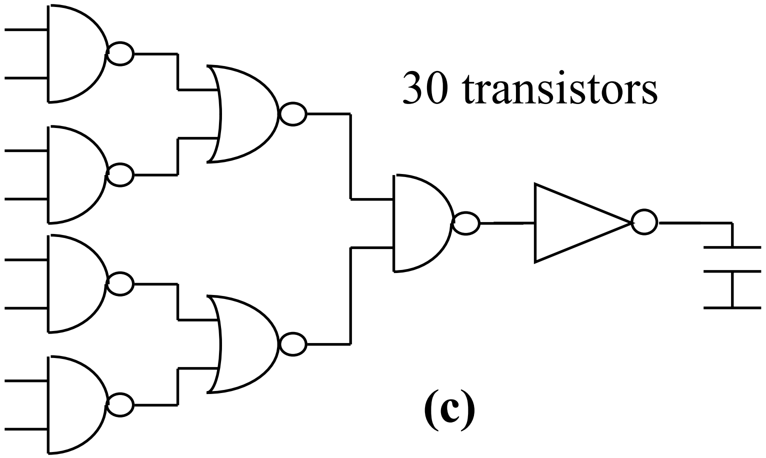
Circuit	Delay of stage 1	Delay of stage 2	Delay of stage 3	Delay of stage 4	Total delay
(a) 18 TRT	2.82 ns NAND8 Fall	3.37 ns INV Rise	-	-	6.2 ns (6.5 ns SPICE)
(b) 20 TRT	0.88 ns NAND4 Fall	4.36 ns NOR2 Rise	-	-	5.24 ns (5.26 ns SPICE)
(c) 30 TRT	0.31 NAND2 Fall	0.4 ns NOR2 Rise	0.32 ns NAND2 Fall	2.17 ns INV Rise	3.19 ns (3.46 ns SPICE)



(a)



(b)



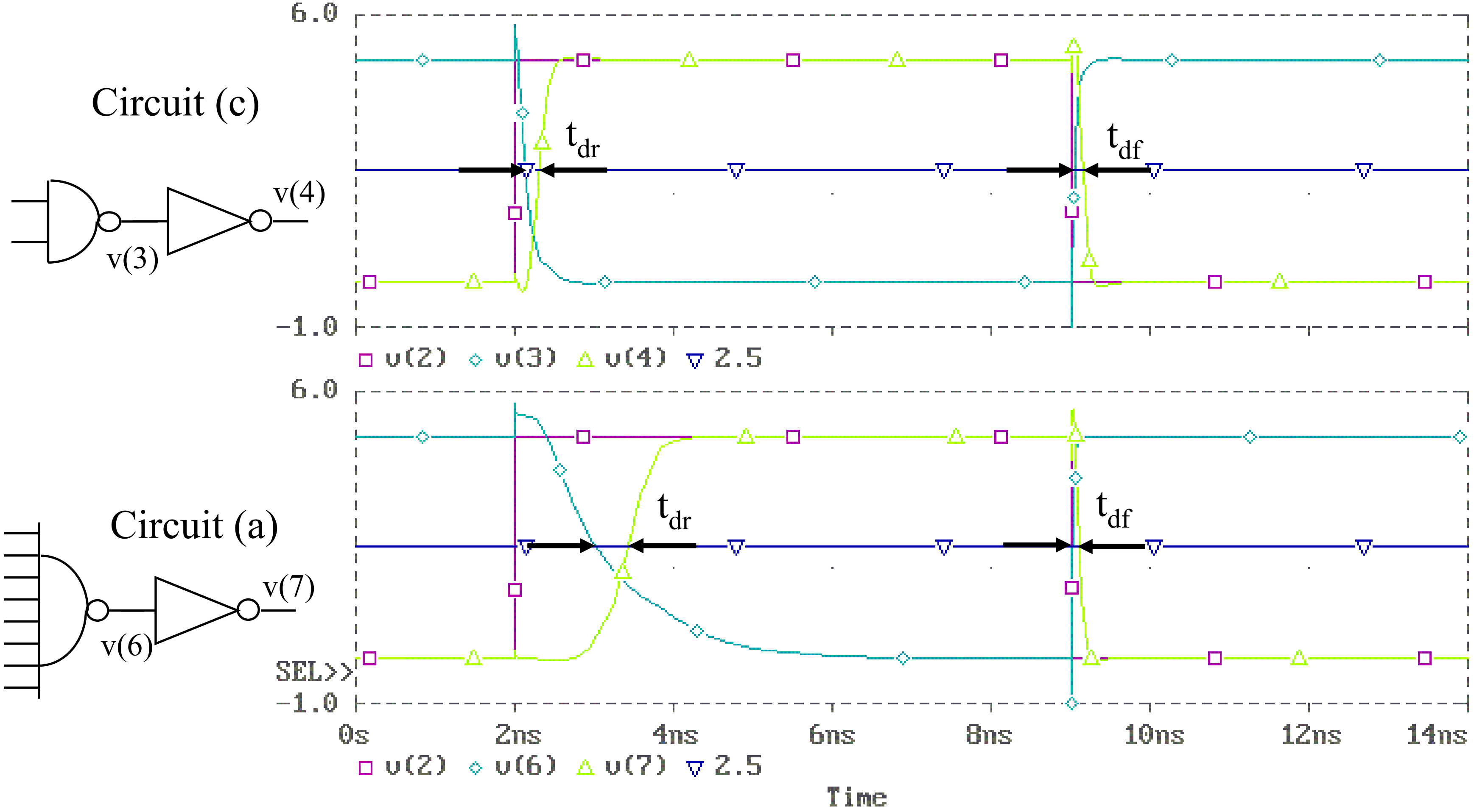
(c)

Delay-area tradeoff (number of transistors)
For speed ⇒ Reduce number of series devices!

2.1 Switching characteristics review

Delay comparison (simulation)

Models of an industrial 1-µm CMOS technology

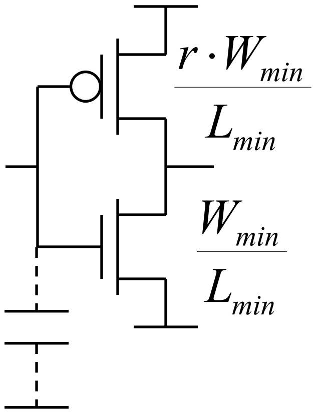


2.2 Delay, logical effort and buffering

Gate-level delay modeling in CMOS technology

2.2.1 Definitions

- ***Unit inverter buffer*:**



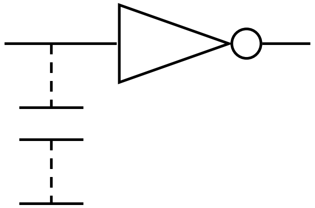
r: Aspect ratio

Ideally $\boldsymbol{r} = \frac{\mu_n}{\mu_p}$

(Typical value: r = 2)

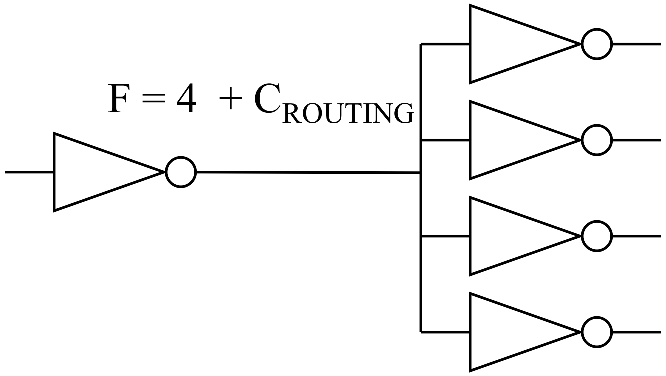
$\boxed{W_{min} \equiv L_{min} \equiv 1}$

- **Input capacitance (unit inverter):**

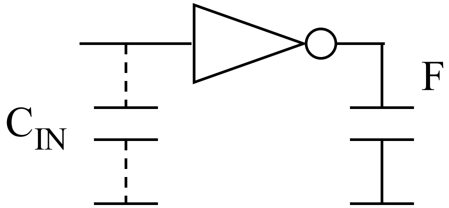


$C_{IN} = C_1 \equiv 1$

- **Absolute Fanout**: Unit load addition (1 load = 1 unit inverter)

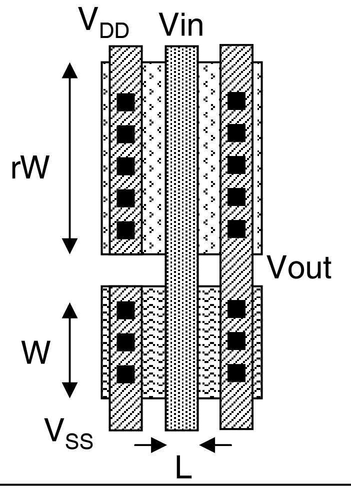


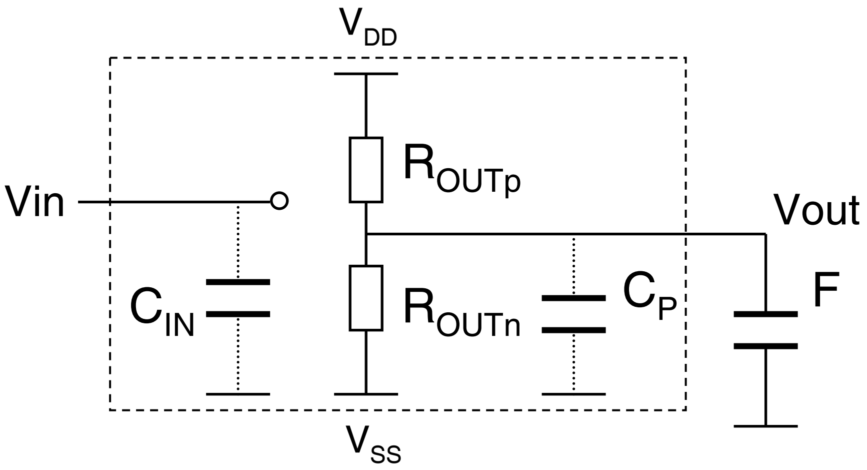
- **Relative fanout**: Absolute fanout and input capacitance ratio



$$\boxed{f = \frac{F}{C_{IN}}}$$

- **Output resistance and parasitic capacitance**





$$R_{OUT} = R_0(V_{IN})\frac{L}{W}$$

$$C_{IN} = (r+1)\cdot C'_{OX}\cdot W\cdot L$$

$$\Rightarrow \tau_{inv} = R_{OUT}\cdot C_{IN} = (r+1)R_0\cdot C'_{OX}\cdot L^2$$

τ_{inv}: Technological constant. Independent of W!

$$C_P = (r+1)\cdot C_{D0}\cdot W$$

C_P: Parasitic capacitance
R_0: Minimum transistor resistance
C'_{ox}: Oxide capacitance/area
C_{D0}: MOS drain capacitance/width

If L= L_{min} = 1 $\Rightarrow$ $\tau_{inv} = (r+1)R_0\cdot C'_{OX}$

➢ Delay estimation with time constants

$$\tau_d = \underbrace{R_{OUT}\cdot F}_{\substack{\text{External parasitic delay}\\ \text{(Effort)}}} + \underbrace{\underbrace{R_{OUT}\cdot C_P}_{\text{Internal parasitic delay}} + \underbrace{\tau_q}_{\text{Non-ideal delay}}}_{\text{(Parasitic)}}$$

τ_q: Extra delay due to

- Internal parasitic capacitances
- Input signal slew-rate
- MOS subthreshold voltage (V_T)

Delay t_d (50 % crossing) is proportional to the time constant τ_d:

$$t_d = k_1\tau_d = \underbrace{t_e}_{\text{Effort delay}} + \underbrace{t_p}_{\text{Parasitic delay}} \qquad where \quad k_1 = \ln 2$$

$$\boxed{\begin{aligned} t_e &= k_1 R_{OUT} C_{IN}\frac{F}{C_{IN}} = t_{e1} f \\ t_p &= k_1 (R_{OUT} C_P + \tau_q) \end{aligned}}$$

, where $\underbrace{f}_{\text{Relative fanout}} \equiv \frac{F}{C_{IN}}$ $\quad \underbrace{t_{e1}}_{\text{Unit effort delay}} \equiv k_1 \cdot \tau_{inv}$

- **Total delay (t_d)**:

$$\boxed{t_d = t_e + t_p = t_{e1} f + t_p}$$

- **Parasitic delay(t_p)**: Signal propagation inside the gate (without load).

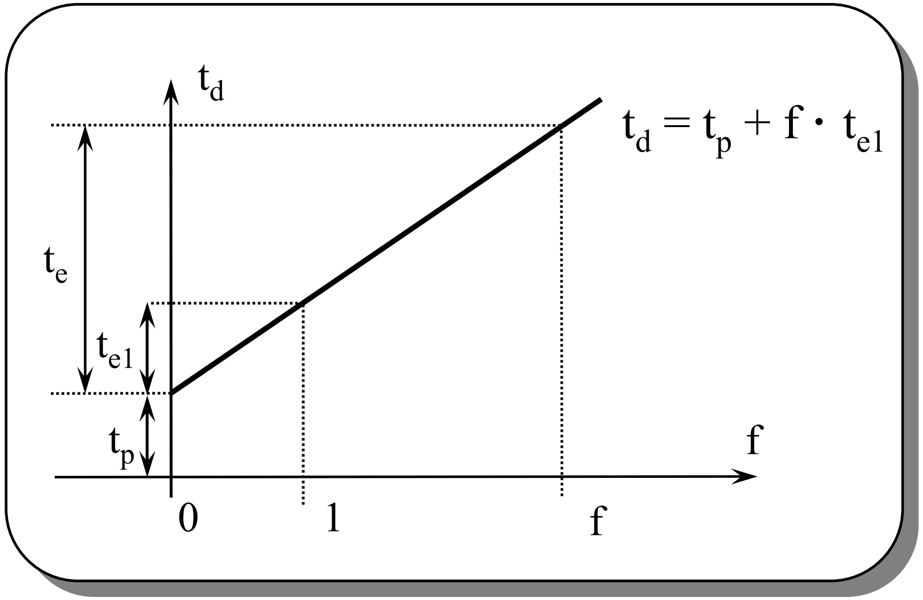
$$\boxed{t_p = k_1 (R_{OUT} C_P + \tau_q)}$$

- **Effort delay (t_e)**: Produced by the load (relative fanout)

$$\boxed{t_e = t_{e1} f}$$

$$t_{e1} = k_1 \tau_{inv}$$

$$f \equiv \frac{F}{C_{IN}}$$



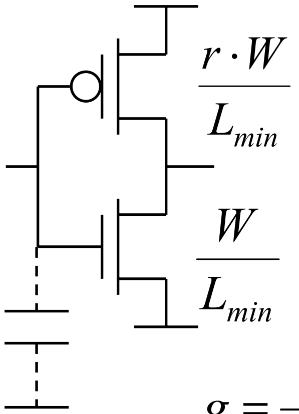
2.2 Delay, logical effort and buffering

2.2.2 Logical effort

Logic gate delay increase factor compared to a unit inverter

$$\boxed{g \equiv \frac{R_{OUT} C_{IN}}{\tau_{inv}}}$$

For an inverter with L = L_{min} and arbitrary W:

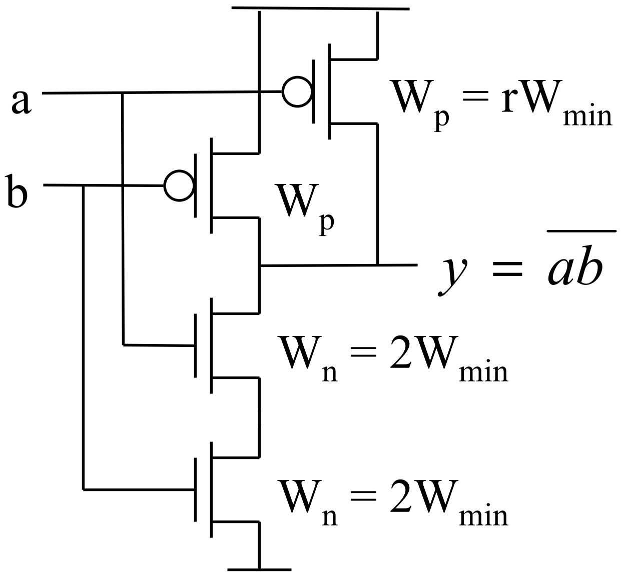


$$g \equiv \frac{\tau_{inv}}{\tau_{unit\,inv}}$$

$$g = \frac{R_0 \frac{L_{\min}}{W}(r+1)C'_{OX}WL_{\min}}{R_0 \frac{L_{\min}}{W_{\min}}(r+1)C'_{OX}W_{\min}L_{\min}} = 1$$

Logical effort is 1 because W cancels out.

Logical effort calculation example: 2-input NAND gate.



a) R_{OUT} is made equal to the inverter’s
b) A single-input C_{IN} is calculated

$$g = \frac{R_{OUT} C_{IN}}{\tau_{inv}} = \frac{R_0 \cdot (r+2) \cdot C'_{OX}}{R_0 \cdot (r+1) \cdot C'_{OX}} = \frac{r+2}{r+1}$$

E.g., for r = 2, $\boldsymbol{g = \frac{4}{3}}$

2.2 Delay, logical effort and buffering

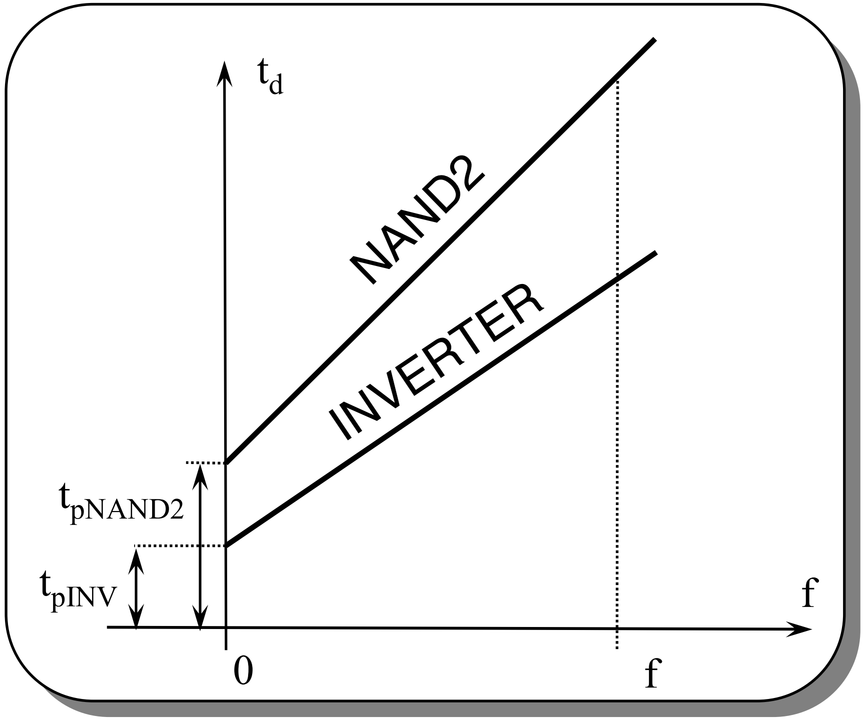
- Logical effort delay is an effort delay
- Non-minimum W transistors increase parasitic delay proportionally

Modified t_d expression including logical effort:

$$\boxed{t_d = t_{e1} \cdot f \cdot g + t_p}$$

Comparative delays for an inverter and for a 2-input NAND gate:

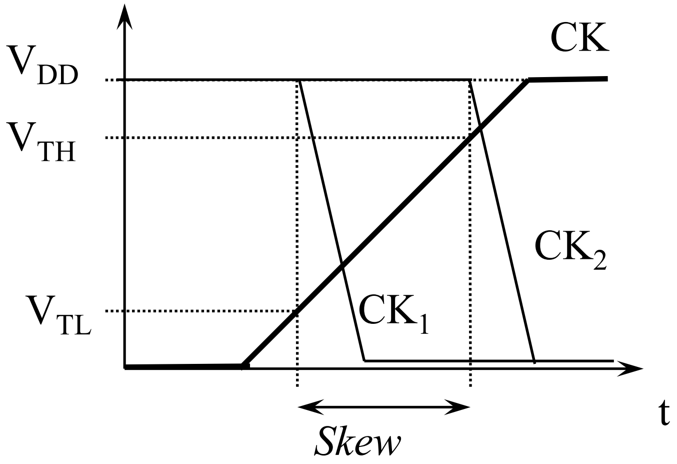
$g_{NAND2} = 4/3$
$t_{pNAND2} = 2 \cdot t_{pINV}$

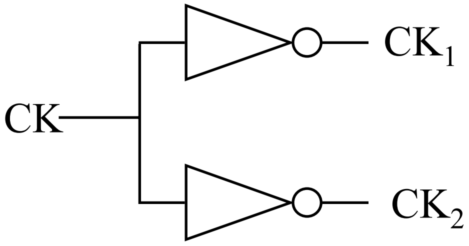


2.2.3. *Buffering*

➢ Slow edge issue

- Combined with different threshold gates it produces skew
- **Clock skew** ⇨ data captured at different times

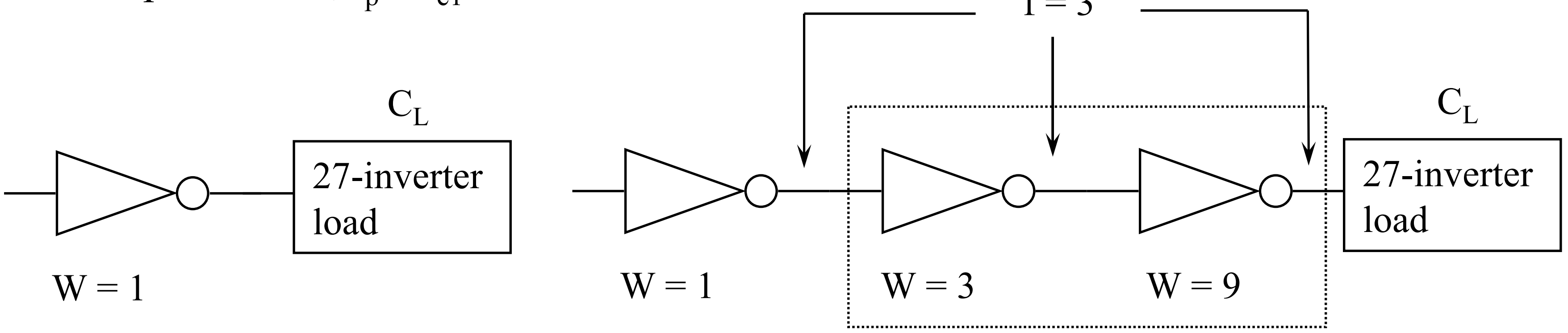




Buffering strategies reduce skew and clock propagation delay.

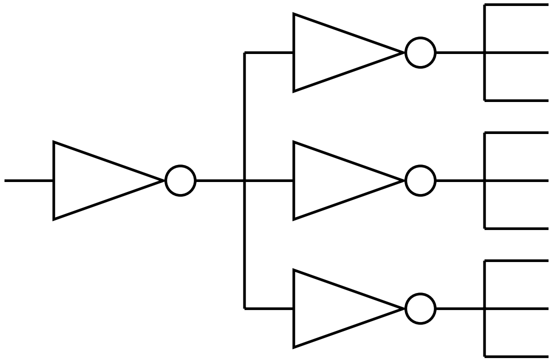
Classes of buffering:

1. Geometric. Inverters are progressively scaled to adapt the driver to the load.
Example: F = 27, $t_p = t_{e1} = 1$



Without buffering: $t_d = 1 + 27 * 1 = 28$
With buffering: $t_d = 1 + 3 * 1 + 1 + 3 * 1 + 1 + 3 * 1 = 12$

2. Tree. The same calculation method is applied,
But output nodes are separated and
distributed to different parts of the circuit.

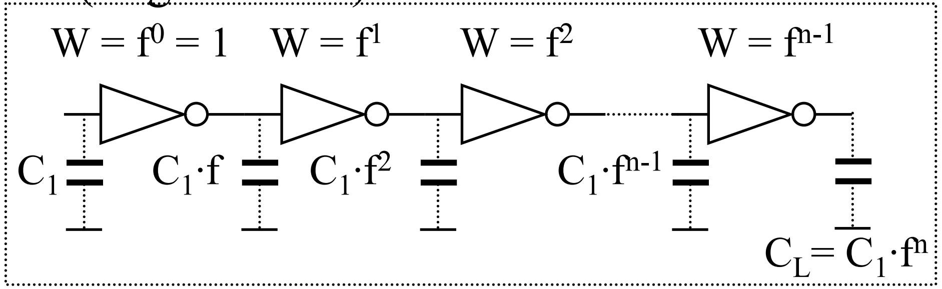


2.2 Delay, logical effort and buffering

Theoretically optimal relative fanout (for t_d minimum and considering only effort delay):

$$t_d = n \times (f \times t_{e1})$$

For constant f, it can be related with n (stage number):

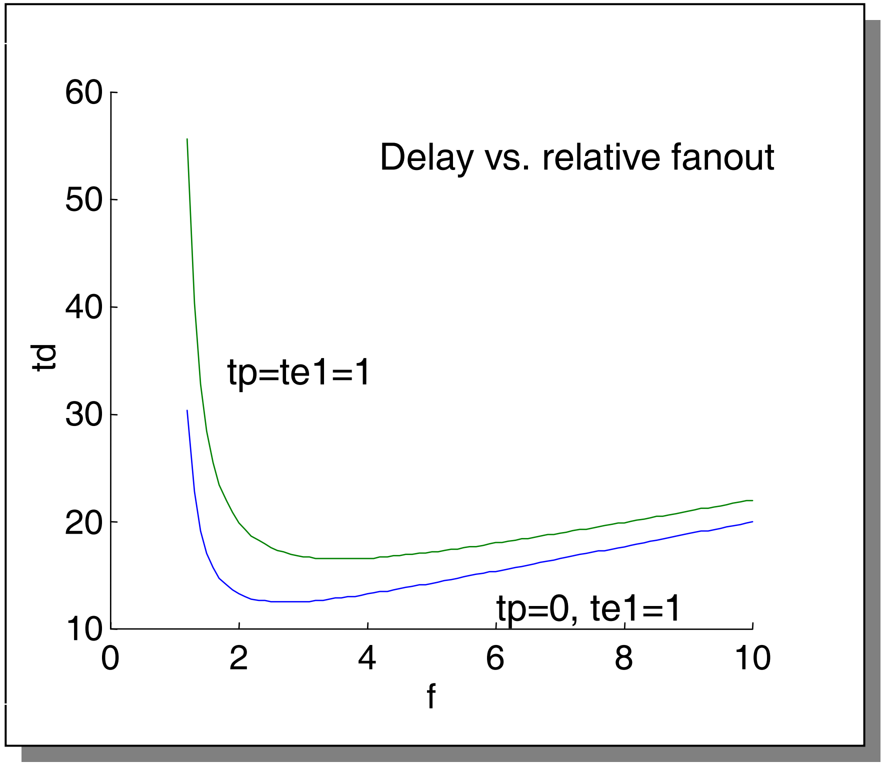


$$\boxed{f = \sqrt[n]{\frac{C_L}{C_1}}}$$

$$\boxed{\frac{\partial t_d}{\partial f} = 0 \Rightarrow f = e = 2.718\ldots}$$

If parasitic delays are accounted, the equation has no analytic solution

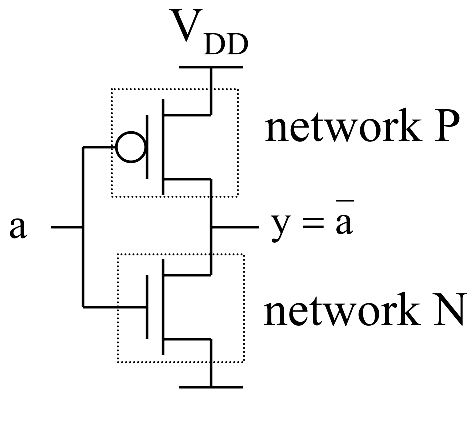
Numerical solution:



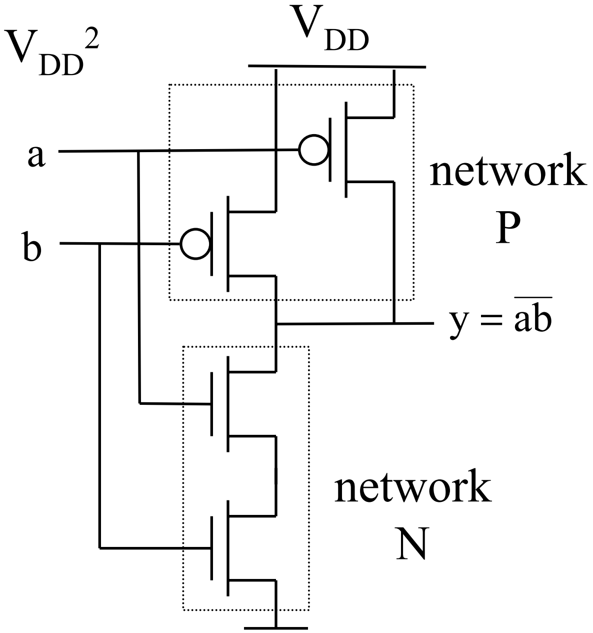
2.3 Logic structures

2.3.1 Complementary (static) CMOS design

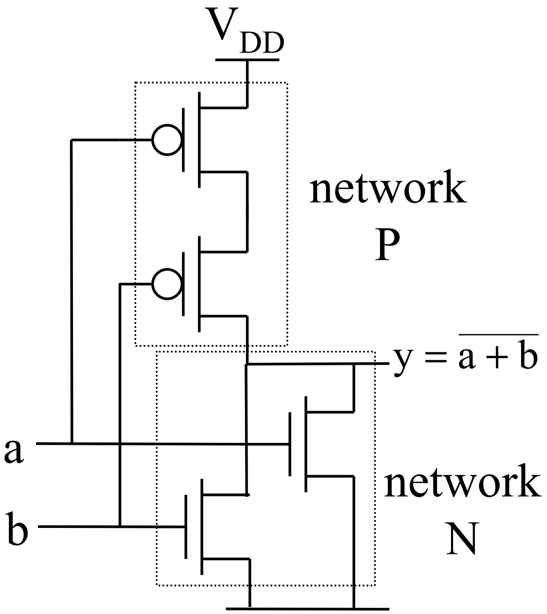
- Robust
- Good noise immunity
- Fast design (standard cells)
- Series-parallel duality in PMOS-NMOS networks
- *2n* transistors for *n* inputs
- Dynamic power: $P_d = C_L \cdot f \cdot V_{DD}^2$



INVERTER

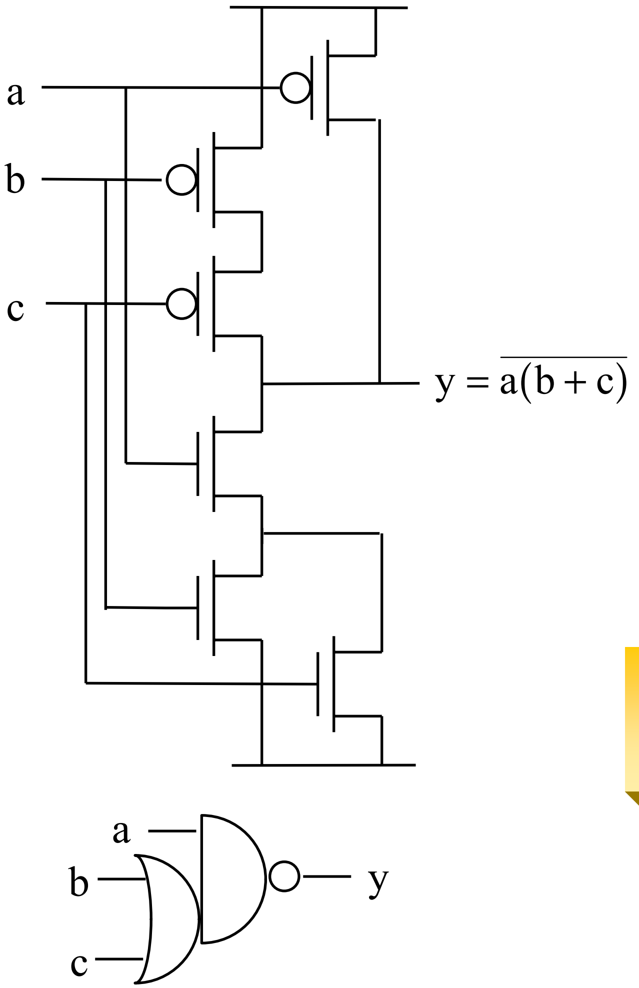


NAND

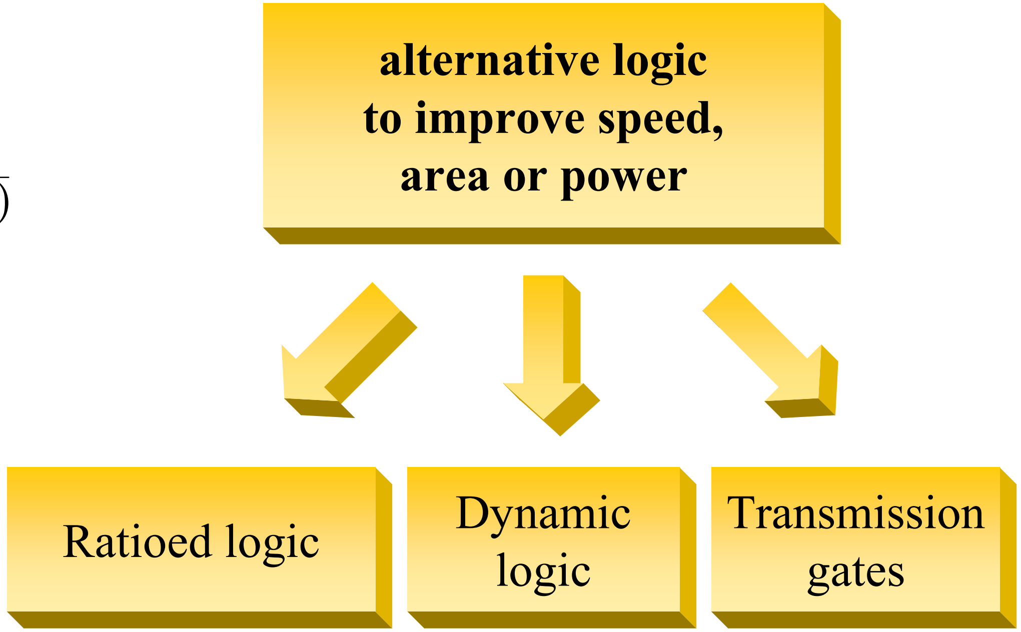


NOR

2.3 Logic structures



OR-NAND
COMPOUND GATE



alternative logic to improve speed, area or power

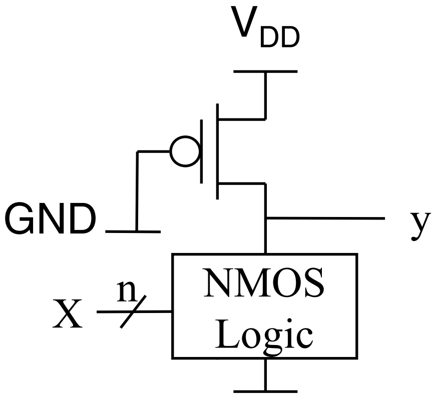
Ratioed logic

Dynamic logic

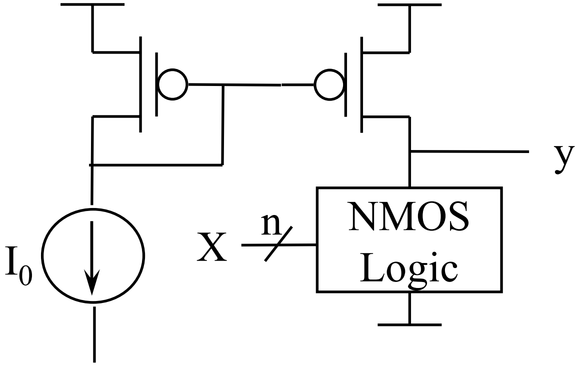
Transmission gates

2.3.2 Pseudo-NMOS Logic

- Single PMOS transistor
- βn/βp Ratioed logic
- Advantages
  - Area: *n + 1* transistors
  - Reduced capacitive load
- Drawbacks
  - PMOS Reduced gain for good noise margin
  - Slow rise time
  - Static power dissipation for output *LOW*

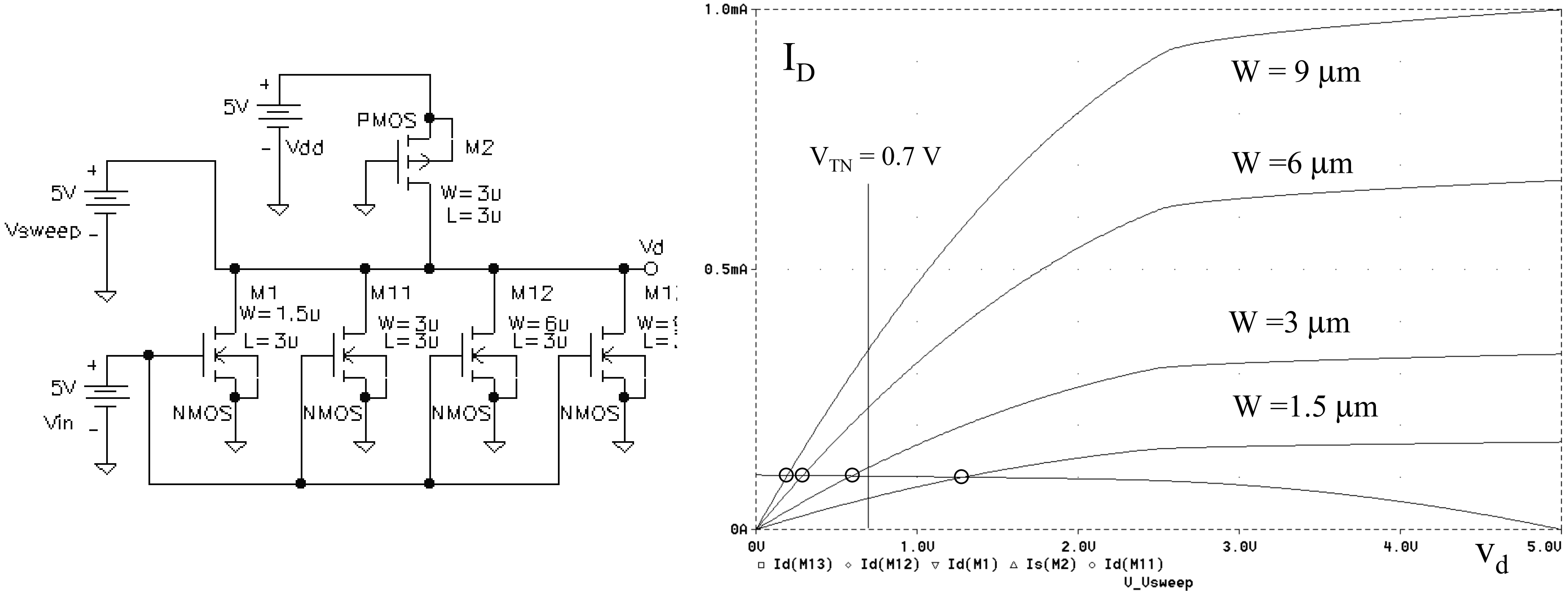


Pseudo-NMOS



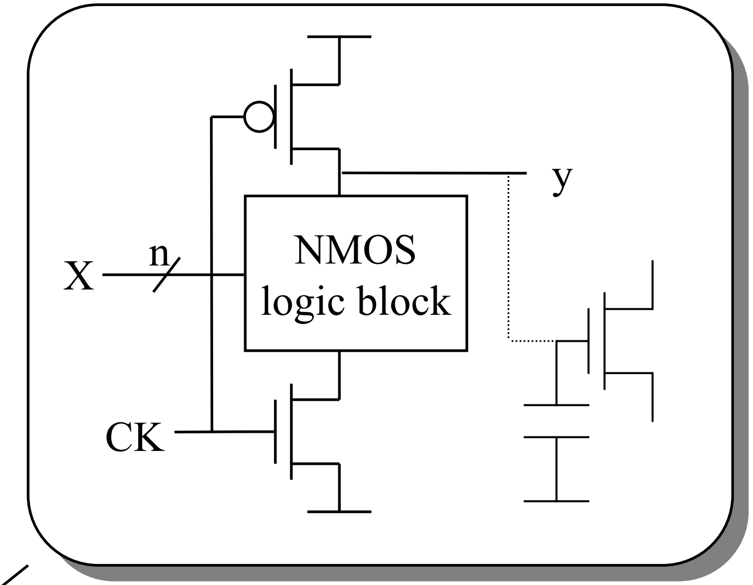
Current-limited pseudo-NMOS

Output-low simulation for several NMOS transistor channel width values



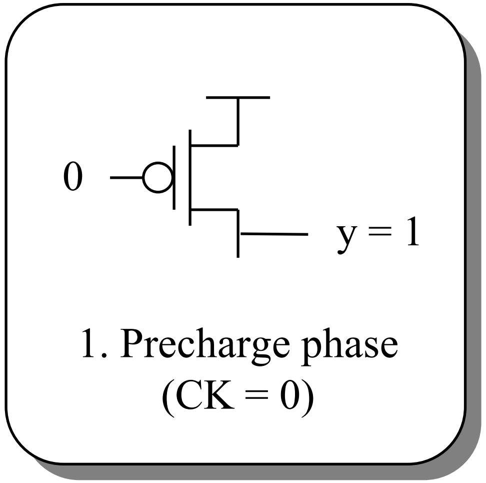
2.3.3 Dynamic CMOS logic

- Area occupancy : *n + 2* transistors for *n* inputs
- Operates with clock
- Input capacitance: same as pseudo-NMOS
- Input capacitance stores the electric charge.
- Drawbacks:
  - Inputs must be fixed during evaluation phase
  - Gates cannot be directly cascaded

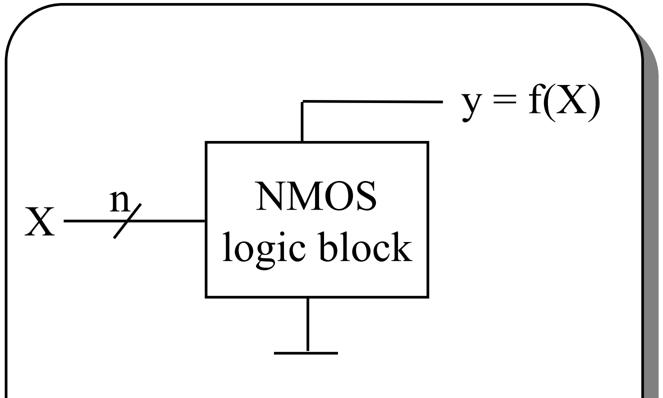


CK = 0

CK = 1

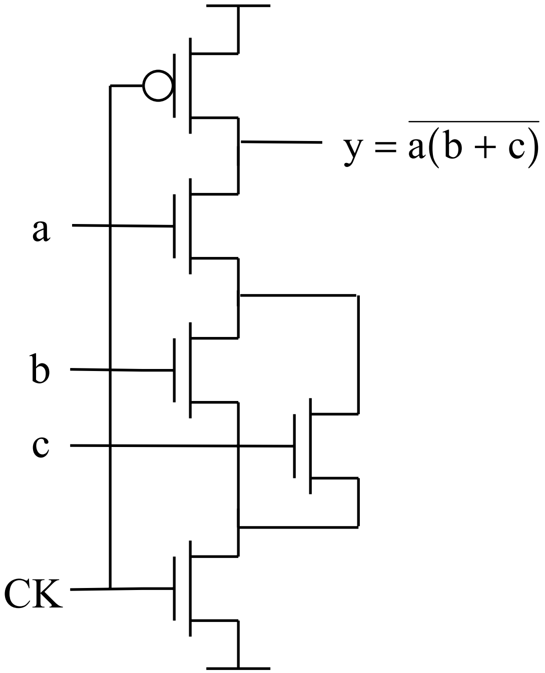


1. Precharge phase (CK = 0)

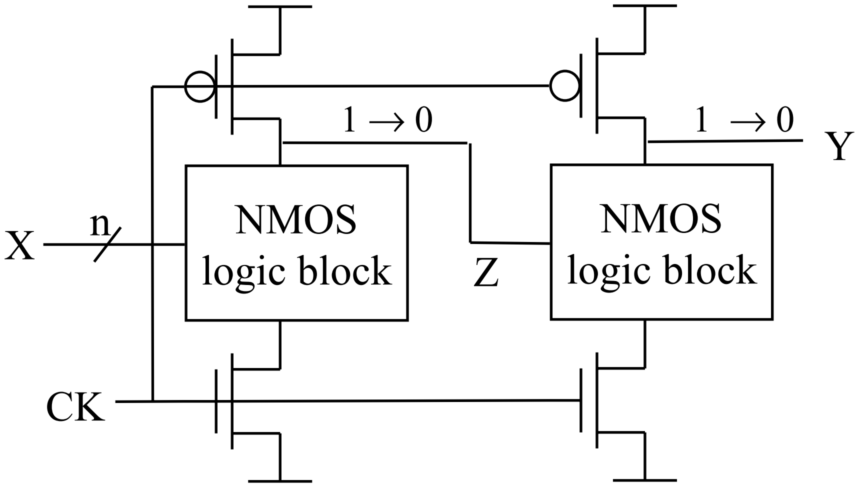


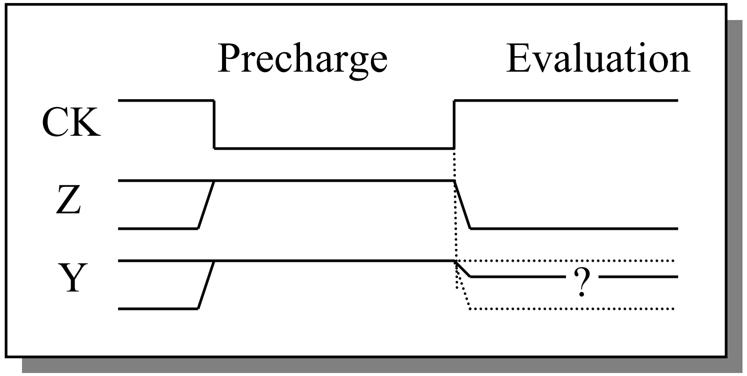
2. Evaluation phase (CK = 1)

Example:



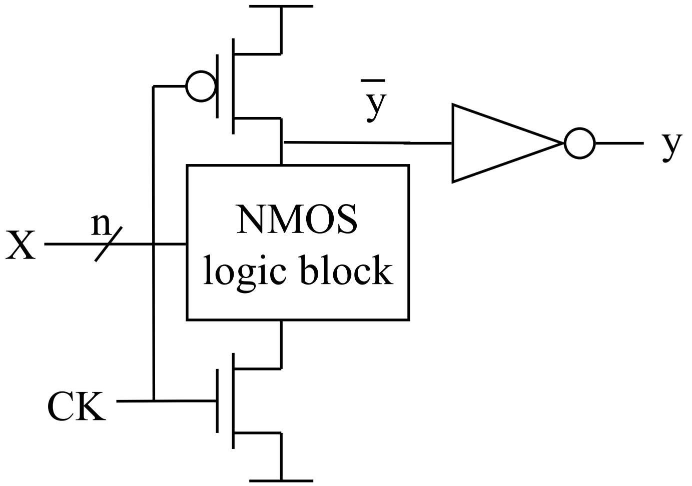
Erroneous design : Direct cascade connection

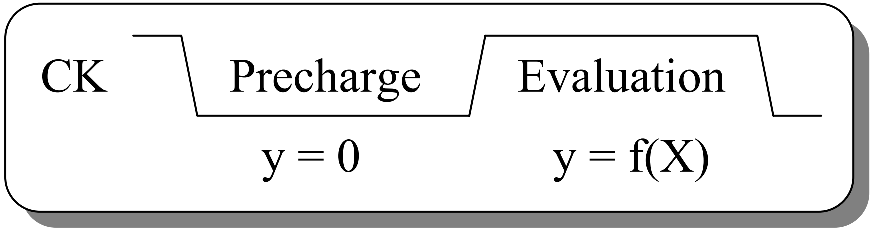




Domino CMOS logic

- Two phases : Precharge and evaluation
- An output inverter is added
- Area: *n + 4* transistors for *n* inputs
- Can be cascaded





- Each stage produces evaluation of next one (domino)

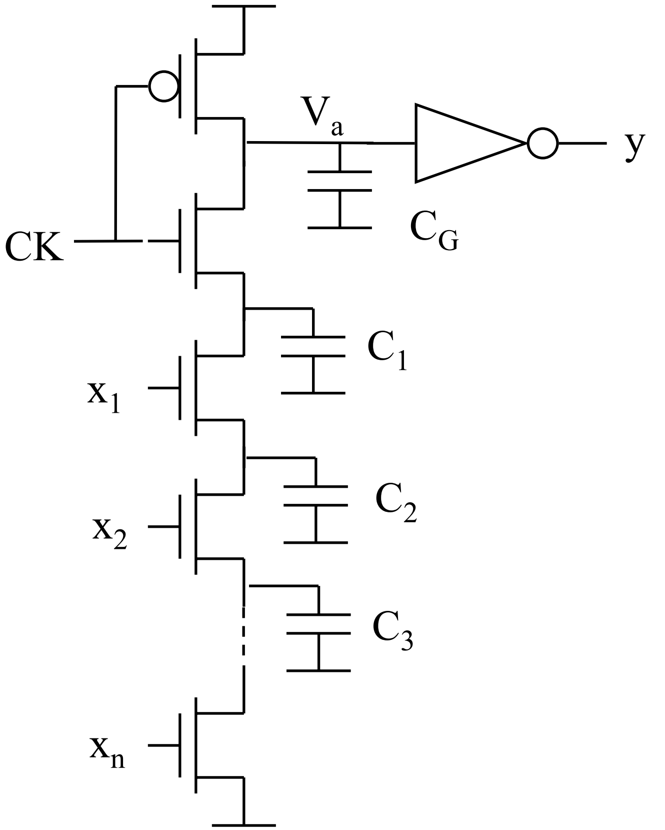
- Precharge ($CK = 0$):
  - $y = 0 \Rightarrow$ following NMOS blocks OFF

- Evaluation ($CK = 1$):
  - Output is conditionally discharged
  - Only one transition 0 ➔ 1

- Total stage delay must be lower than clock evaluation time.

- Limitations
  - Output buffer is always necessary
  - Only non-inverting logic Is possible
  - Charge redistribution
  - Floating nodes

Charge redistribution effect



For $x_1..x_{n-1} = 1$, $x_n = 0$, assuming $V_{C1}...V_{Cn} = 0$ V, during the evaluation phase,

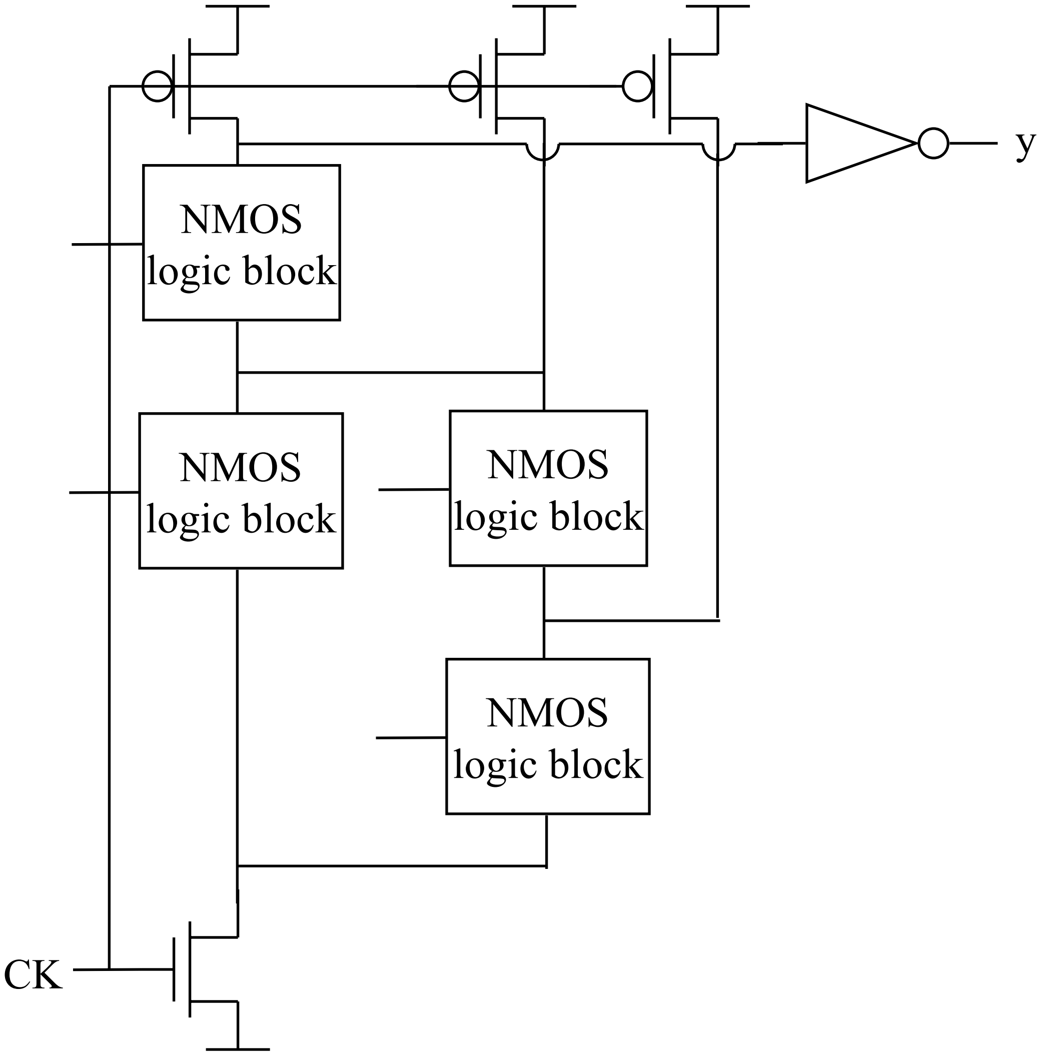
$$\left.\begin{aligned} Q^- &= C_G \cdot V_{DD} \\ Q^+ &= (C_G + \sum_{i=1}^{n} C_i) \cdot V_a \end{aligned}\right\} \quad V_a = \frac{C_G}{C_G + \sum_{i=1}^{n} C_i} V_{DD}$$

Numerical example :

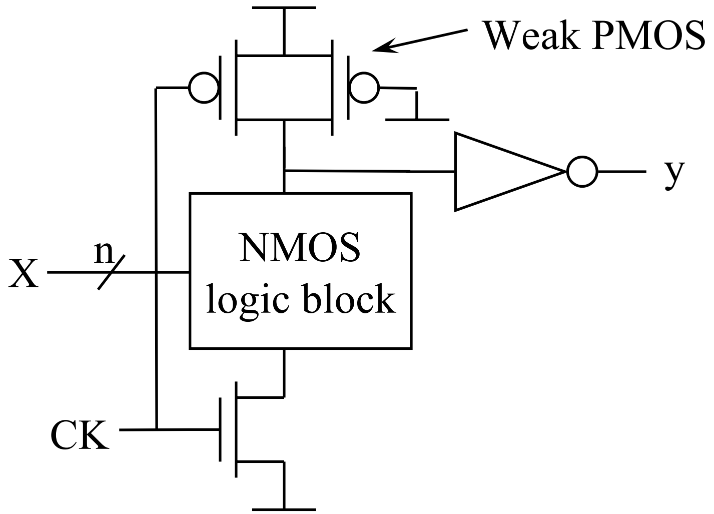
For *n = 6; C_G = 3 C_1 = ... = 3 C_n*

$$V_a = \frac{V_{DD}}{3}$$

Reducing charge redistribution: Precharge transistors



- Static domino logic version :

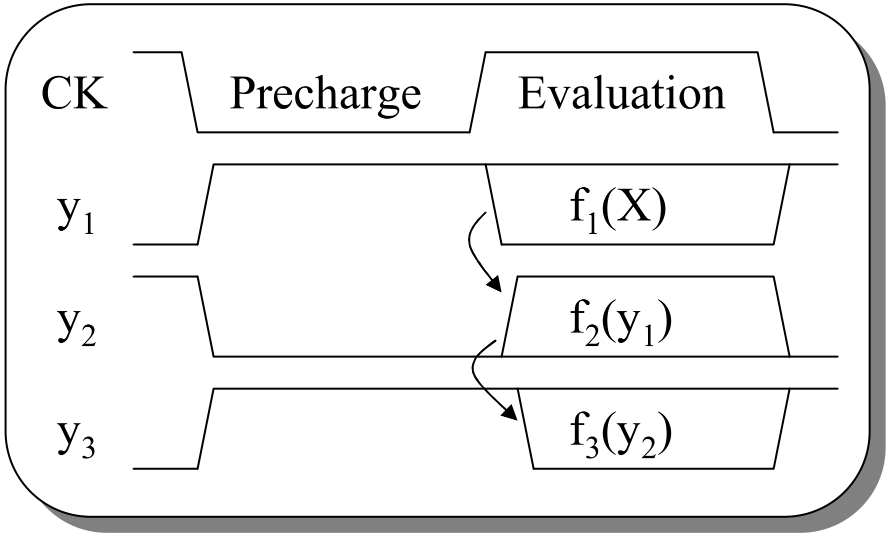


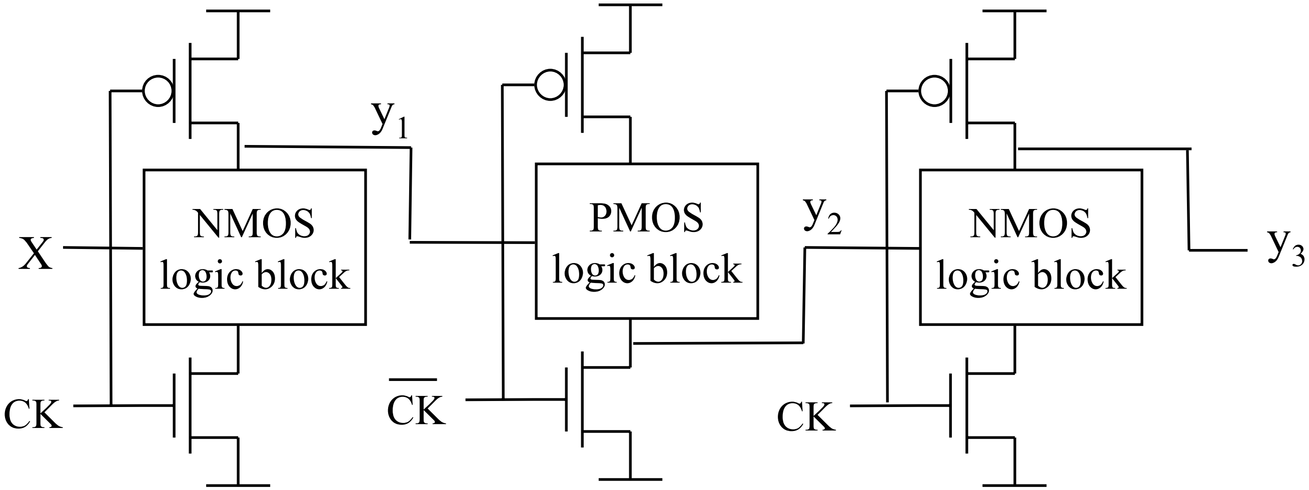
- Fixing CK = 1
  - Static operation (at low frequency)
  - Pseudo-NMOS (slow pull-up)

- Dynamic operation
  - Extra pull-up current
  - Introduces extra parasitic capacitance (pull-up drain).

NP domino logic (or Zipper)

- Alternates N and P blocks.
- Alternates *1* and *0* precharging.
- Simultaneous precharge in all blocks
- Simultaneous evaluation in all blocks
- Area: *n* + *2* transistors for *n* inputs

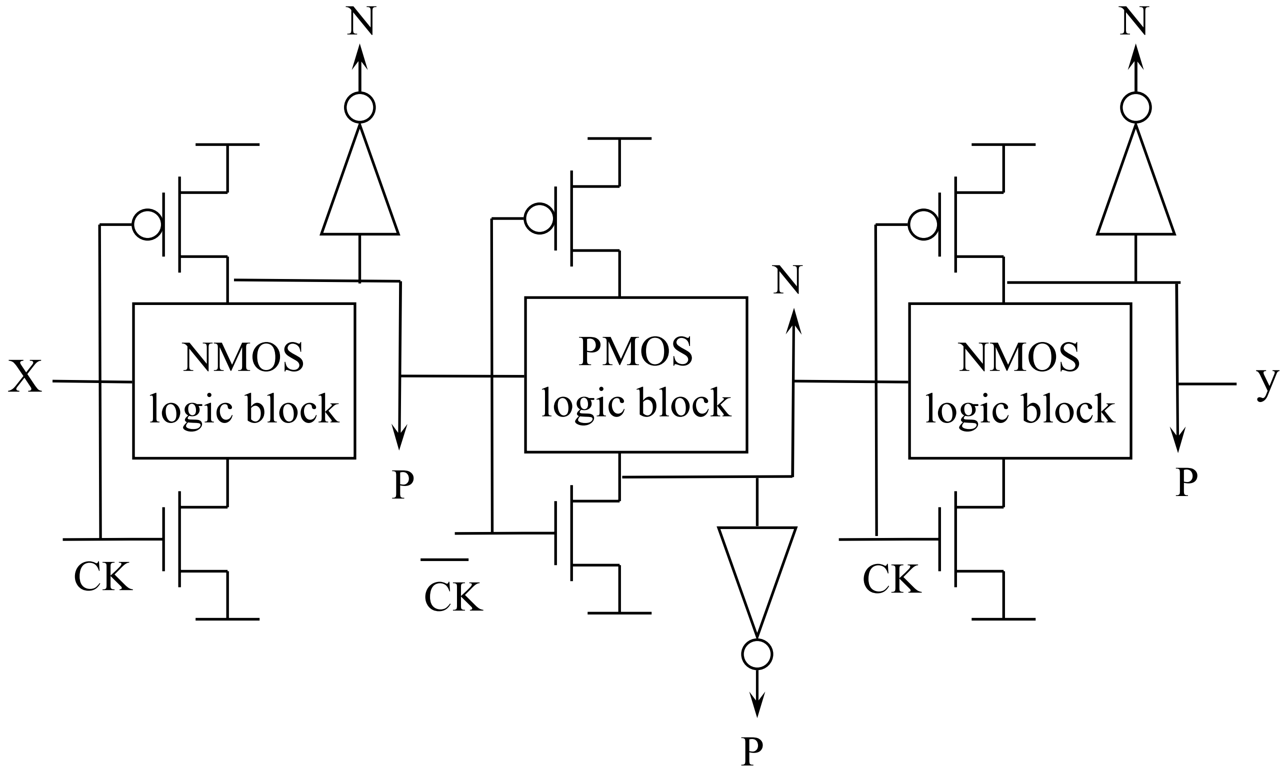




Domino connections

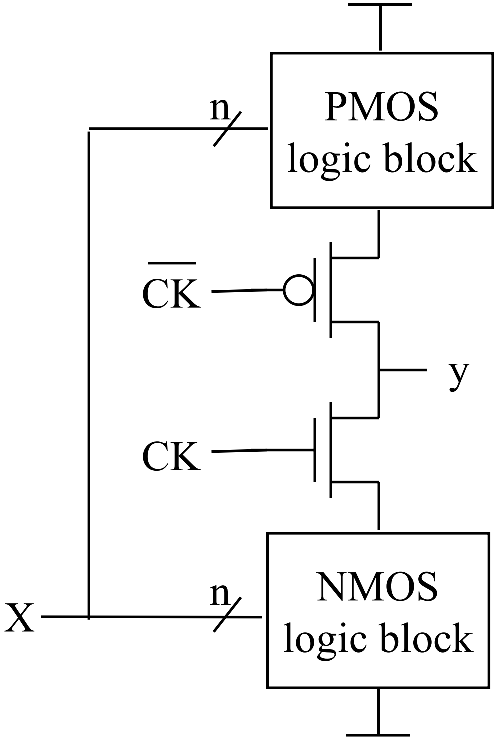
Options:

- •Alternating N and P blocks
- •Inverter insertion between equal-type blocks



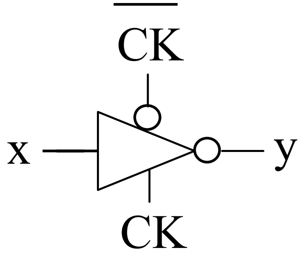
Clocked CMOS logic (C^2MOS)

- Application: latches in synchronized structures
- Input capacitance: The same as static CMOS
- Area: *2n + 2* transistors for *n* inputs
- An extra series transistor: slow

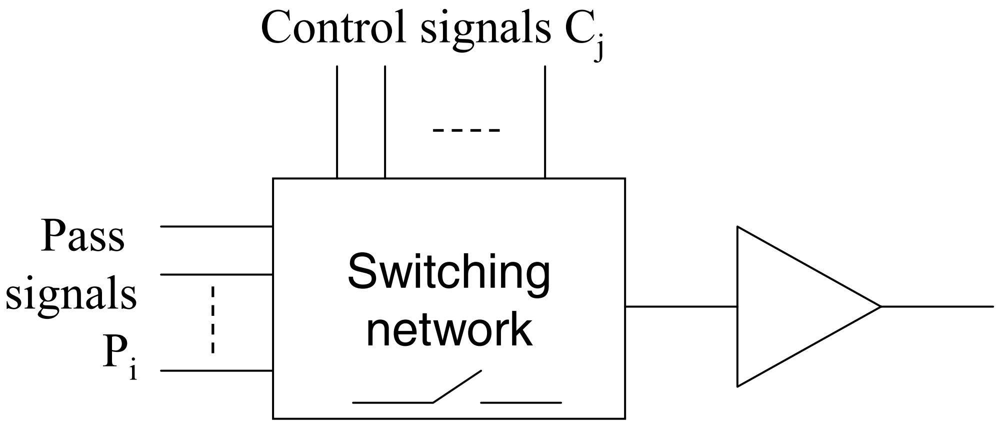


Example:

C^2MOS tri-state inverter buffer



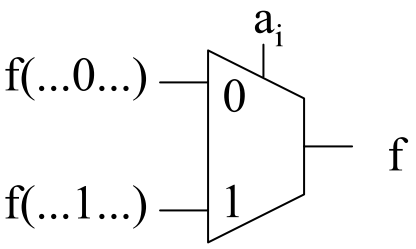
2.3.4 Pass-transistor logic



$$y = \sum_{i=1}^{n} P_i f_i\left(C_1,..,C_m\right)$$

- Multiplexer direct implementation
- Based on Boolean function expansion :

$$f(a_1,..,a_i,..,a_n) = \overline{a}_i f(a_1,..,0,..,\ a_n) + a_i f(a_1,..,1,..,\ a_n)$$

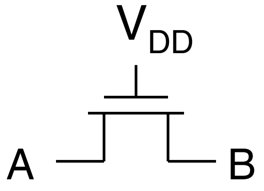


2.3 Logic structures

- Transistors as switches:

NMOS switch

- Good propagation of low level

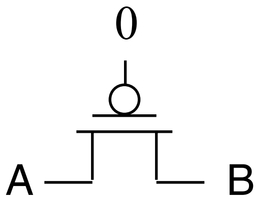


$0 \rightarrow 0$

$V_{DD} \rightarrow V_{DD}$-
V_{TN}

PMOS switch

- Good propagation of high level

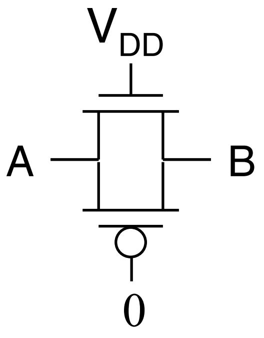


$0 \rightarrow 0 + |V_{TP}|$

$V_{DD} \rightarrow V_{DD}$

Complementary switch:

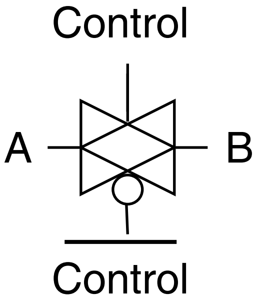
- Good propagation of both levels



$0 \rightarrow 0$

$V_{DD} \rightarrow V_{DD}$

- Symbol:



Example: XNOR gate

a	b	$\overline{(a \oplus b)}$	Pass signals
0	0	1	$\bar{a}$ ó $\bar{b}$ ó 1
0	1	0	a ó $\bar{b}$ ó 0
1	0	0	$\bar{a}$ ó b ó 0
1	1	1	a ó b ó 1

Karnaugh map :

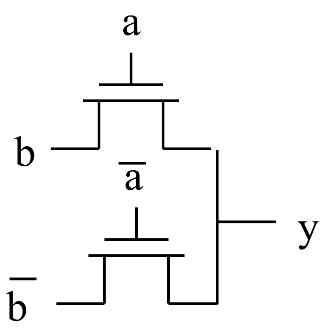
		a	
		0	1
b	0	$\bar{a}$ 1 $\bar{b}$	$\bar{a}$ 0 b
	1	a 0 $\bar{b}$	a 1 b

$$y = \bar{a}(\bar{b}) + a(b)$$

control signal (pass signal)

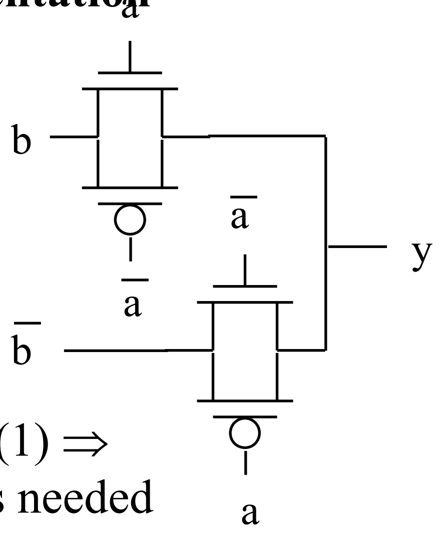
NMOS implementation

- Slow rising time



Complementary implementation

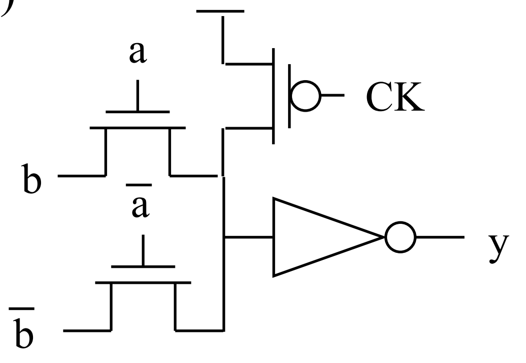
- Shorter rising time
- Longer falling time
- P pass function dual to N
- Good logic level's
- Major area
- For pass signals fixed to 0(1) ⇒ Only N(P)MOS transistors needed



Implementation with pull-up

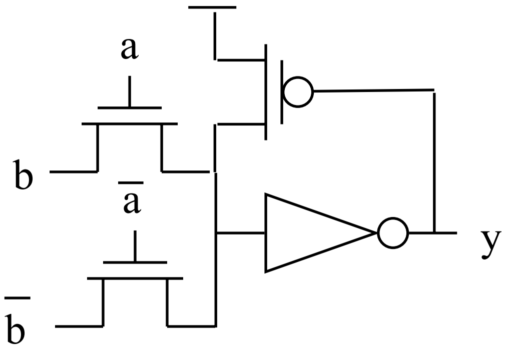
Dynamic version :

- It is not necessary to guarantee 1-terms (Z is enough)

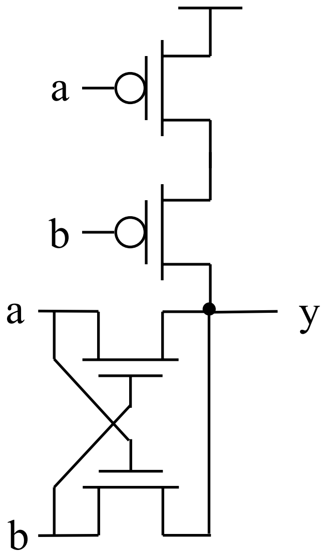


Static version :

- Weak PMOS is necessary
- 1-terms must be guaranteed



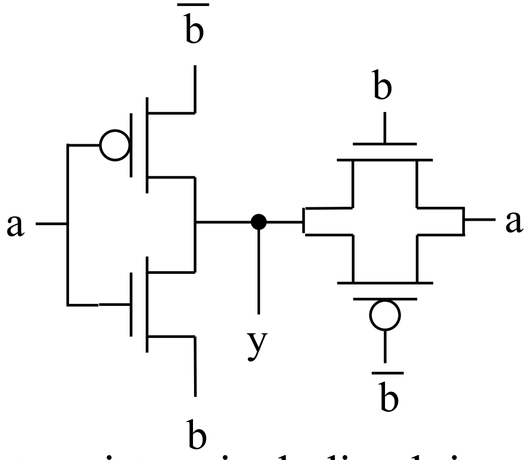
Crossed implementation



- $\overline{a}$, $\overline{b}$ are not required
- Pass and control signals are combined
- Drawback: a = b = 1 (NMOS propagating 1)

		a	
		0	1
b	0	$\overline{a}$ 1 $\overline{b}$	$\overline{a}$ 0 b
	1	a 0 $\overline{b}$	a 1 b

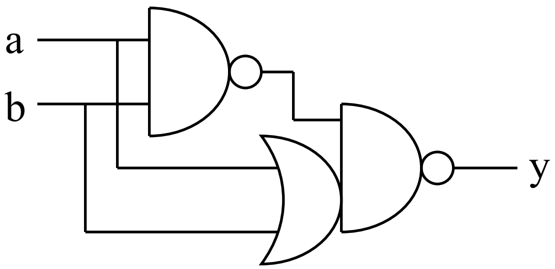
Hybrid implementation pass transistors / transmission gate



- Only 6 transistors including b inverter
- Output connected to appropriate pass transistor

		a	
		0	1
b	0	$\bar{a}$ 1 $\bar{b}$	$\bar{a}$ 0 b
	1	a 0 $\bar{b}$	a 1 b

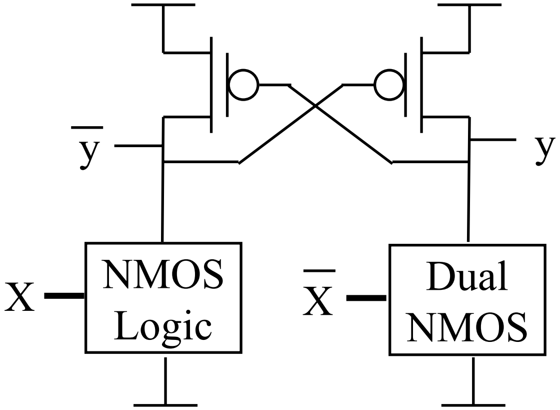
XNOR Static implementation



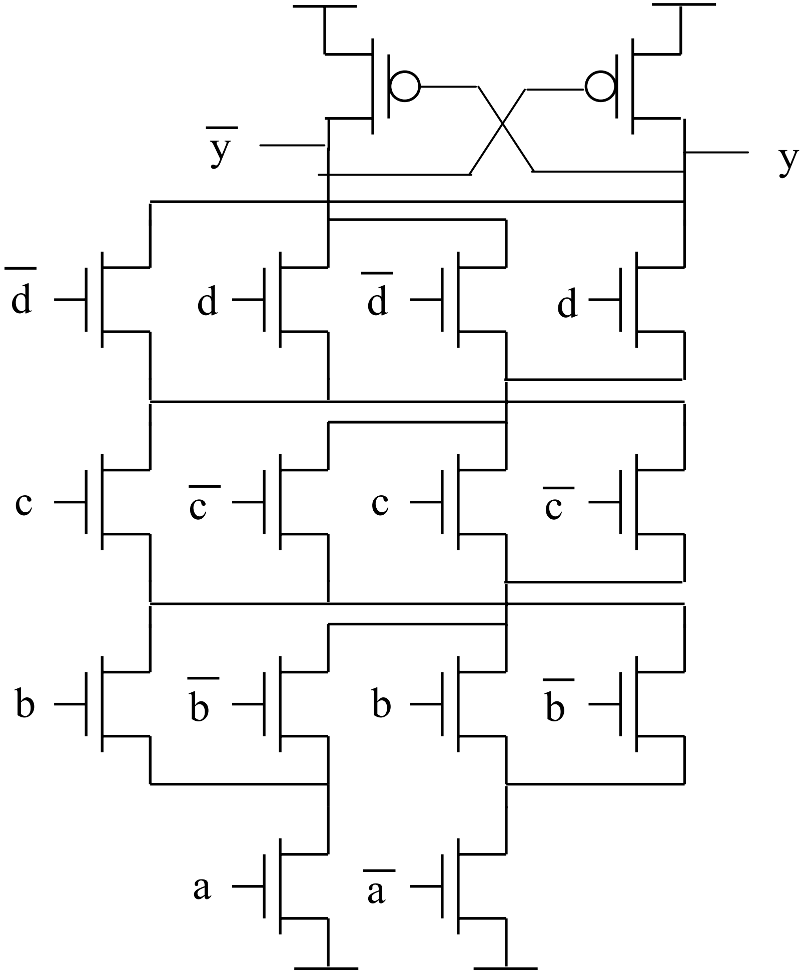
- 10 transistors are required

2.3.5 Cascade Voltage Switch Logic (CVSL)

- Differential logic (the signal and its complement)
- Basic gate:

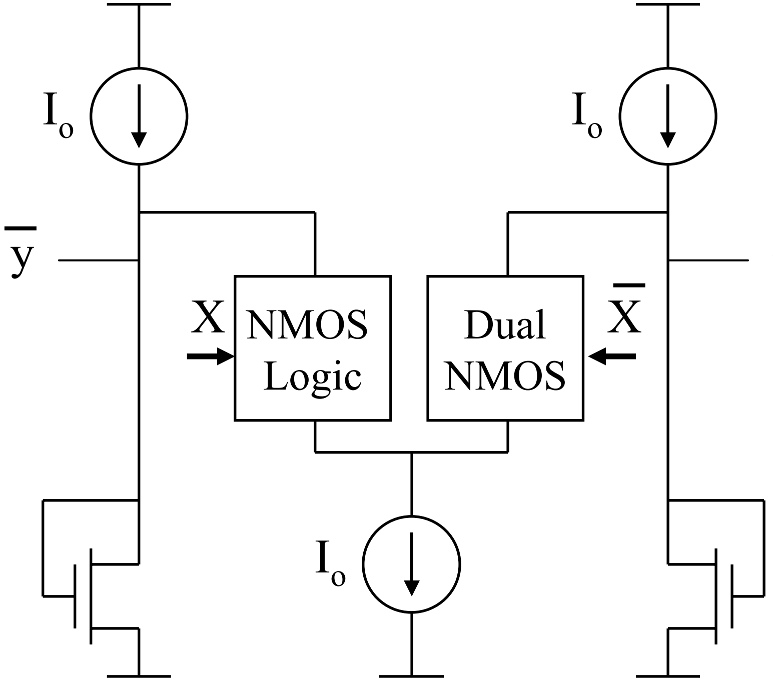


- Example: 4-input XOR

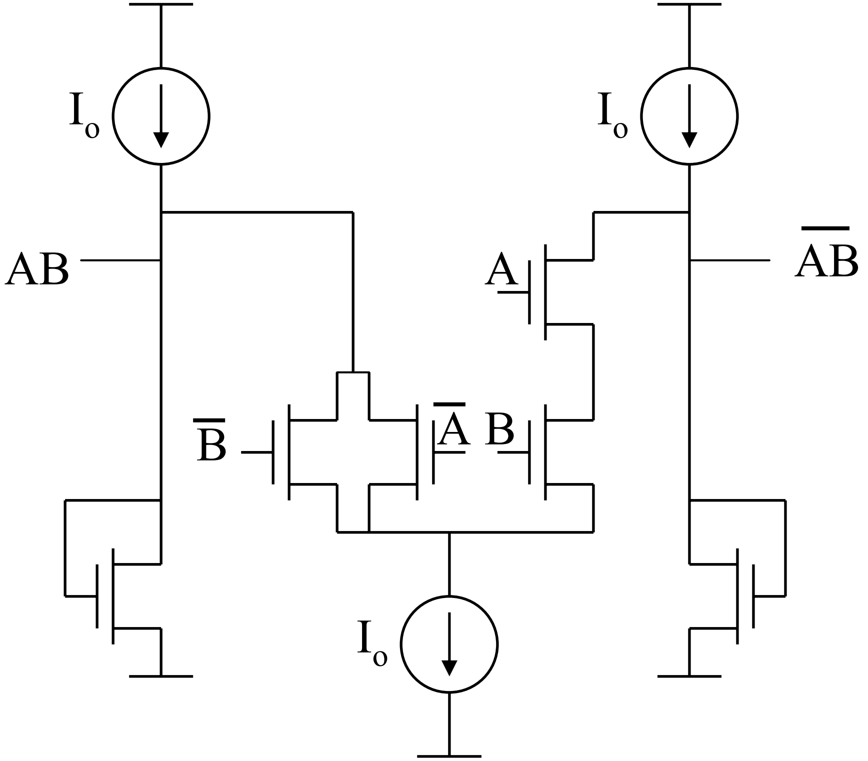


2.3.6 Folded Source-Coupled Logic (FSCL)

- Low digital noise differential logic
- Constant current, independent on transitions



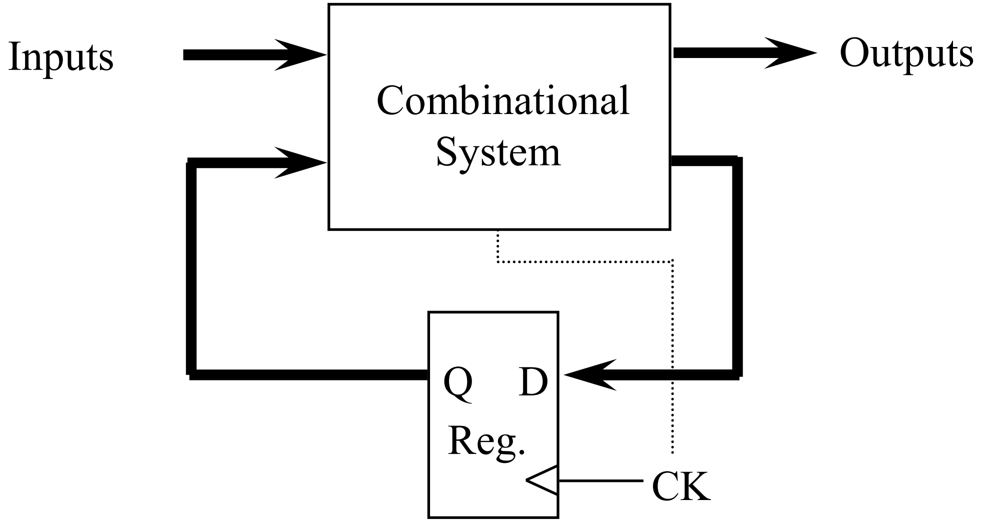
Example: AND/NAND Gate



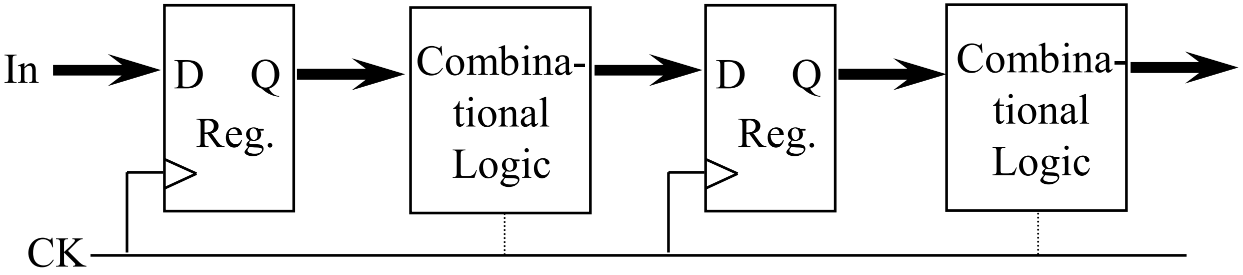
2.4 Clock strategies

2.4.1 Clock in logic systems

Finite State Machine (FSM):
Control Unit



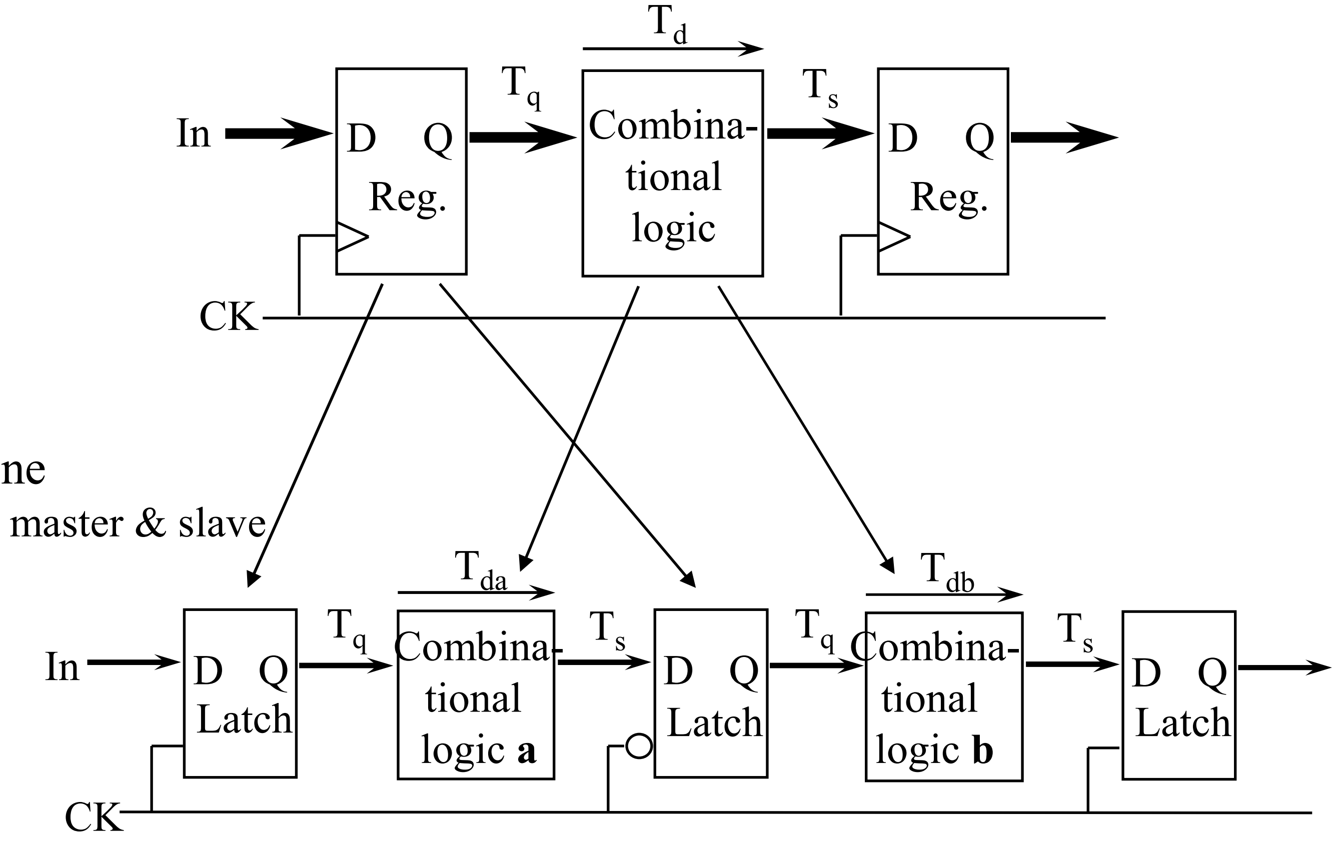
Pipeline System:
Datapath



Timing

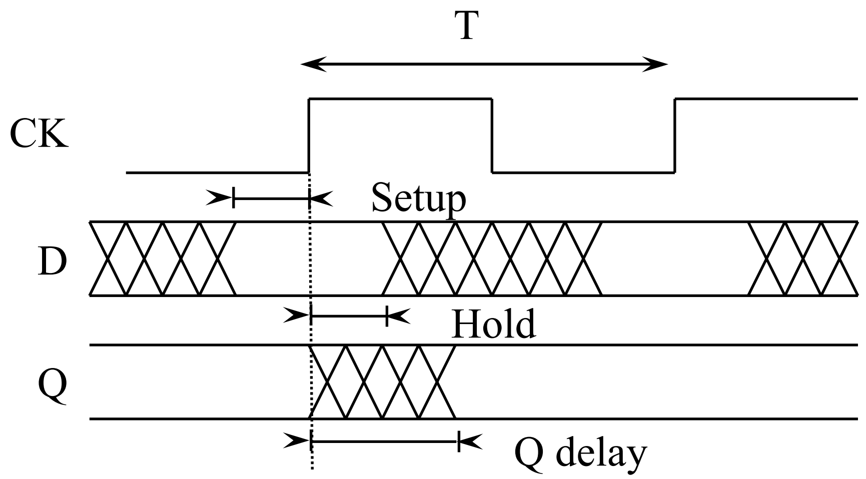
a) Register-based pipeline

b) Latch-based pipeline
Logic embedded within master & slave



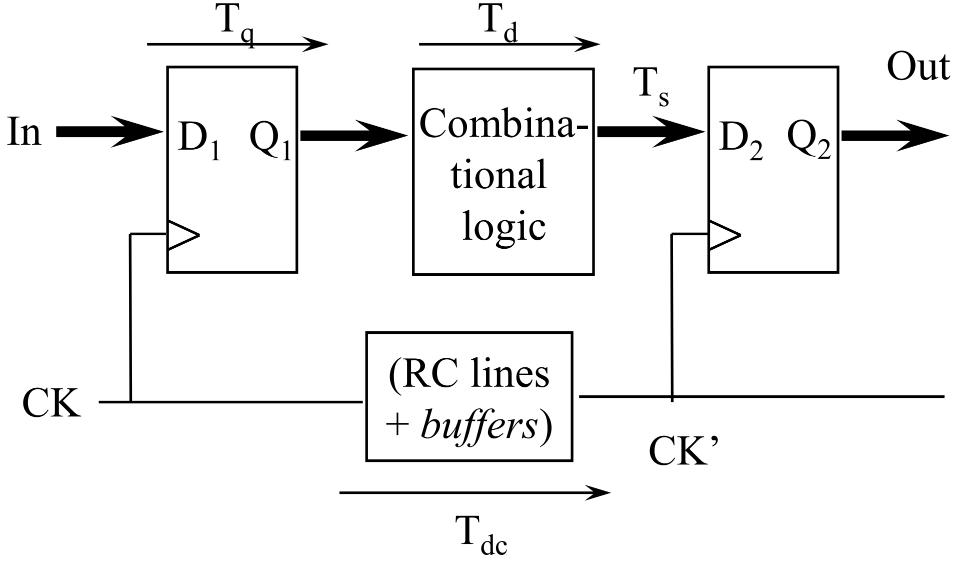
Latch or flip-flop time parameters

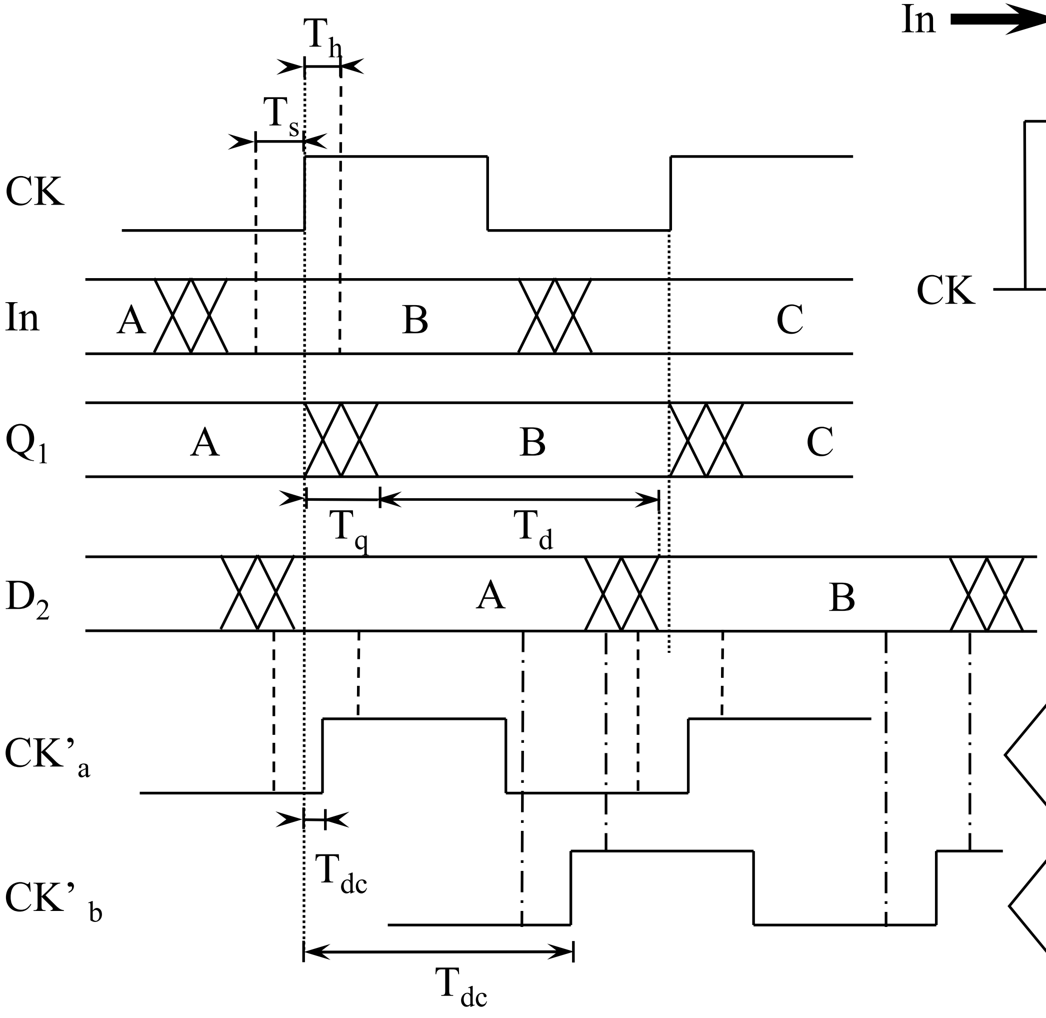
- Setup time
- Hold time
- Propagation delay



2.4 Clock strategies

Setup **and** ***hold*** **time requirements**





$$\boxed{T \geq T_q + T_d + T_s - T_{dc}}$$

a *setup* time violation (Fast clock / slow logic)

b *hold* time violation (Slow clock / fast logic)

$$\boxed{T_{dc} \leq T_d + T_h}$$

Clock strategy classes:

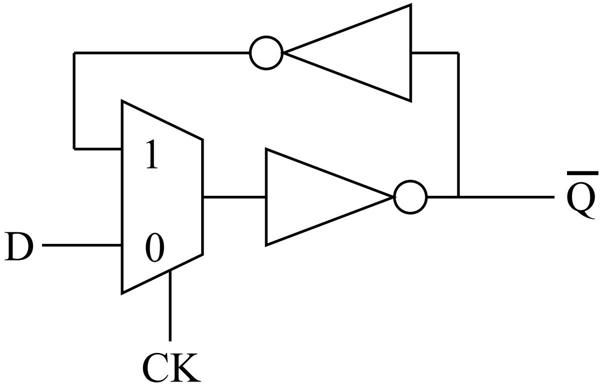
- **Single-phase**
- **2-phase**
- **4-phase**

2.4.2. Single-phase clock

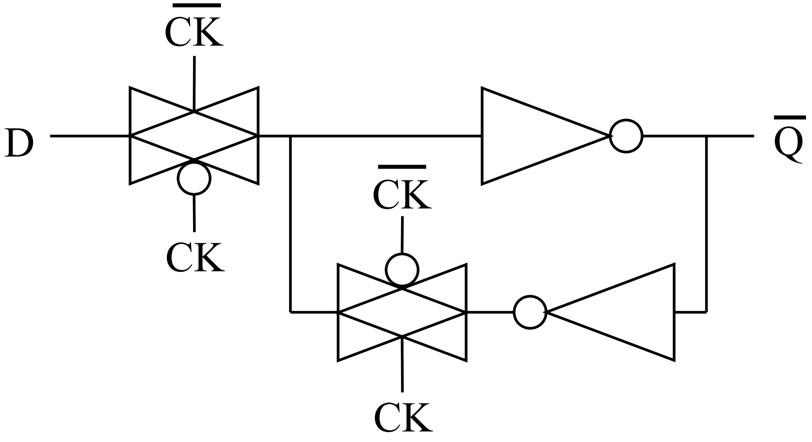
2.4.2.1 Memory structures

Pseudo-static l*atch* (level active)

Functional diagram

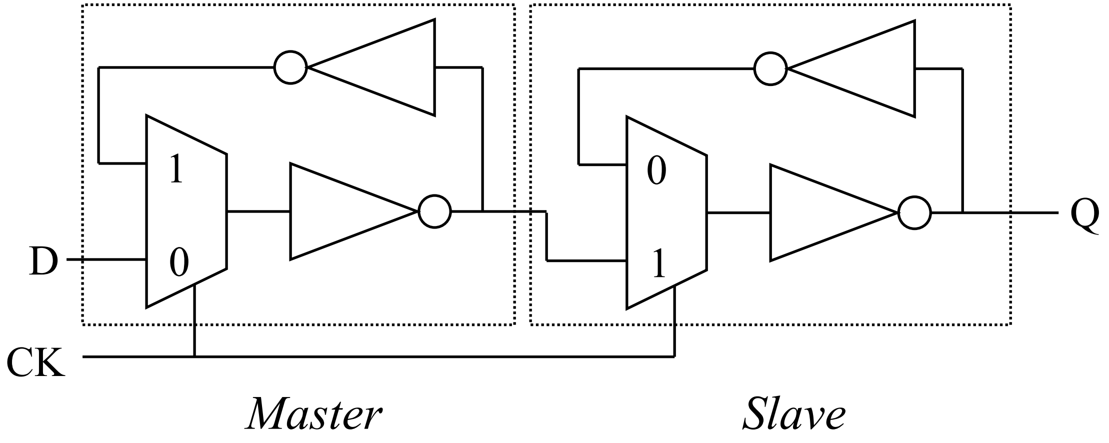


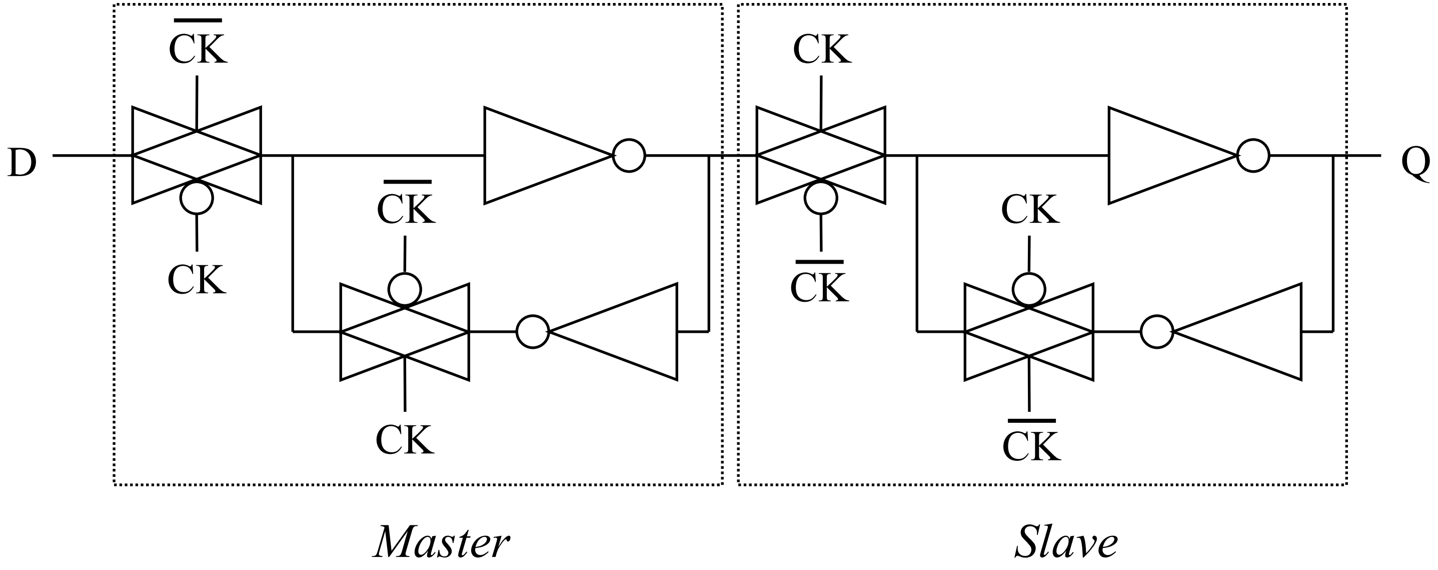
Transmission gate implementation



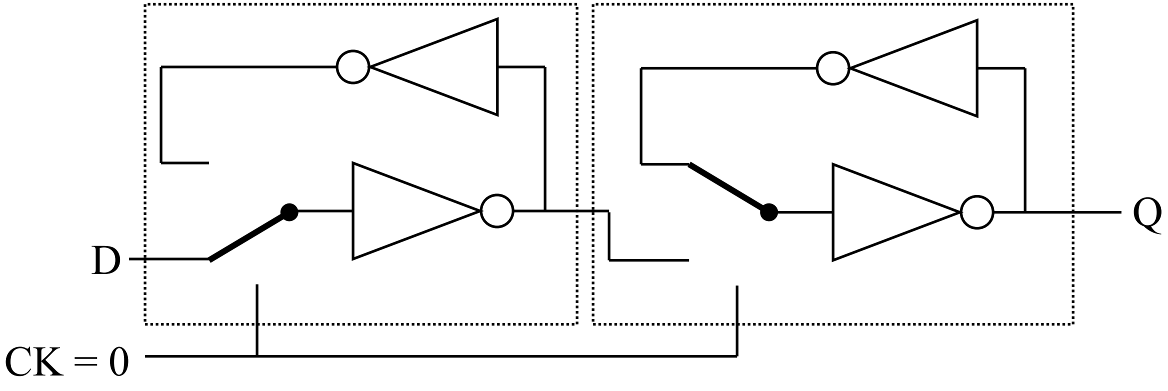
2.4 Clock strategies

D-*flipflop* (edge active)

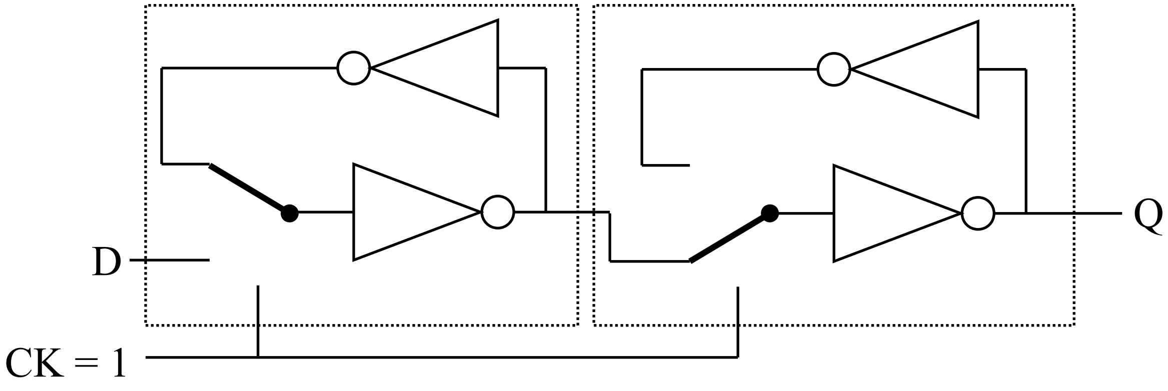




D-*flipflop operation*

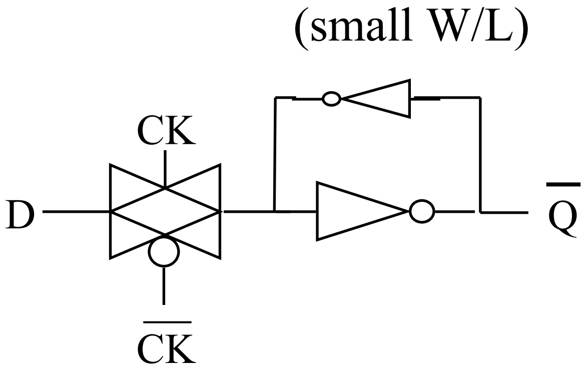


Sample



Hold

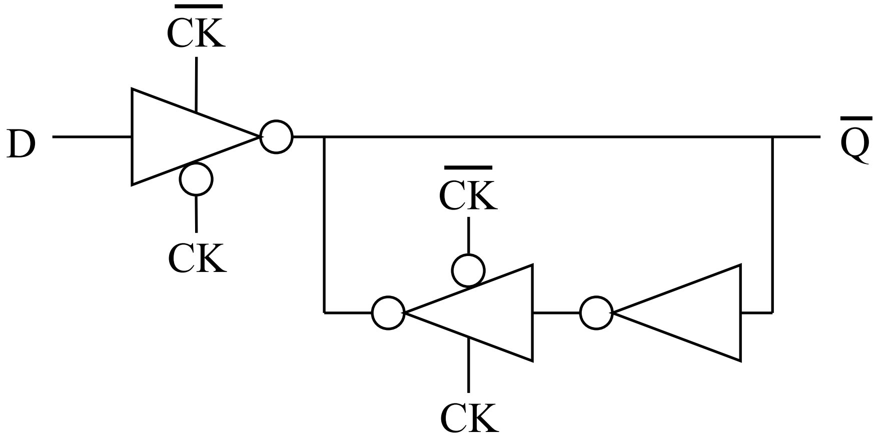
Simplified D-latch



Drawbacks

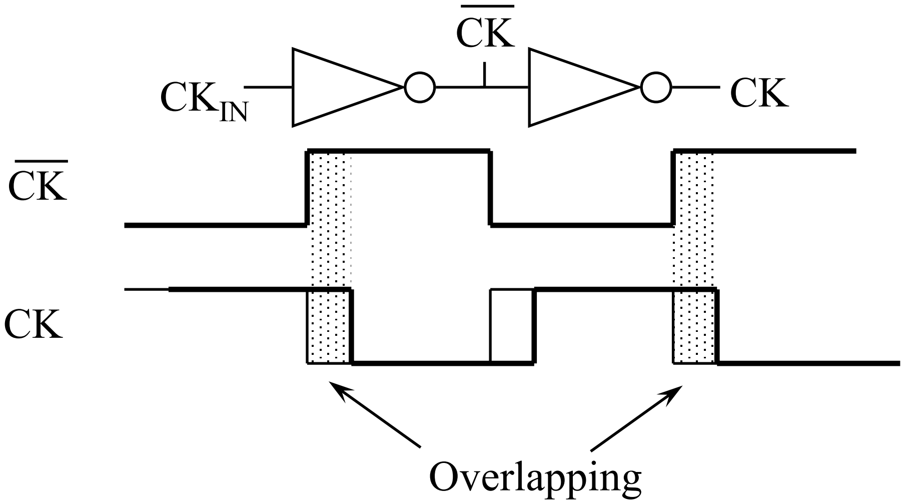
- Transition power
- Non-minimal transistors
- Ratioed logic

Latch implementation using tri-state inverters (C^{2}MOS)

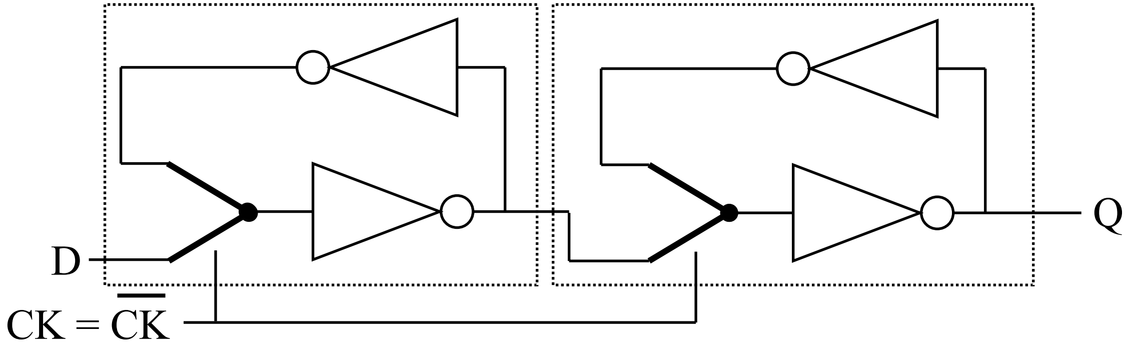


2.4.2.2 Single-phase clock limitations

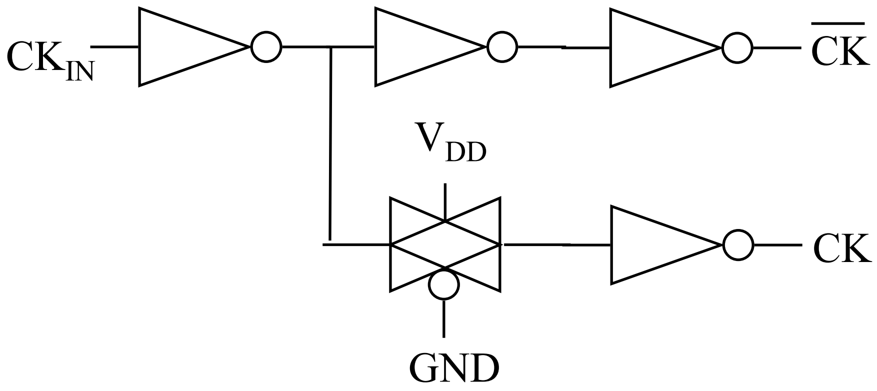
CK and $\overline{CK}$ overlapping



For example, in D-flipflop:



Delay equalizing:

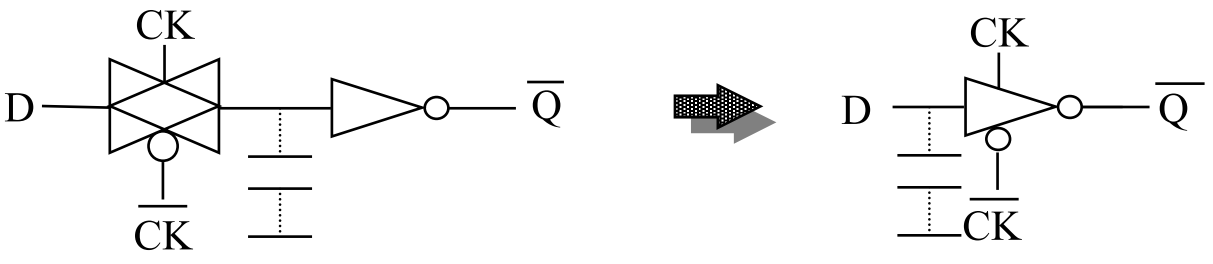


Dimension tuning by simulation.
Routing delay also has to be tuned!

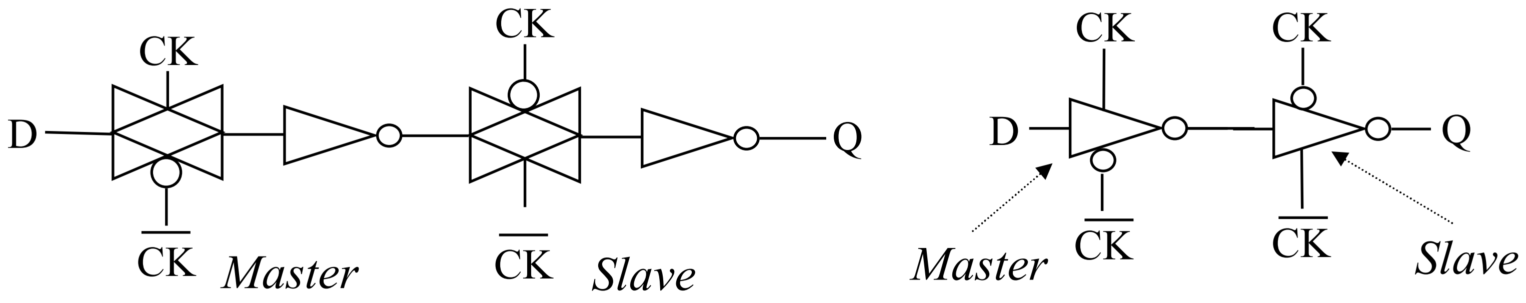
2.4.2.3 Dynamic flipflops

- Feedback can be suppressed for dynamic operation
- MOS gate parasitic capacitance stores the state

Dynamic latch :



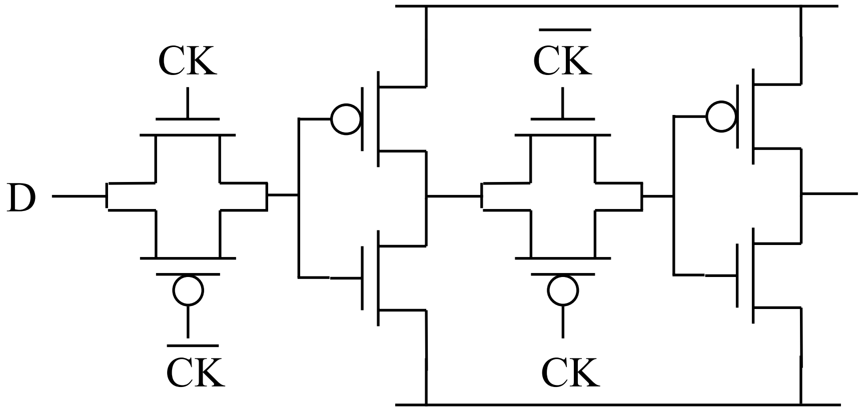
Dynamic flipflop :



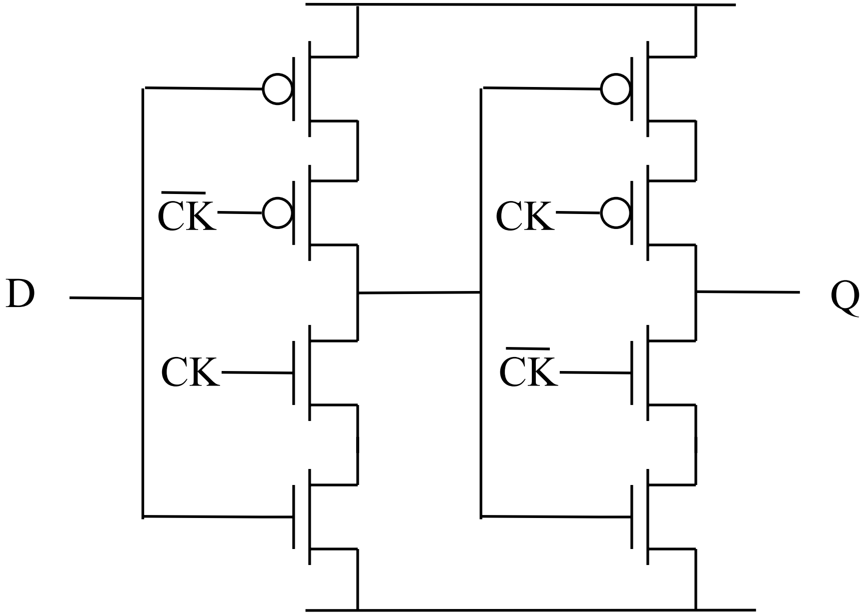
- Very small $\overline{D}$ ➔ Q delay (especially in shift registers)

CK and $\overline{\text{CK}}$ overlapping (CK = $\overline{\text{CK}}$)

a) Transmission-gate flipflop



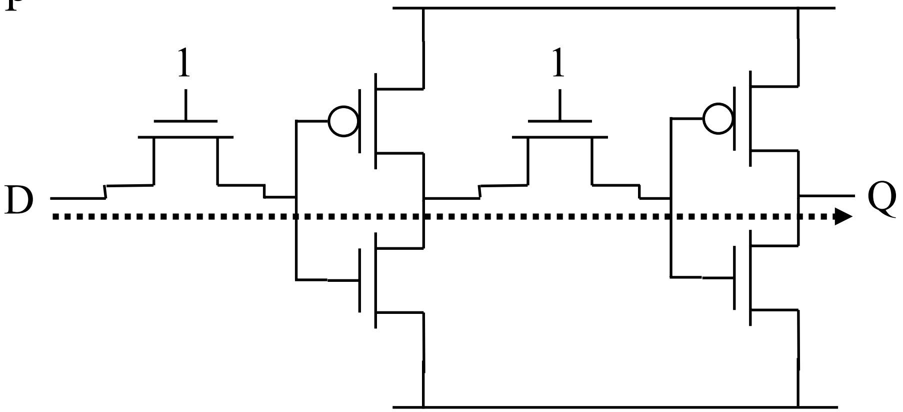
b) C^{2}MOS flipflop



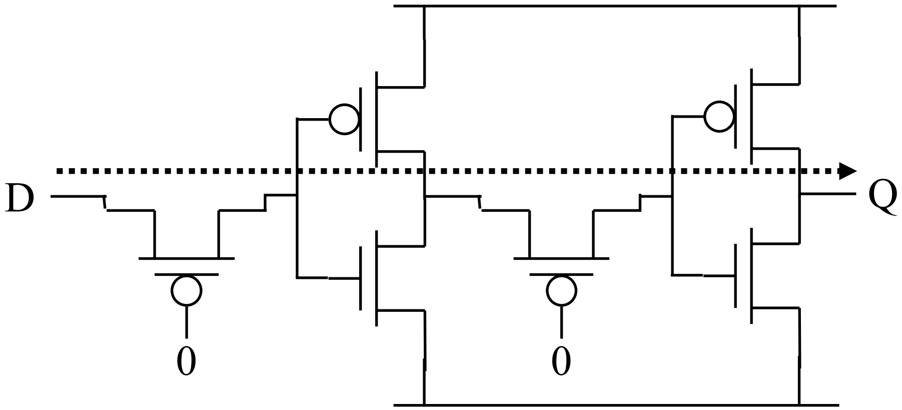
2.4 Clock strategies

a) Transmission-gate flipflop

CK = 1

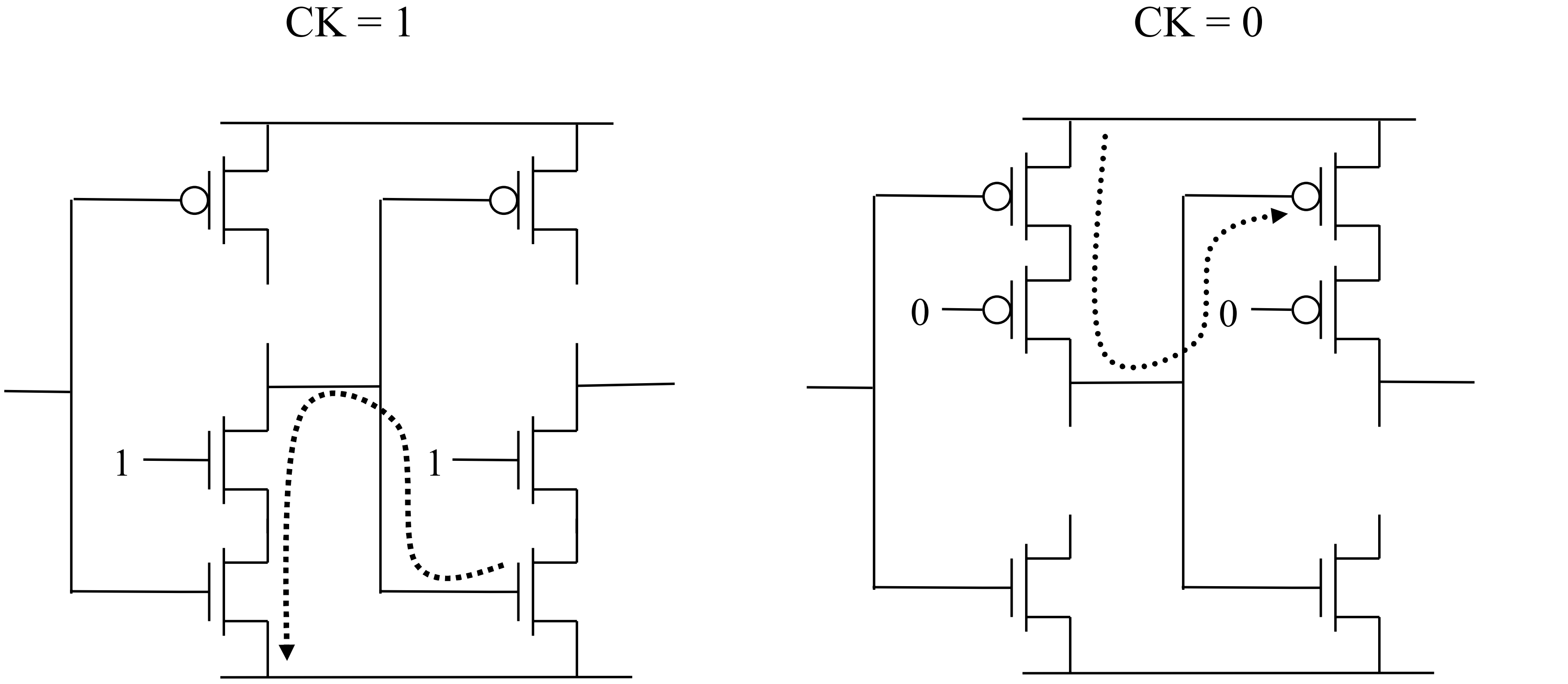


CK = 0

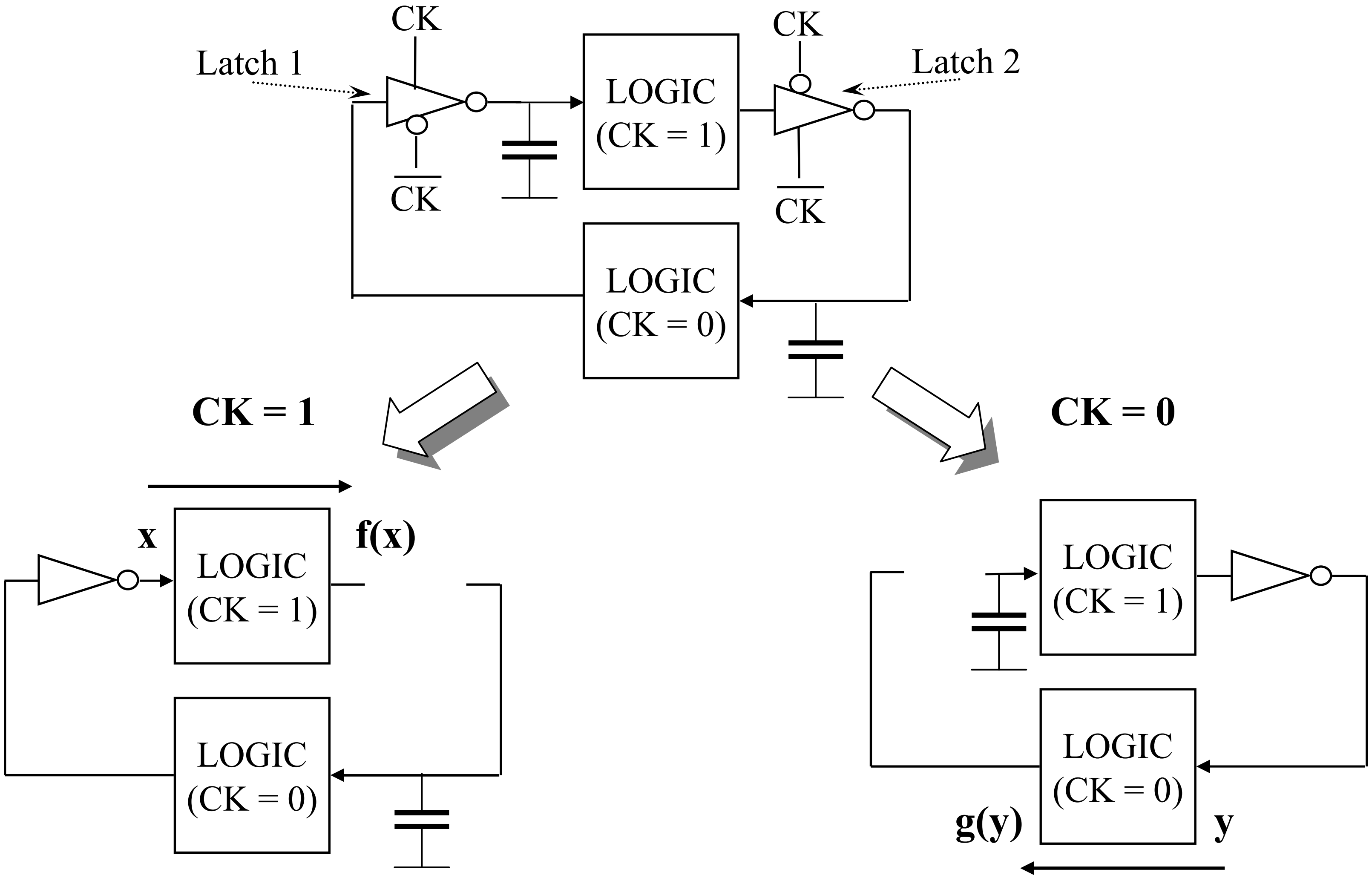


2.4 Clock strategies

b) C^{2}MOS flipflop

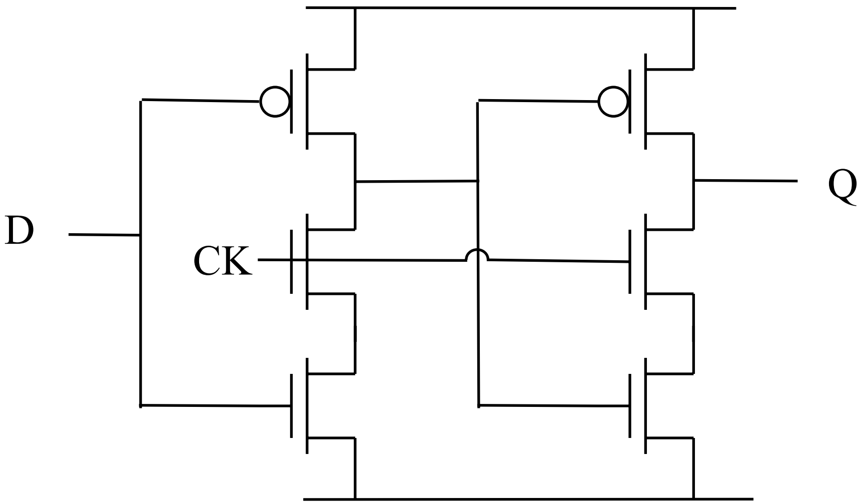


2.4.2.4 Single-phase dynamic timing

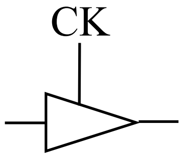


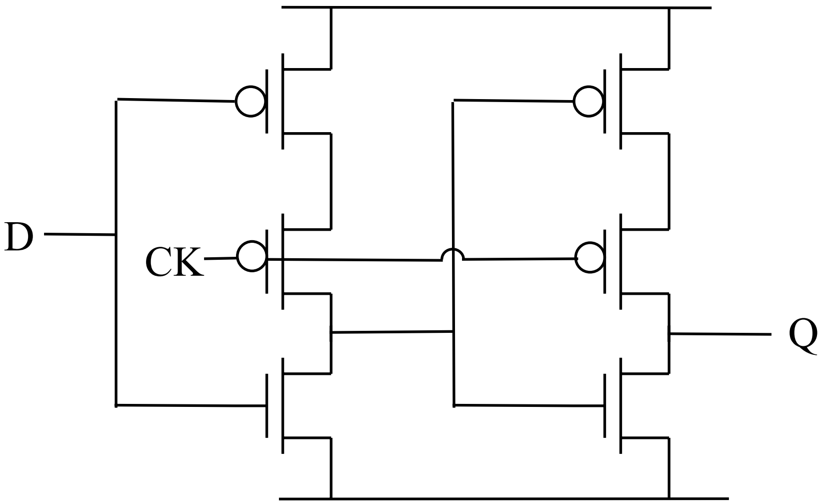
2.4 Clock strategies

TSPC (*True Single Phase Clock*) logic

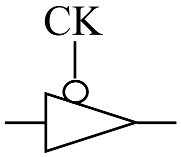


N Latch



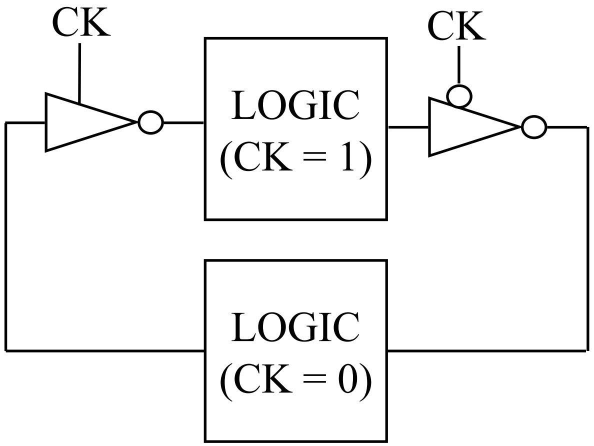


P Latch



2.4.2.5 Example: ALPHA microprocessor (DEC)

- 64-bit RISC processor
- High-speed
- Dynamic logic without inverted CK



Correct operation for fast clock edges:

$\mathbf{t_r, t_f \leq 1.0\ ns}$

Maximum delay time to refresh dynamic logic.
Example:

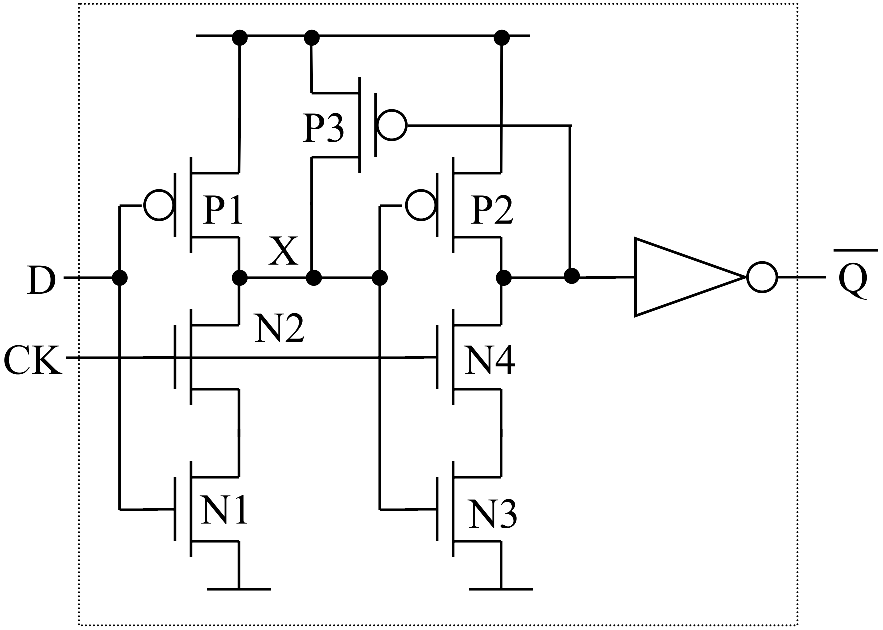
- Drain and source reverse PN-junction leakage current < 1 nA (worst case)
- Gate capacitance < 20 fF
- Discharge voltage: 5 V

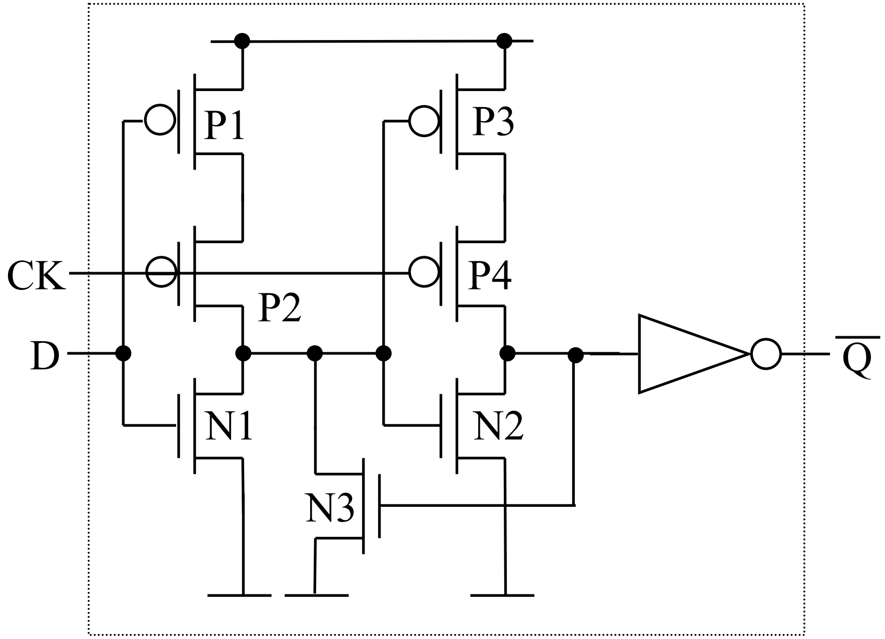
$$\Delta t = C\frac{(\Delta V)}{(\Delta i)} = 100\mu s$$

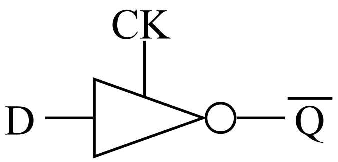
It is discouraged to leave unrefreshed dynamic nodes, since intermediate voltage values produce important currents in gates whose inputs are connected to that nodes.

2.4 Clock strategies

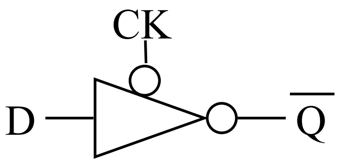
Latchs:







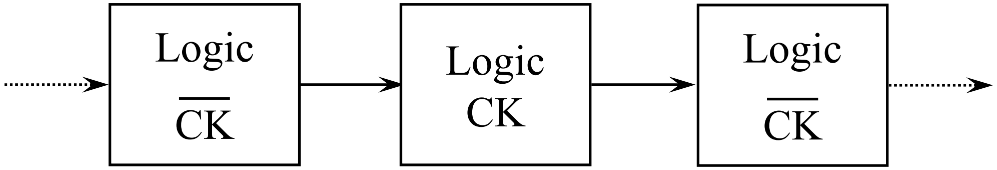
Non-inverting phase



Inverting phase

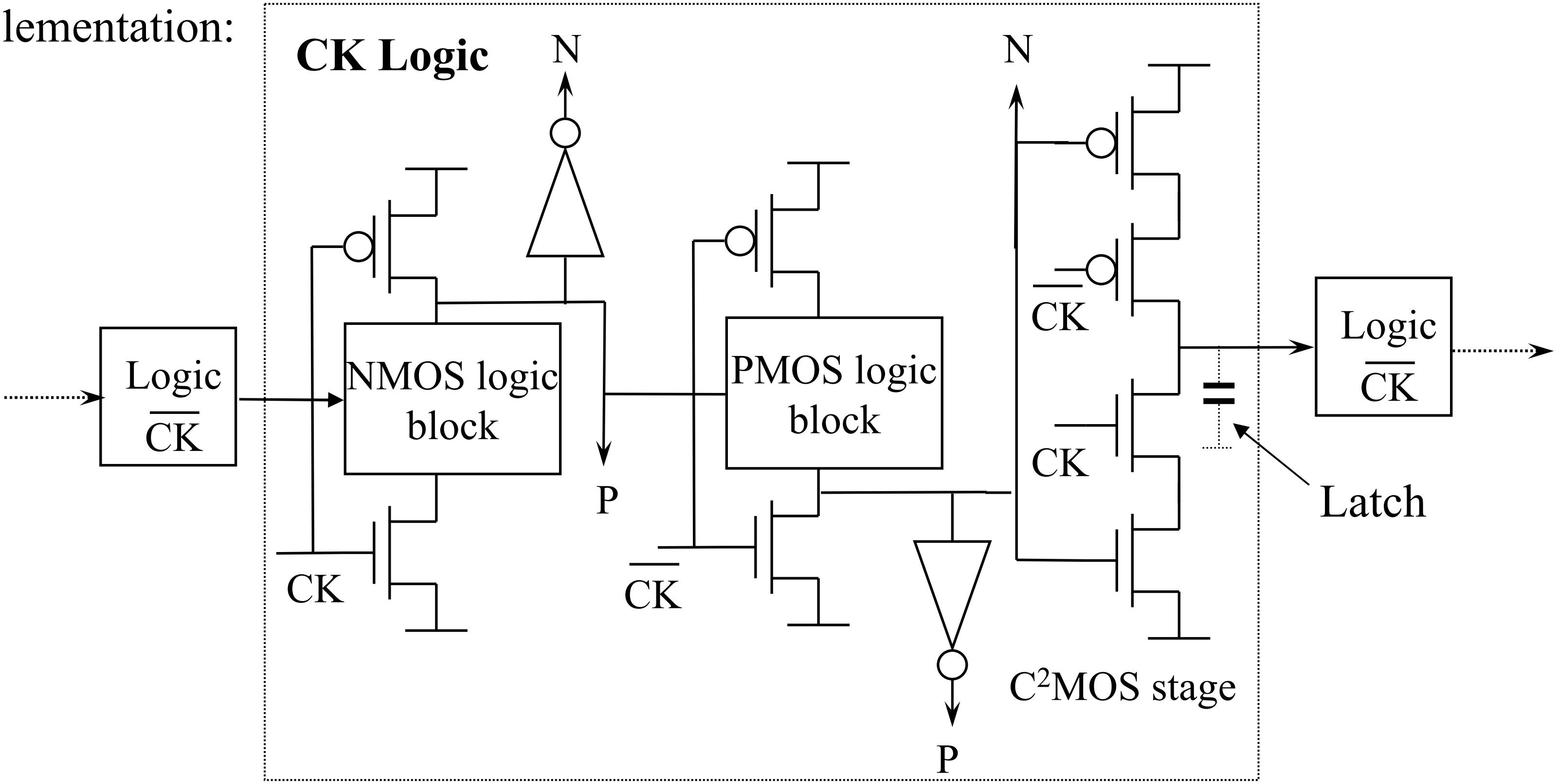
2.4.2.6 Dynamic logic with single-phase clock

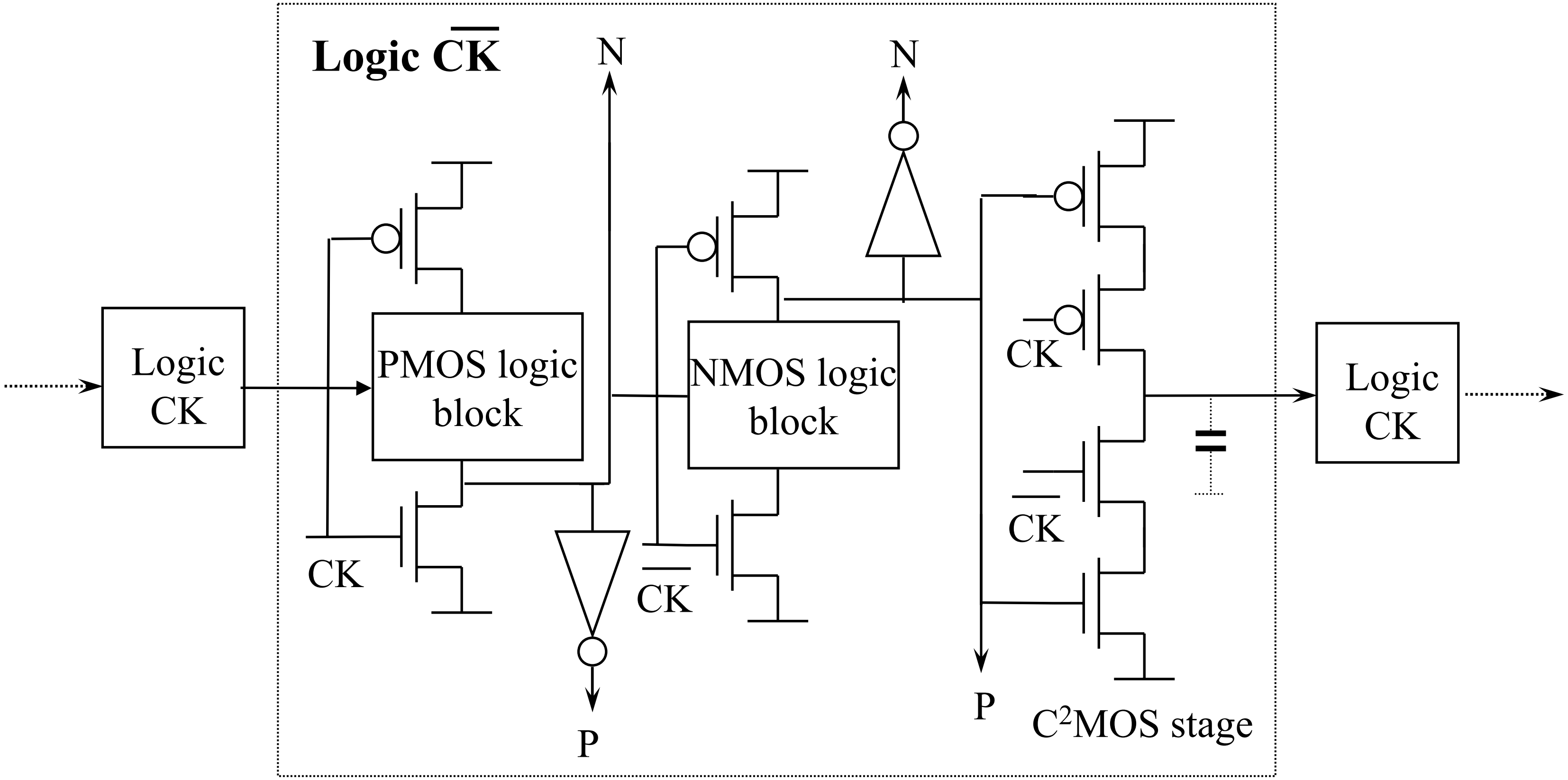
NP CMOS (Zipper) logic example.
Pipeline:
Precharge and evaluation blocks are interleaved.



CK = 0	Evaluation	Precharge	Evaluation
CK = 1	Precharge	Evaluation	Precharge

Implementation:





Problems:

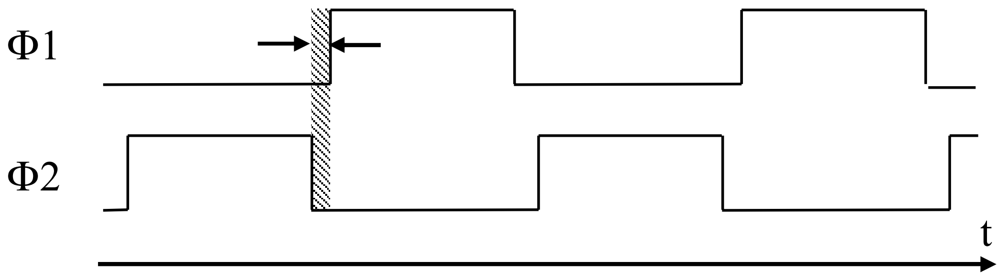
- Clock skew
- Combination with static logic: Any glitch has to be prevented since it could accidentally discharge a precharged node.

2.4.3 2-phase clock

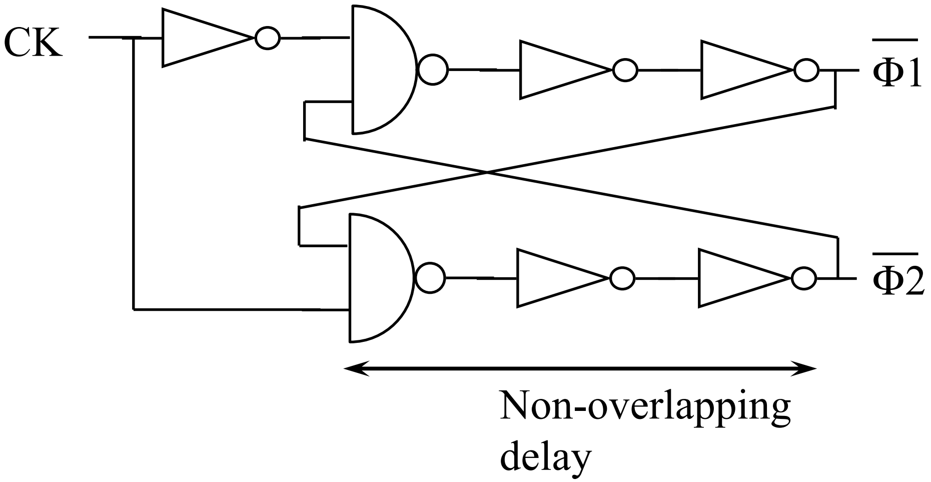
Eliminates the single phase clock overlapping problem.
Still overlapping could appear due to:

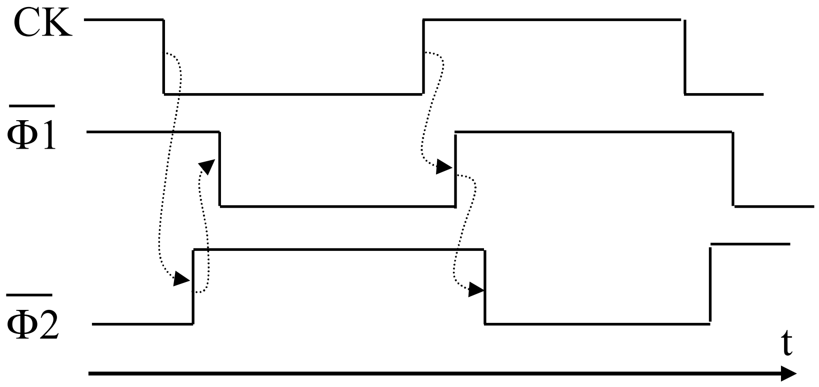
- Skew
- Slow rising/falling time

$$\Phi 1(t) \cdot \Phi 2(t) = 0$$



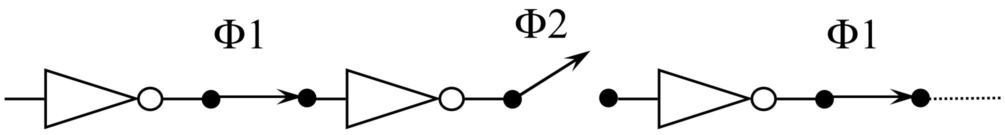
2-phase generation from CK signal:



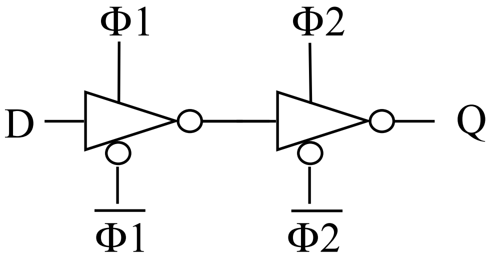


2.4.3.1 Memory structures

Single-phase structures can be applied:



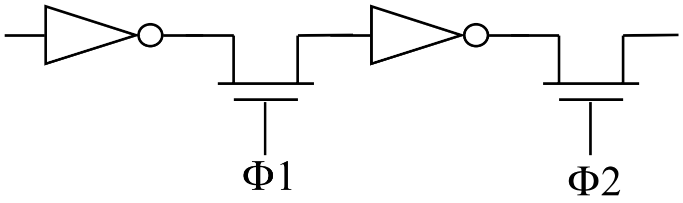
Disadvantage: four clock line distribution.



Solution: Use only NMOS pass transistors

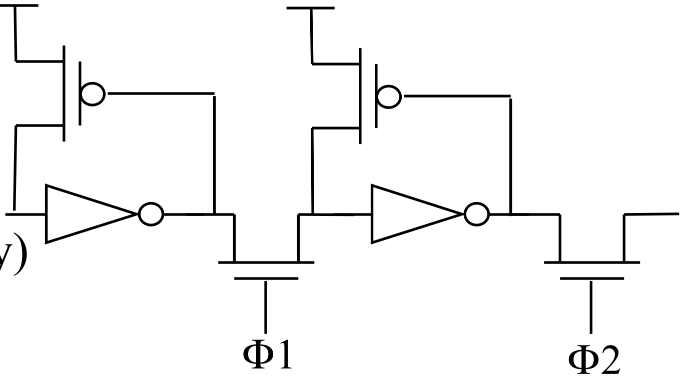
Problems:

- High (1) Level to V_{DD} - V_{TN}
- Noise margin reduction
- Possible static power consumption



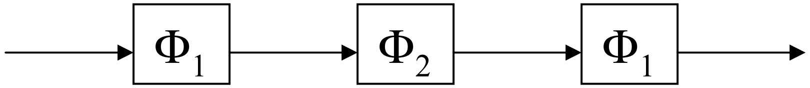
Improvements:

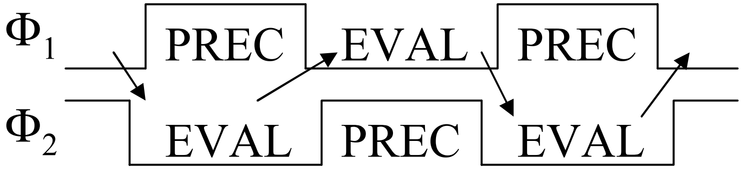
- To increase switch V_{ON}
- Charge bombs
- To add a weak PMOS (at the cost of increasing delay)



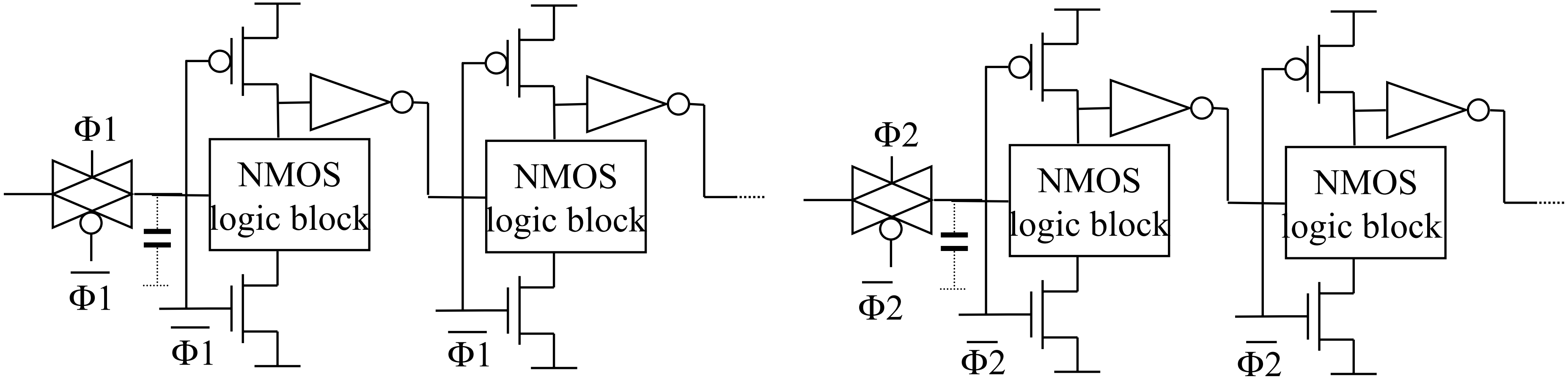
2.4.3.2 Two-phase dynamic logic structures

Example: domino logic :





- Phase 1 stage
- Phase 2 stage

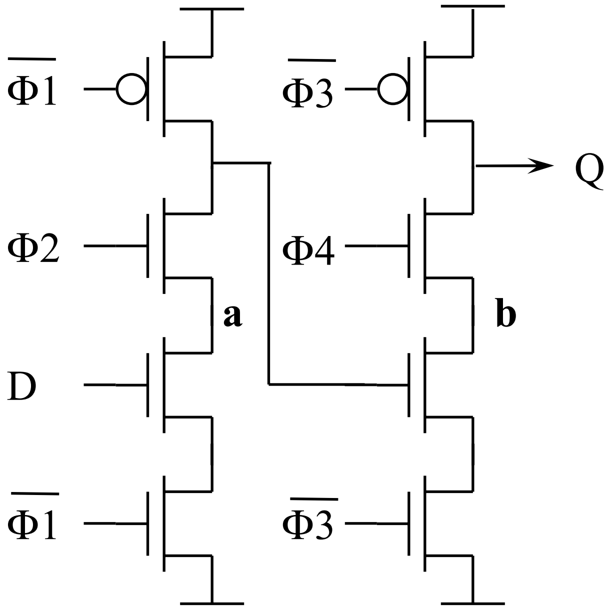


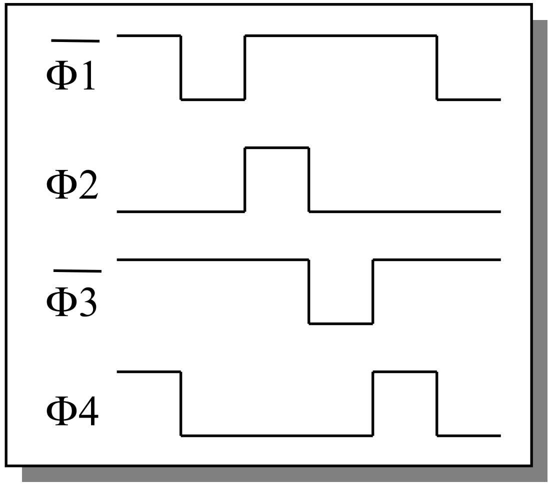
Only transmission gates are required for latches

2.4.4 4-phase clock

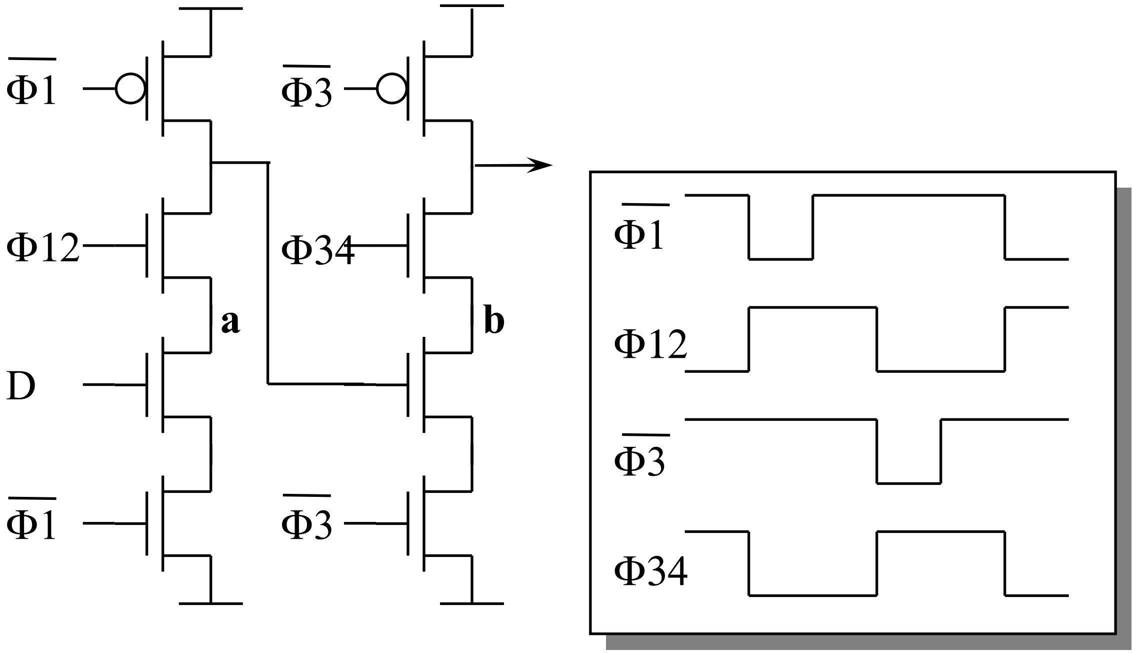
Advantages: Simple and robust
Disadvantage: Clock signals distribution

2.4.4.1 4-phase memory structures

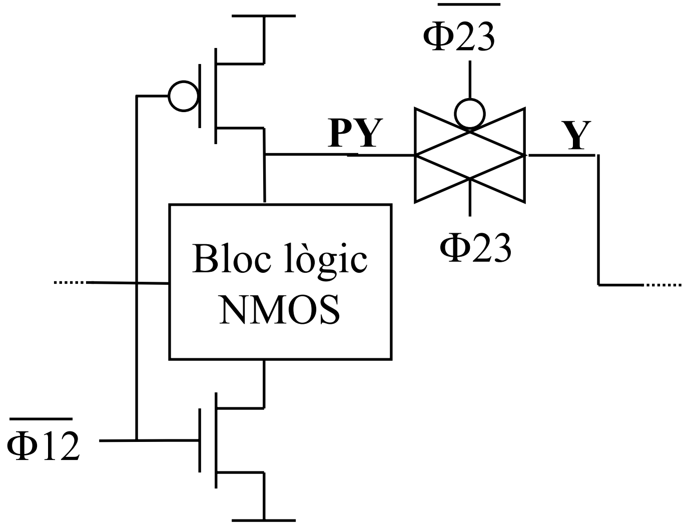


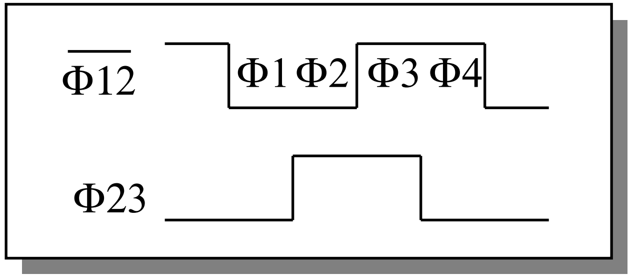


Charge redistribution elimination (**a** and **b** nodes):



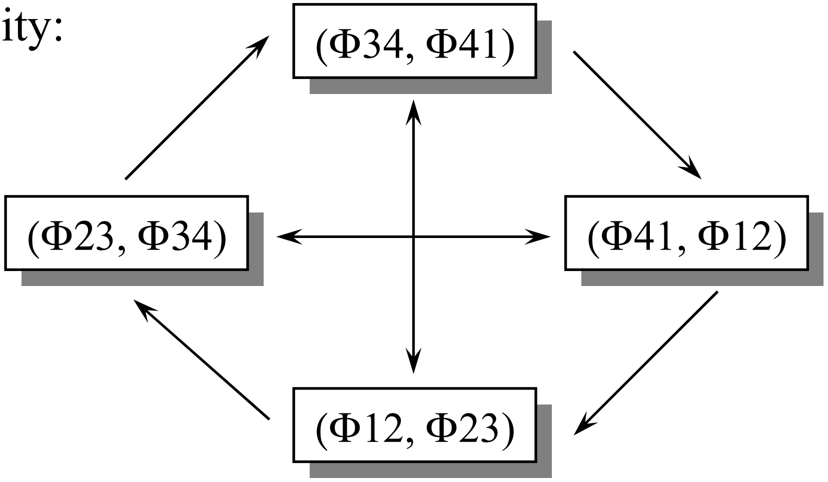
2.4.4.2 Logic structures





- Φ1: **PY** precharge
- Φ2: **Y** precharge
- Φ3: **Y** evaluation
- Φ4: **Y** hold

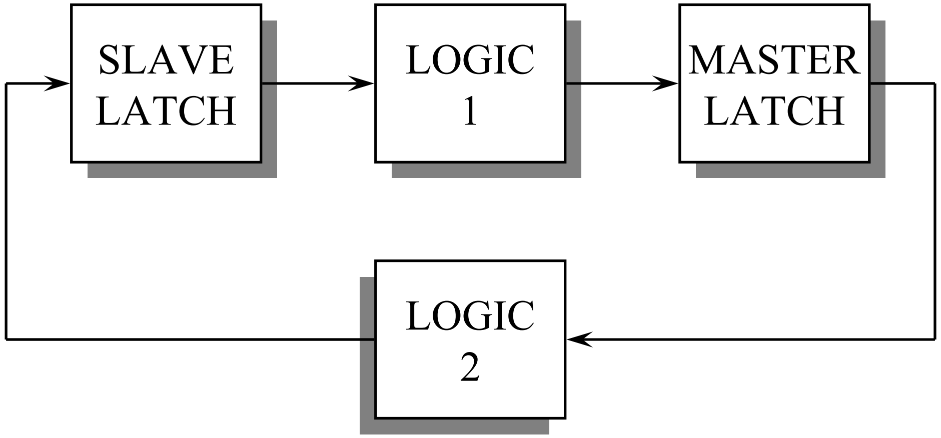
Phase connection compatibility:



4-phase clock became popular at the VLSI beginning.

- In some cases the required lines are the same as in 2-phase.
- Ratioless circuits ⇒ Very regular layout
- Synchronization strategy:

Speed inefficient (some phases are not used to their limit)



2.4.5 Recommended clock strategy

- Static logic with single-phase clock: Low-cost and automated design (in many cases it is the only option with automated synthesis tools).
- Dynamic logic with 2-phase clock: Increase speed. Robust.
- Dynamic logic with single-phase clock : Maximum performance. Clock is critical.
- Dynamic Logic with 4-phase clock: Rarely used.