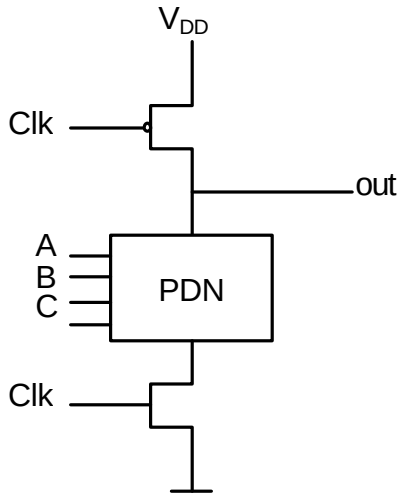
Dynamic CMOS design

uses N+2 transistors bit no static power



Clk=0→ out=V_{DD} ← precharge state

as PMOS=on. NMOS=off

Clk=1 → out → f(A,B,C,D) ← evaluation state

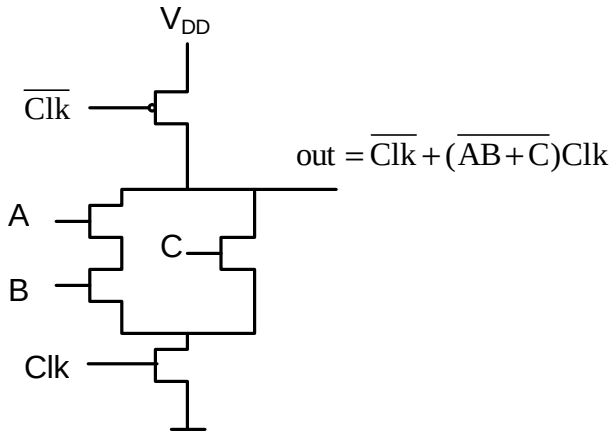
as PMOS=off. NMOS =on

During Clk=1 if out : V_{DD} → 0

then inputs change out cannot be charged to V_{DD}

and we have to wait till next cycle.

Example



Dynamic logic is faster than complementary logic due to smaller # transistors

→ logical effort is less 2-in NOR : $g_{dyn} = 2/3$, $g_{stat} = 5/3$

Disadvantages → during eval. V_{out} cannot change from 0 → V_{DD}

low NML since $V_M - V_{tn}$

higher switching activity

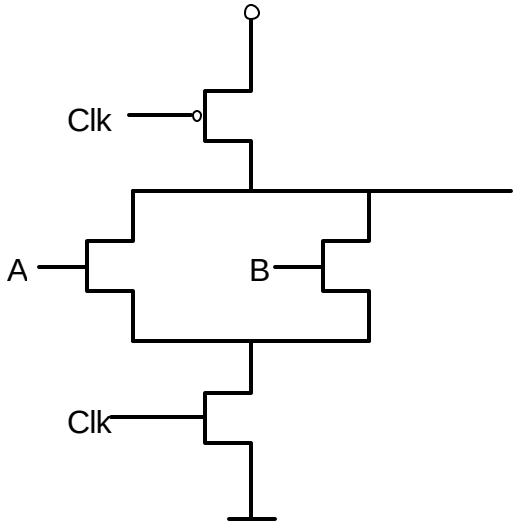
also $a_{0\to 1} = P_0$ not $P_0(1-P_0)$

as the output goes to 1 at each Clk cycles

trsnsition probability for N-input gate (uniform dist.)

$$a_{0\to 1} = \frac{N_0}{2^n}$$

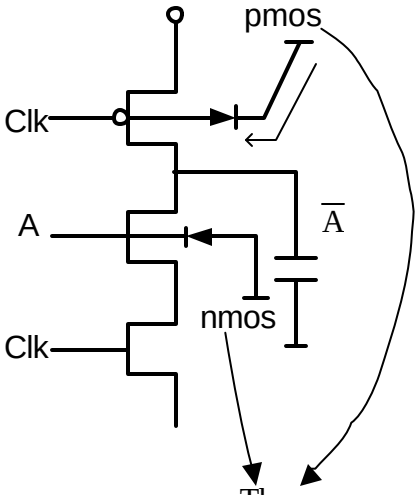
Dynamic Circuit

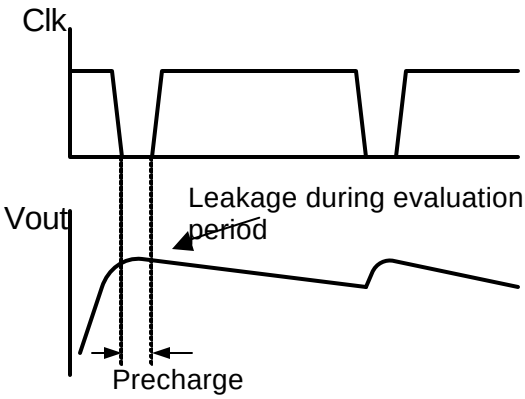


Dynamic NOR

Problems with Dynamic Ckts

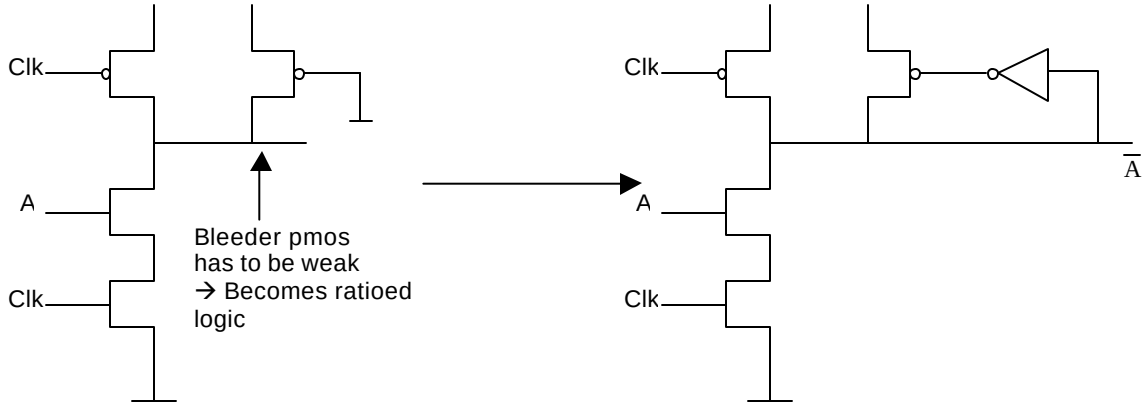
1. Charge leakage





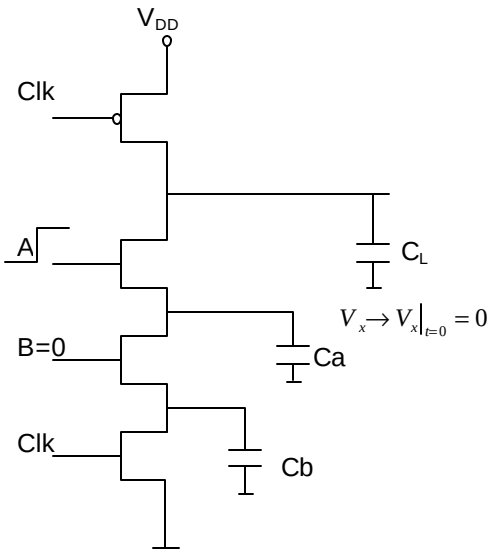
These pmos and nmos have leakage and work against each other. Pmos charges the capacitance with its leakage current and nmos discharges the capacitance.

To solve the problem we can add current bleeder



→ To avoid ratio problem and avoid static power consumption

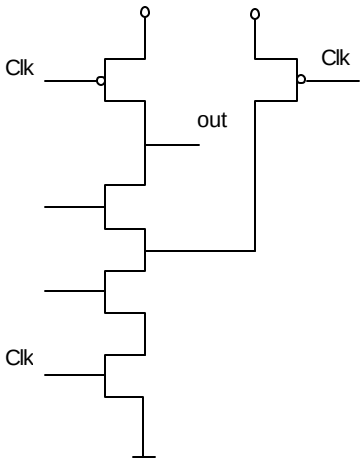
2. Charge sharing



Precharge → C_L is charged to V_{DD}

Evaluation → if B=0, A $_{0\rightarrow 1}$, then the charge on C_L is shared with C_a

To solve the problem we can precharge the internal nodes



3. Capacitive Coupling

Dynamic logic → high impedance output node → coupling can be a problem

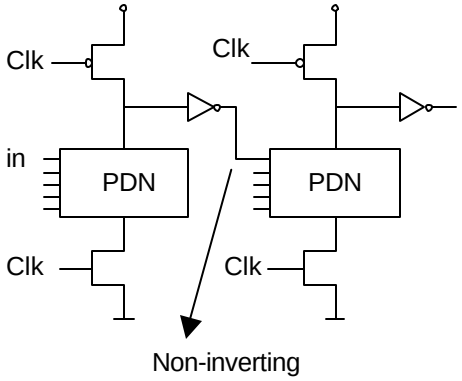
→ to avoid the problem, careful layout design is important

4. Clock feed through

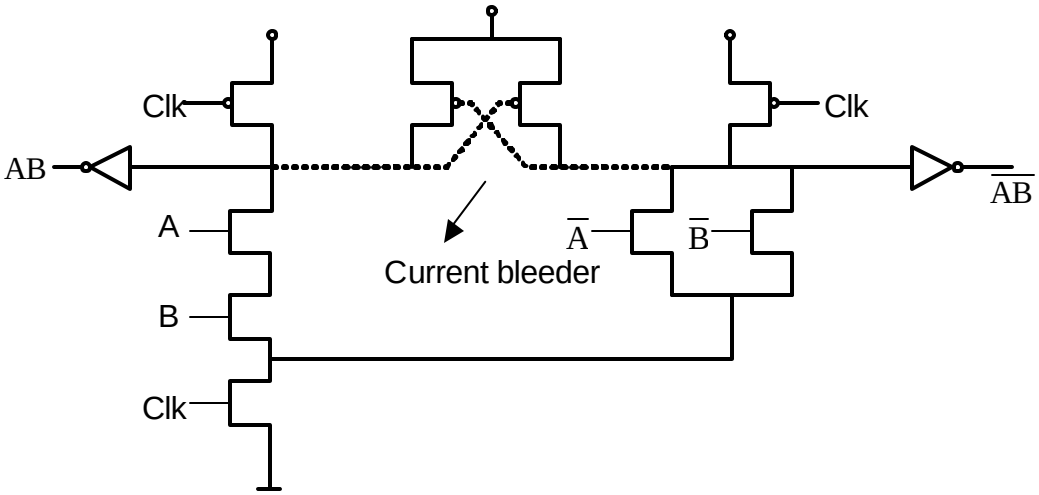
Capacitive coupling for clk signed causes the high impedance output node to have voltages higher than V_{DD} → precharge transistor reverse biased diode → forward biased→ electron injection into substrate

Cascading Dynamic Gates

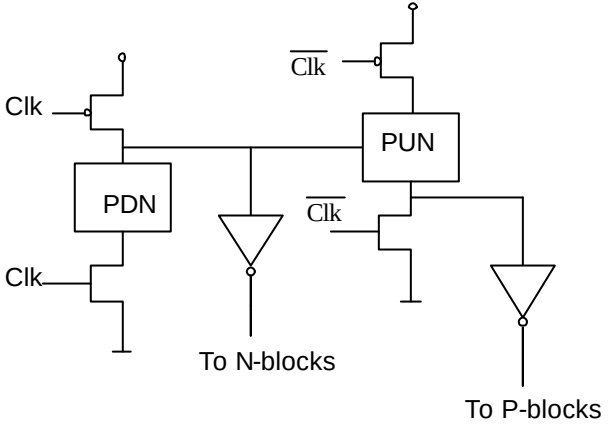
1. Domino logic



Differential Domino logic

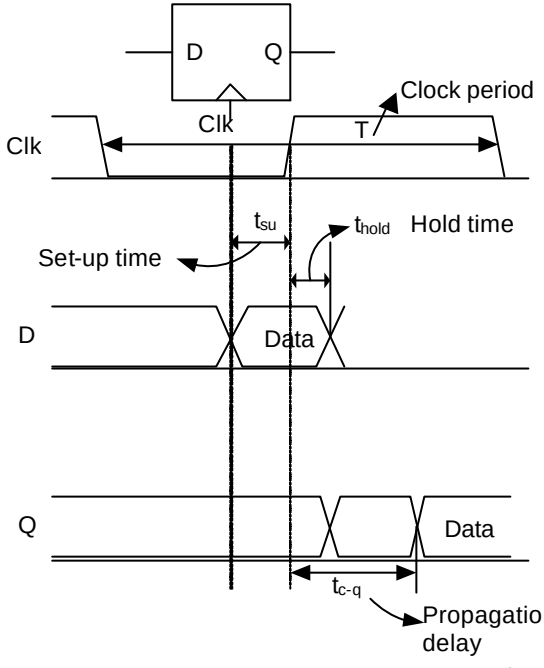


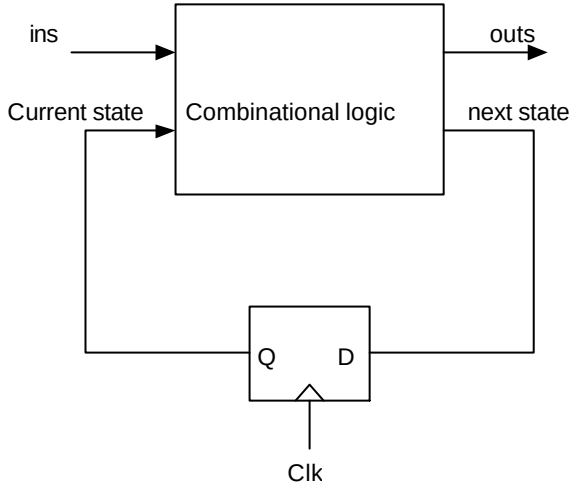
2. np CMOS



Sequential logic

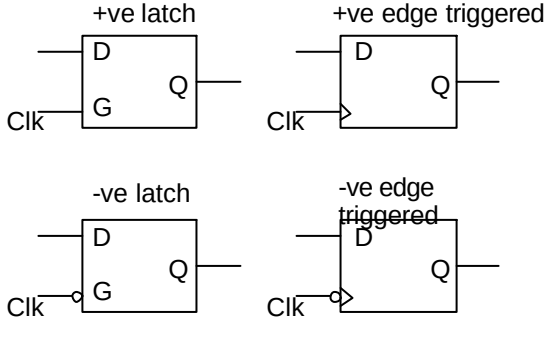
- Defining important timing parameters



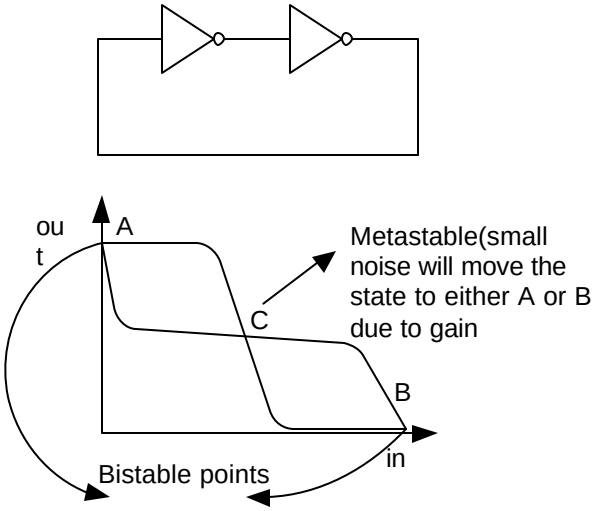


Latch vs. Register

Level sensitive edge sensitive

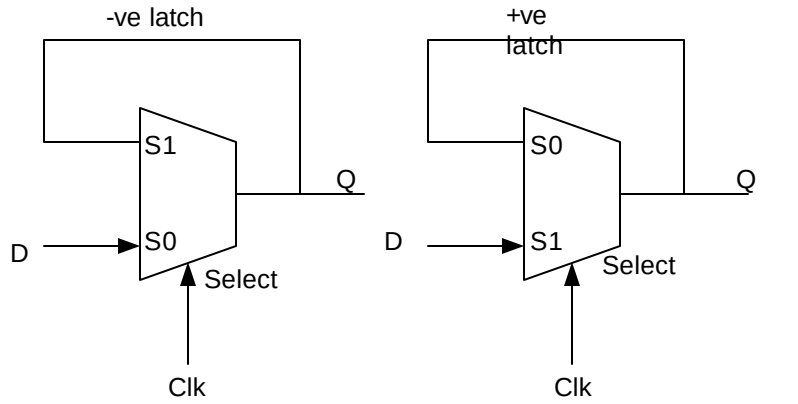


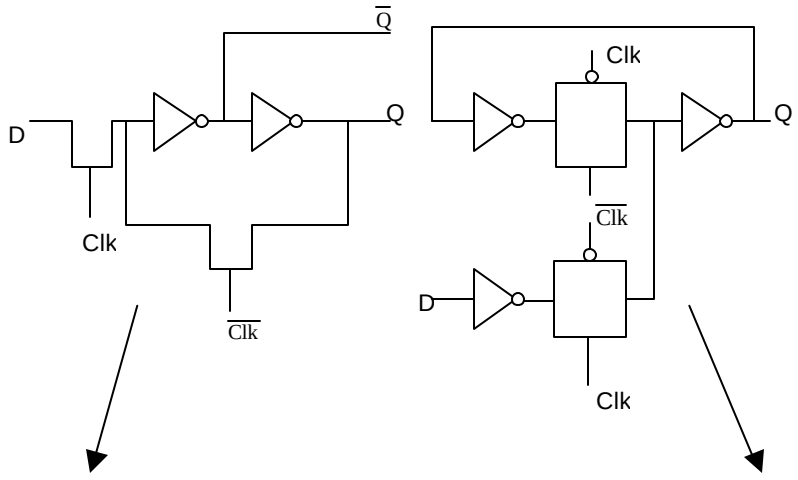
Static latches and registers



→ writing data is possible by either
cutting the feedback loop
or overpowering the feedback loop

Multiplexer-Based latch

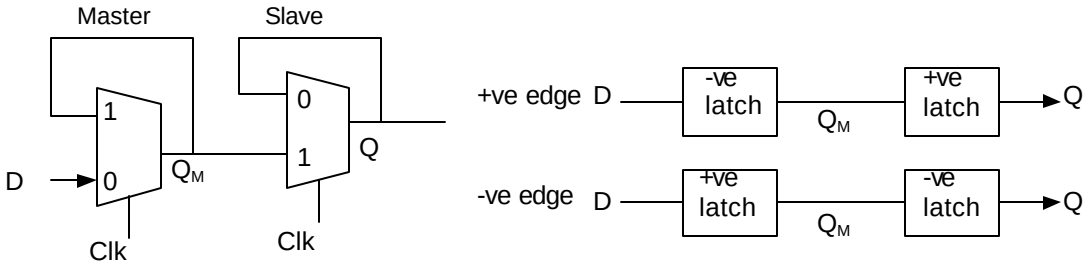




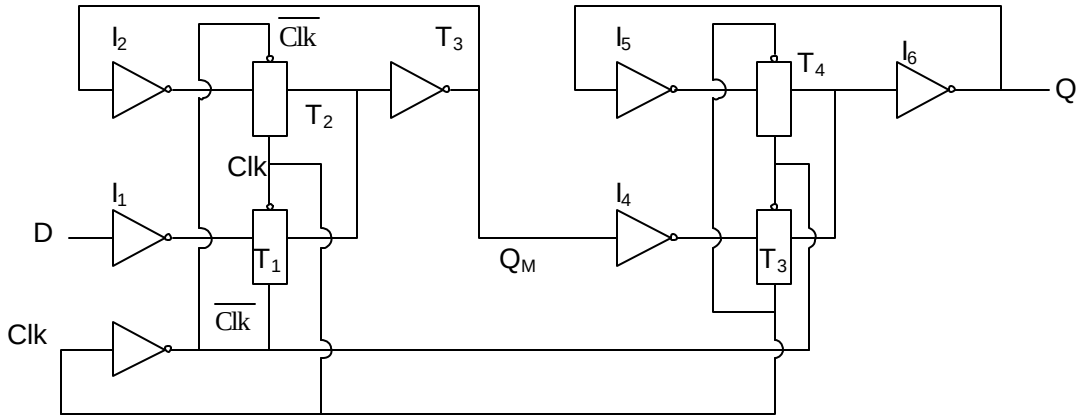
MUX using NMOS pass transistor

using transmission gate

Master-Slave Edge-triggered Register



Timing Properties of MUX-based Master-Slave register



tpd-inv : ← inverter propagation delay

tpd-tx : ← transmission gate prop. Delay

Set-up time: D→ Q_M

t_{su} (I_1,T_1,I_3,I_2) → $3t_{pd\text{-}inv} + t_{pd\text{-}tx}$

Propagation delay : Q_M→Q

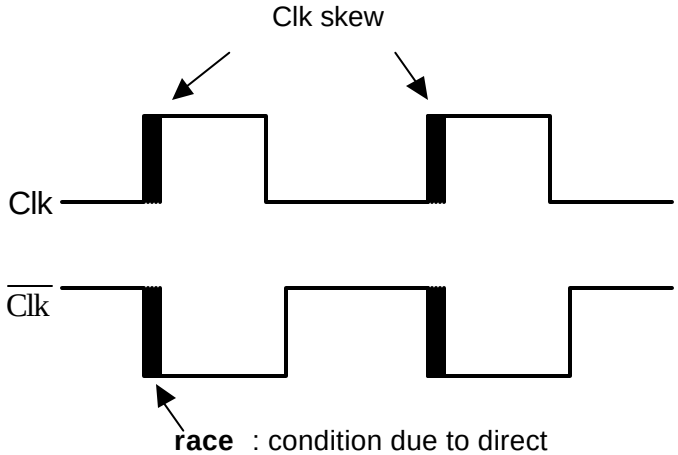
$t_{c\text{-}q}$ (T_3, I_6) → $t_{pd\text{-}inv} + t_{pd\text{-}tx}$

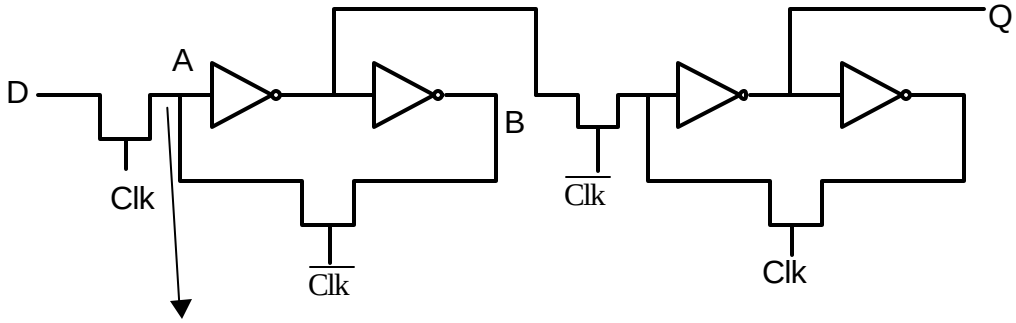
I_4 has the same delay as I_2 which is already taken into account

Holding time

$t_{hold} = 0$ ← D and clk pass through inverter before reading T_1, so their change before Clk on T_1 will not be seen.

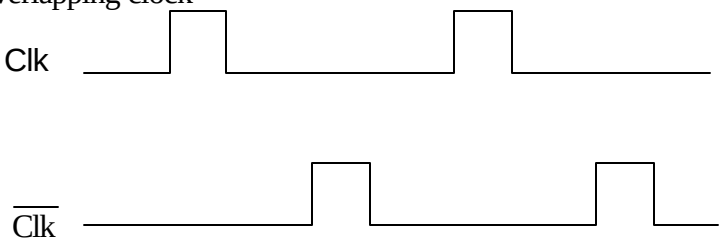
Non-ideal Clk signal



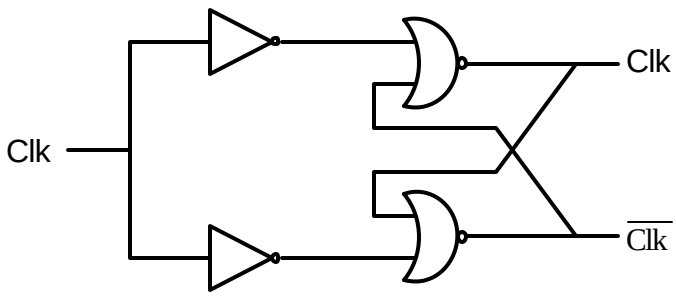


overlap between Clk and $\overline{Clk}$ results in undefined state A=f(D,B)

Non-overlapping clock

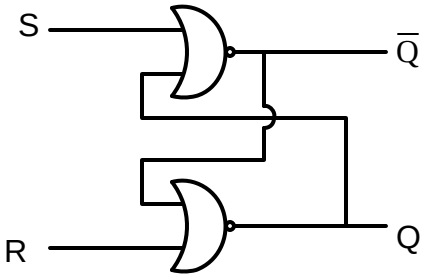


To generate non-overlapping clock



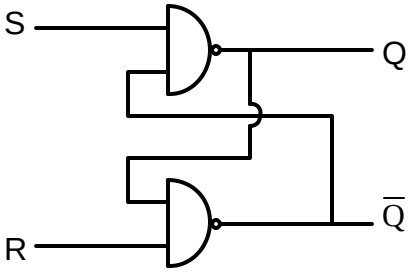
Set-reset latch

Using overpowering the feedback



S	R	Q	
0	0	Q	
1	0	1	
0	1	0	
1	1	0	← forbbiden

NOR-based



S	R	Q	
0	0	0	← forbbiden
0	1	0	
1	0	1	
1	1	Q	

NAND-based

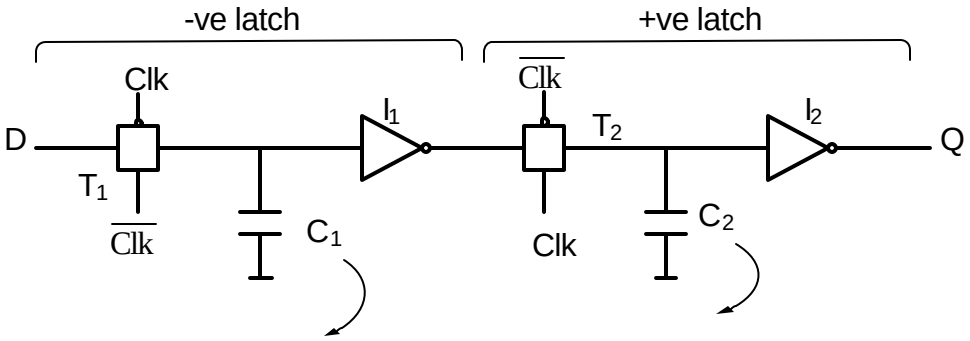
Dynamic latches and registers

Works based on charge storage @ a high impedance node

→ leakage will ruin the data so periodic refreshing is needed.

Dynamic Transmission-Gate Edge-Triggered Register

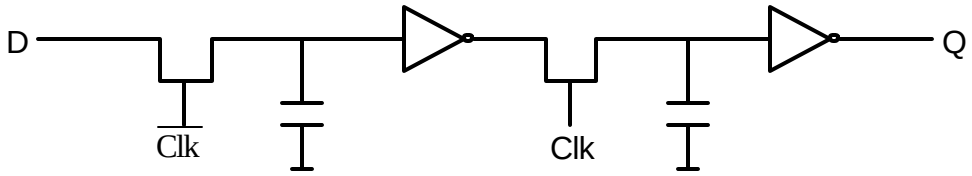
8-transistors version



Capacitance due to Inverter gate cap + junction cap of transmission gate

+ overlap cap of transmission gate

6-transistors version (using NMOS only)



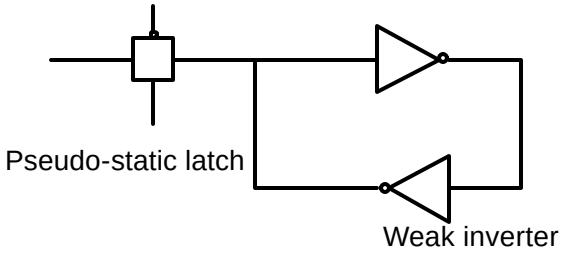
t_{su} : delay of the transmission gate

t_{hold} : almost 0 (transmission gate is turned off on the Clk edge and further input changes are ignored)

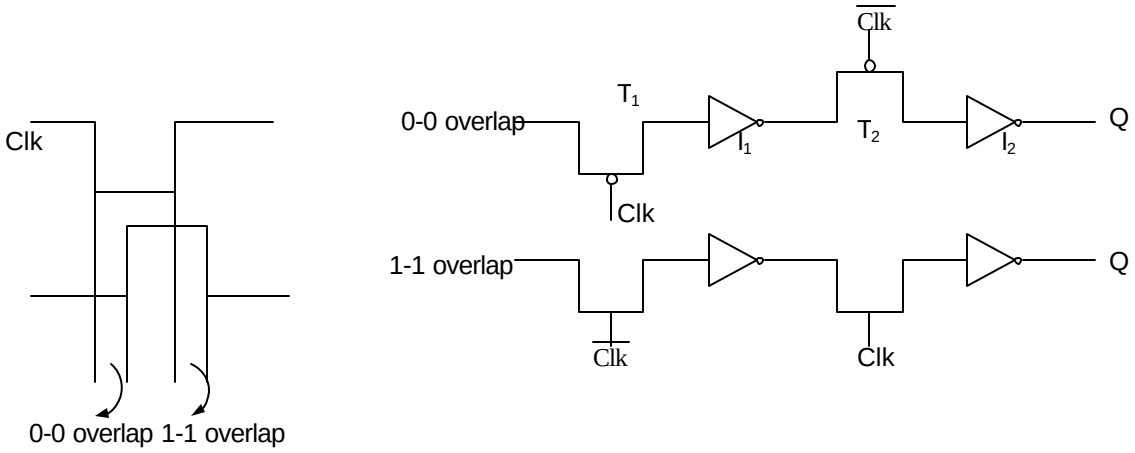
$t_{c\text{-}q}$: $2t_{inverter} + t_{delay\ trans.\ gate\#2}$

Problem : high impedance node → more prone to noise

→ use pseudostatic latch



Problem : Clock overlap → both 0-0 and 1-1 ones lap



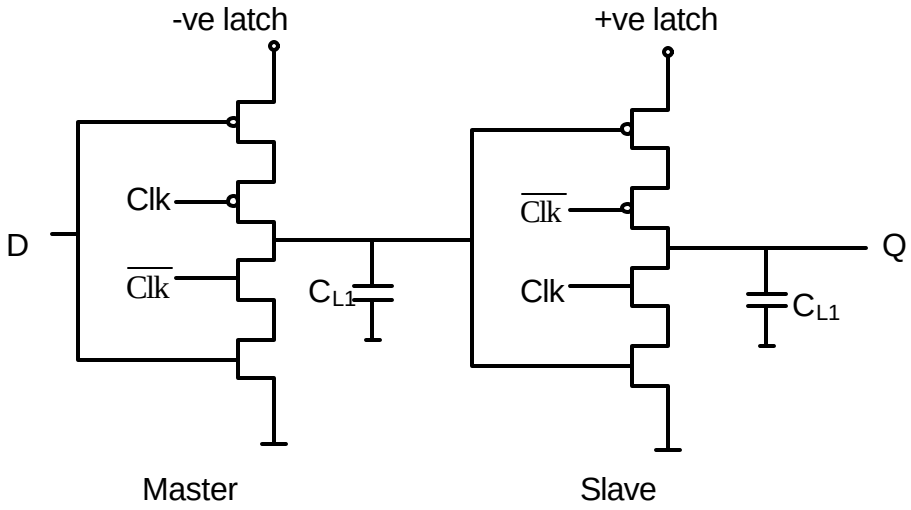
To avoid problem : $t_{overlap\ 0\text{-}0} < t_{T1}+t_{I1}+t_{T2}$

$t_{overlap1\text{-}1} < t_{hold}$

→ make sure data is valid beyond the +ve edge of the clock for more than $t_{overlap1\text{-}1}$

C^2 MOS

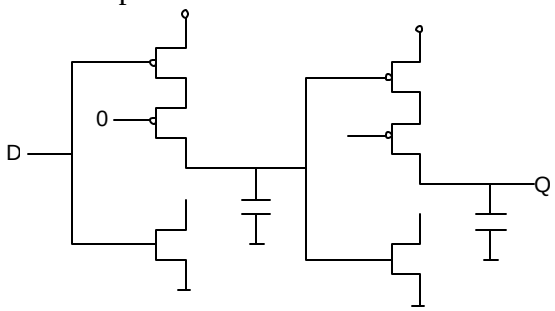
Insensitive to clk skew (Clk overlap)



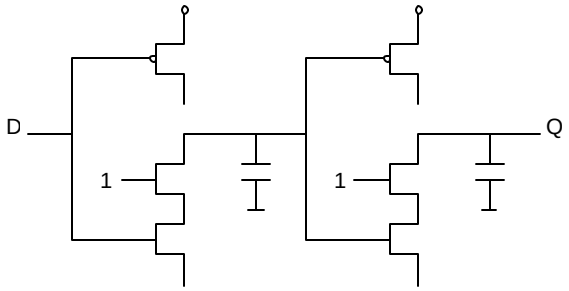
Clk=0 → Master is inverter, Slave is disabled
Clk=1 → Master is disabled, Slave is an inverter

How it is insensitive to Clk skew

0-0 overlap

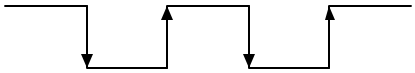


1-1 overlap

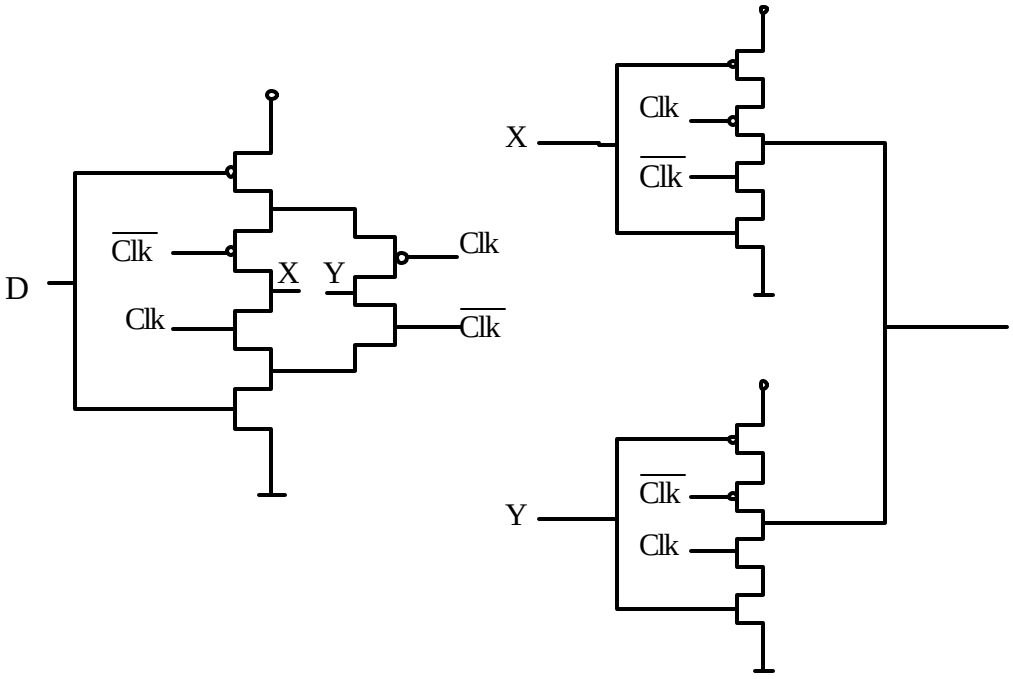


→ athis circuit will operate correctly by setting a hold time for the duration of overlap (data should not change during overlap)

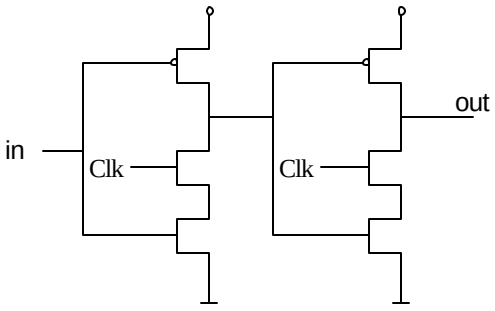
Dual Edge register



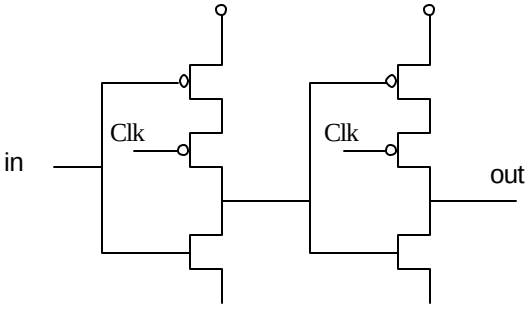
Data is written on both edges
Advantage → Clk frequency is half therefore less power is consumed



True Single Phase latches and registers

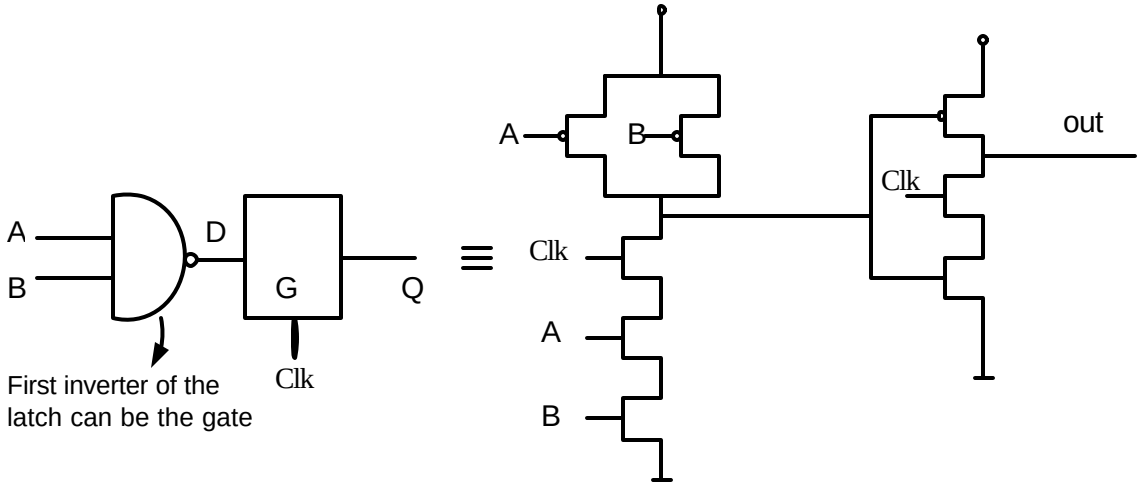


Clk = 1 → +ve latch

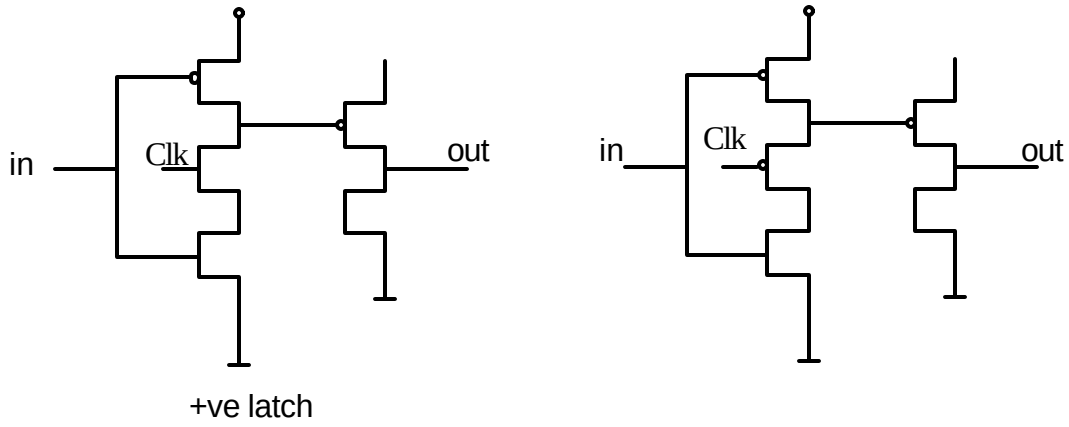


Clk = 0 → -ve latch

*With true single phase resisters you can include logic

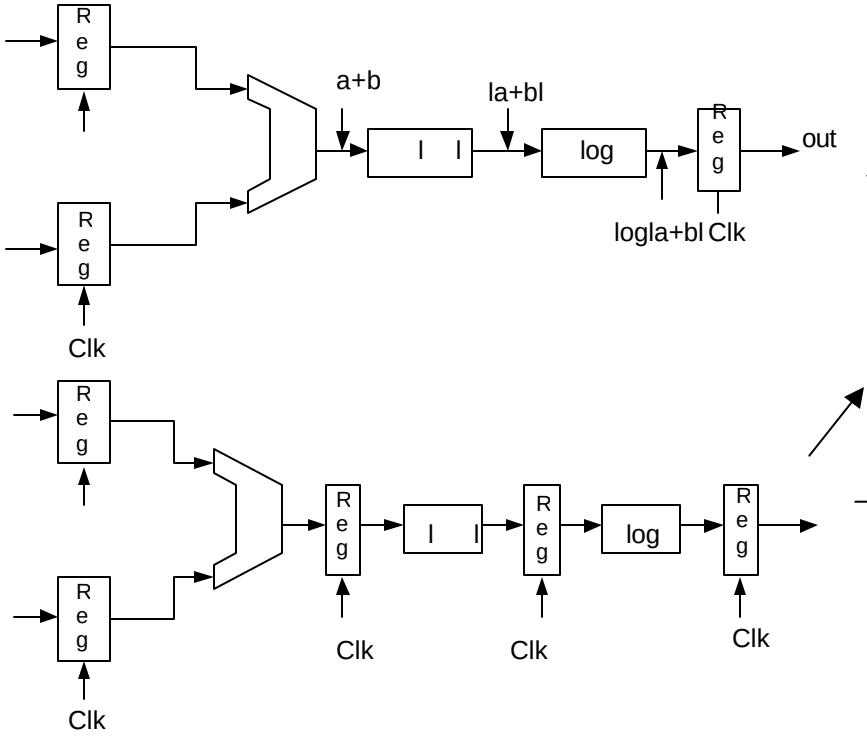


True Single Phase latch can be further simplified (split-output)



+ve latch

Pipelining (to speed up Sequential Ckts)



Clk is slower. You have to wait for the entire arithmetic function to take palce (worst path) and then read data with Clk.

On each clk cycle you can read data

Clk	Adder	| |	log
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\lvert a_1 + b_1 \rvert$	
3	$a_3 + b_3$	$\lvert a_2 + b_2 \rvert$	$\log \lvert a_1 + b_1 \rvert$
4	$a_4 + b_4$	$\lvert a_3 + b_3 \rvert$	$\log \lvert a_2 + b_2 \rvert$
5	$a_5 + b_5$	$\lvert a_4 + b_4 \rvert$	$\log \lvert a_3 + b_3 \rvert$

← we have data output on each clock

←cycle.(except for the first two cycle)