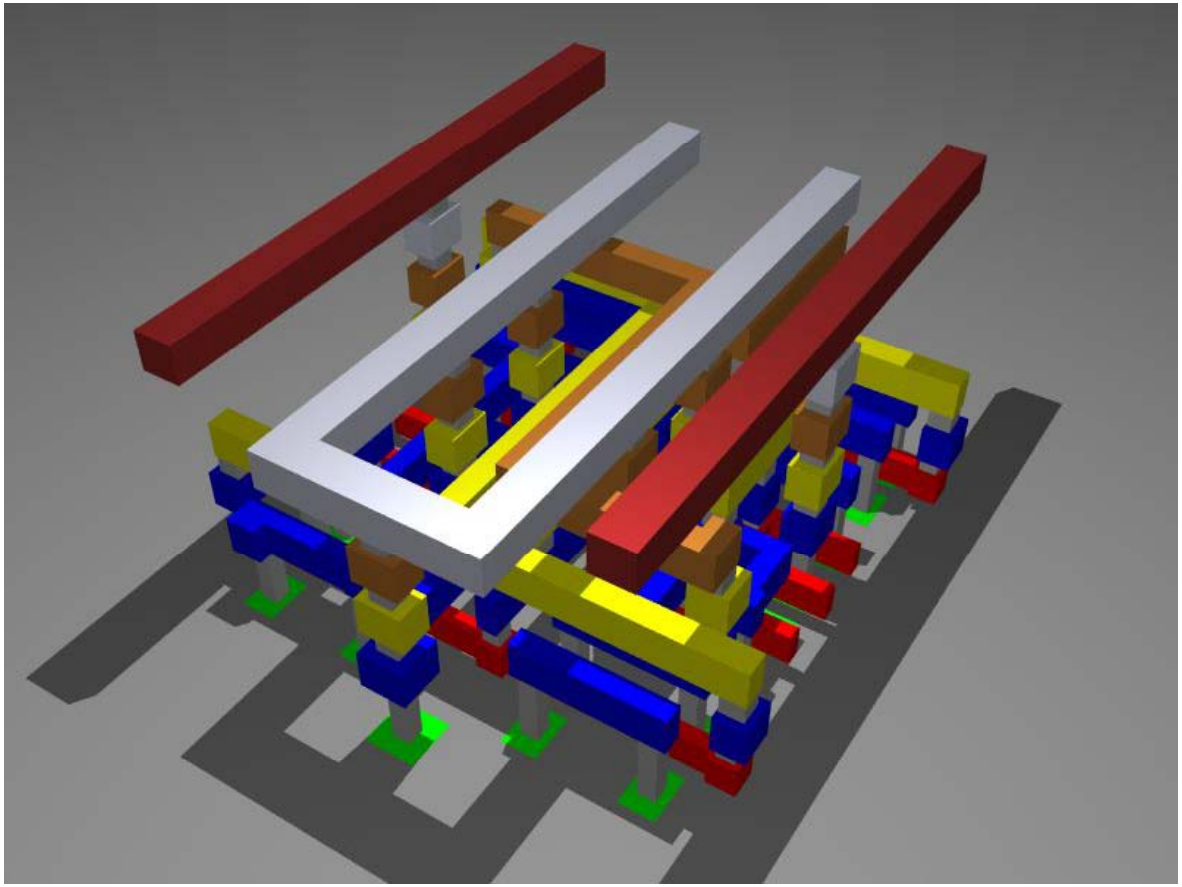
MODULE 5
Chapter 7



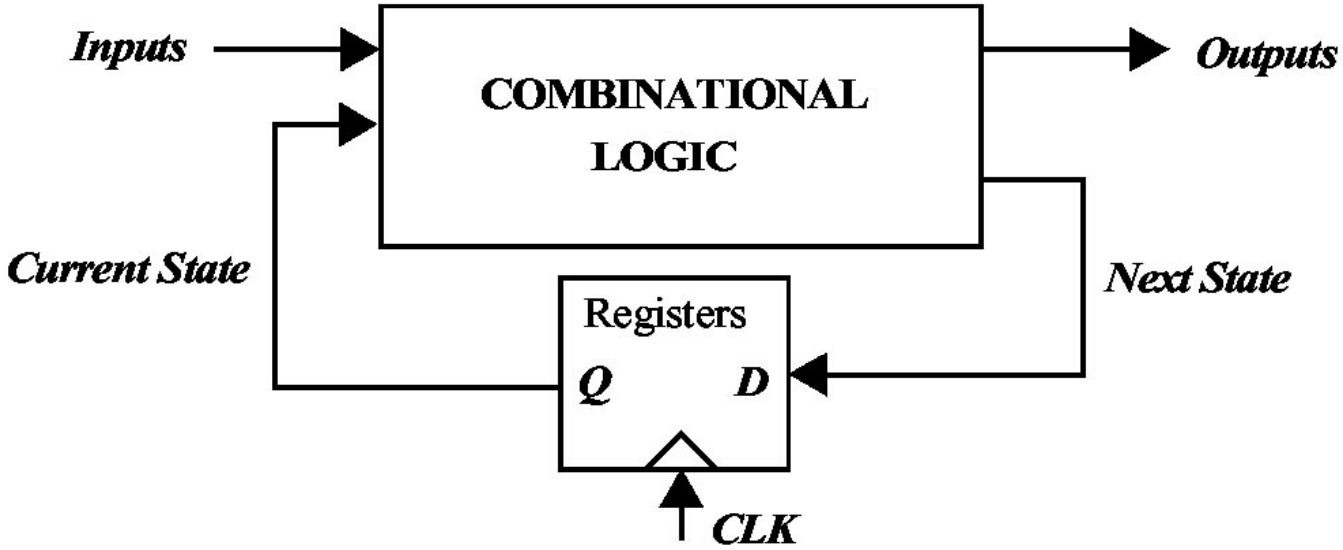
Clocked Storage Elements

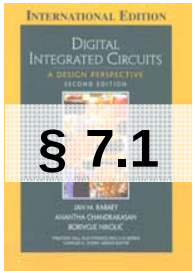
Clocked Storage Elements

Outline

- **Background**
  - **Timing, terminology, classification**
- **Static CSEs**
  - **Latches**
  - **Registers**
- **Dynamic CSEs**
  - **Latches**
  - **Registers**

FSM with Positive Edge Triggered Registers





- **CSEs/Flip-flops provide memory/state**
- **VLSI uses predominantly D-type flip-flops**

Memory elements

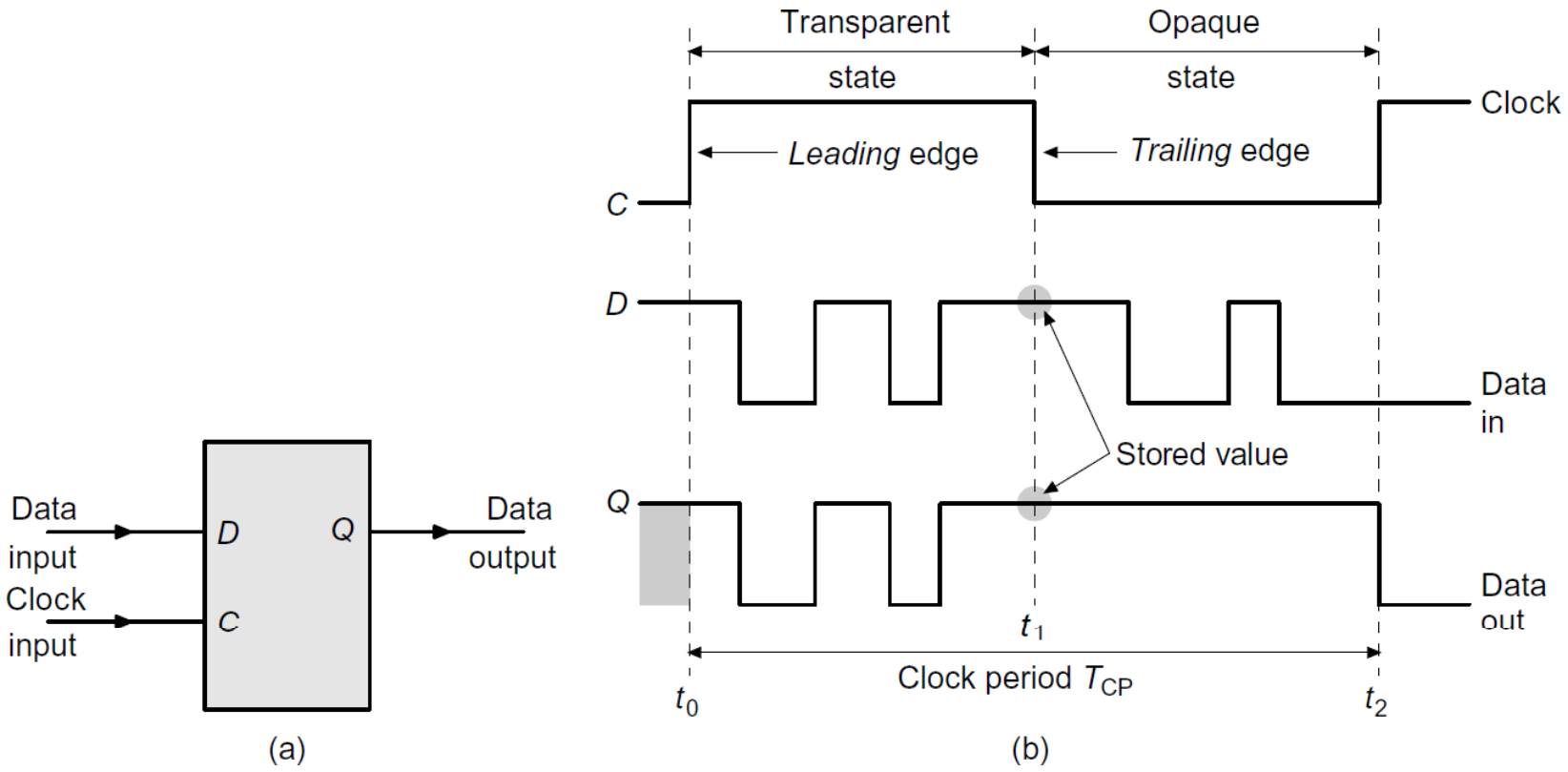
- Store a temporary value, remember a state
- Separate the past, current, future
- Typically controlled by clock.
- May have load signal, etc.
- In CMOS, memory is created by:
  - capacitance (dynamic);
  - feedback (static).

- Also see http://en.wikipedia.org/wiki/Flip-flop_(electronics)

Variations in memory elements

- **Form of required clock signal.**
- **How behavior of data input around clock affects the stored value.**
- **When the stored value is presented to the output.**
- **Whether there is ever a combinational path from input to output.**
- **Noise sensitivity on input and output**
- **…**

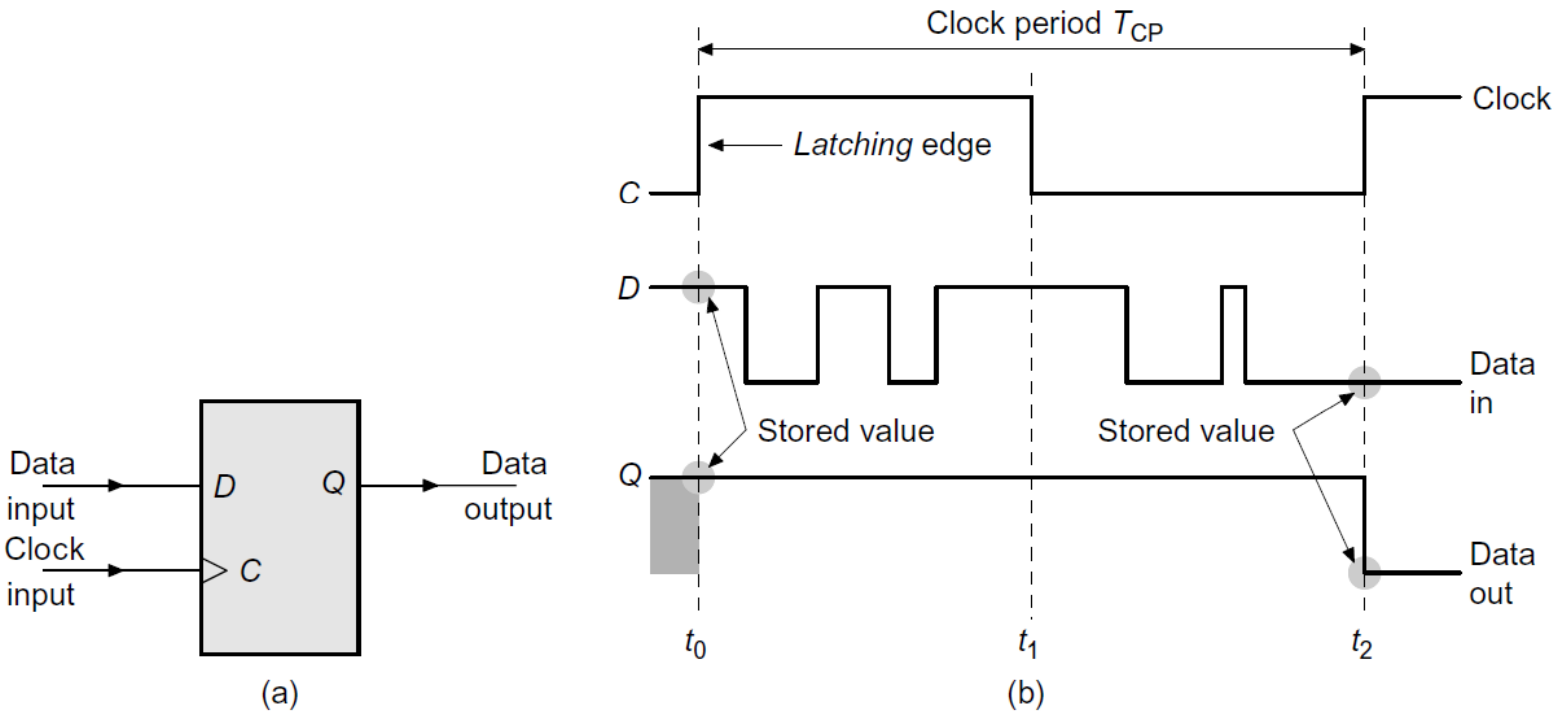
D-Latch



[Taskin, Kourtev & Friedman, The VLSI Handbook]

D-Register

(more commonly known as D-Flip-Flop)



[Taskin, Kourtev & Friedman, The VLSI Handbook]

Latches vs. Registers

Terminology of book

Latch

Level-sensitive

Transparent when clock is active

Clock active high: positive latch

Clock active low: negative latch

Faster, smaller

Register

Edge-triggered

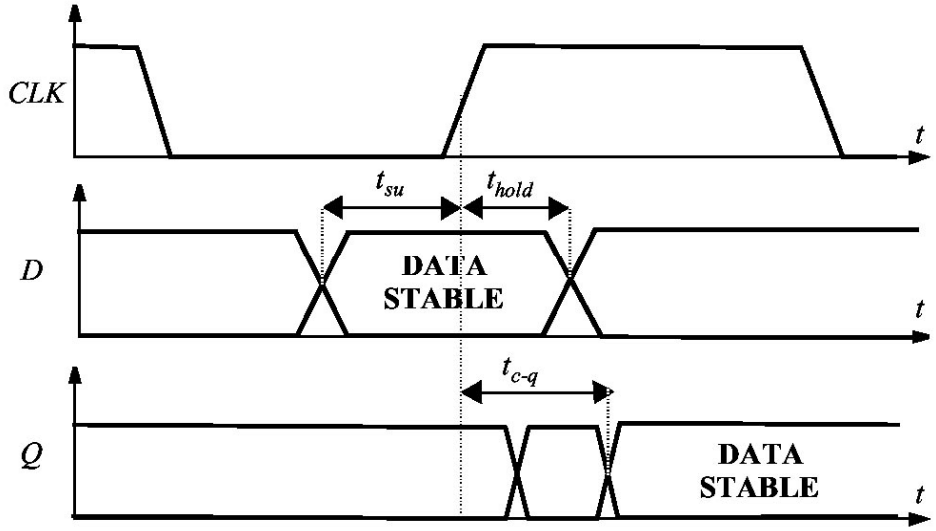
Input and output isolated

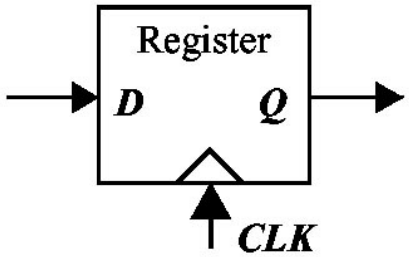
Sampling on 0 → 1 clock: positive edge triggered

Sampling on 1 → 0 clock: negative edge triggered

Safer

Timing Metrics Reminder





$t_{c\text{-}q}$: delay from clock (edge) to Q

t_{su} : setup time

t_{hold} : hold time

t_{plogic} : worst case propagation delay of logic

t_{cd} : best case propagation delay (contamination delay)

T : clock period

$$T \geq t_{c\text{-}q} + t_{plogic} + t_{su}$$

$$t_{cdregister} + t_{cdlogic} \geq t_{hold}$$

Static vs. Dynamic Memory Elements

Static

Operate through positive feedback

Preserve state as long as power is on

Can work when clock is off

More robust

Dynamic

Store charge on (parasitic) capacitor

Charge leaks away (in milliseconds)

Clock must be kept running (for periodic refresh)

Faster, smaller

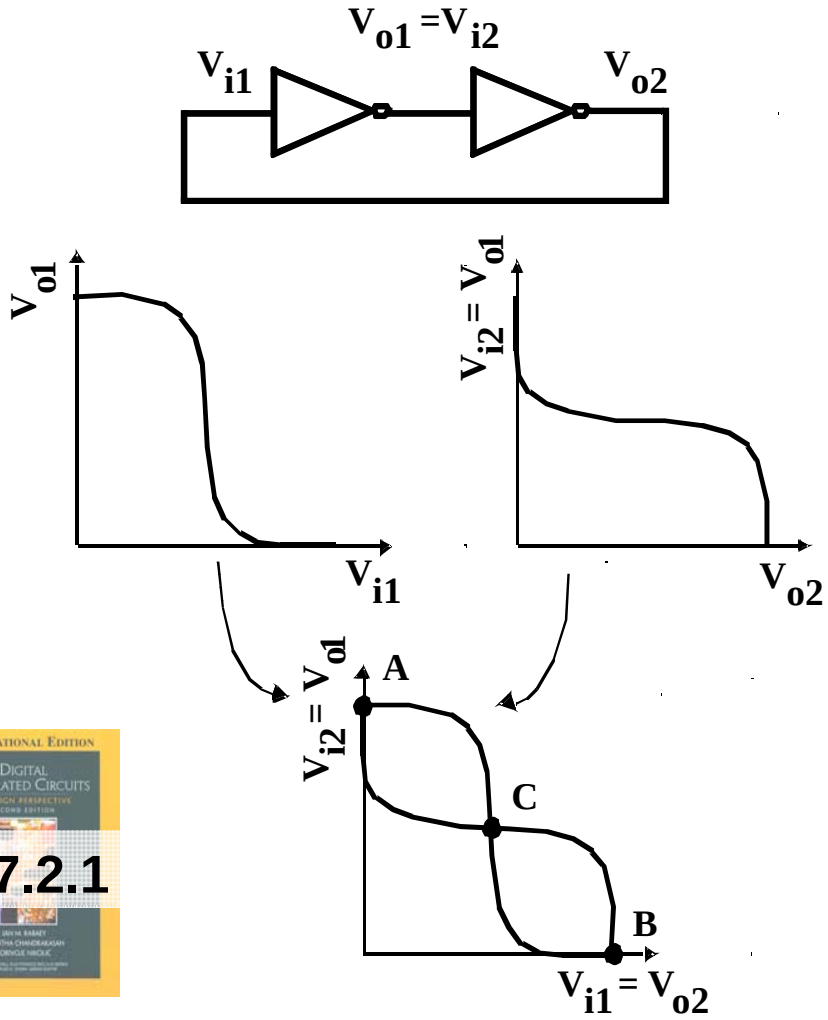
Static CSEs / Flipflops

- Latches
- Registers

Large part of literature talks about Clocked Storage Elements

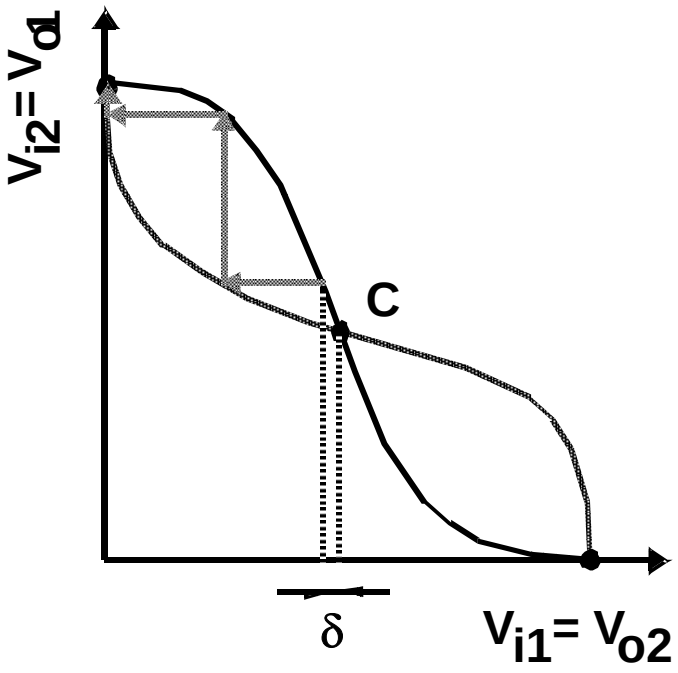
Are called Flip-Flops in book

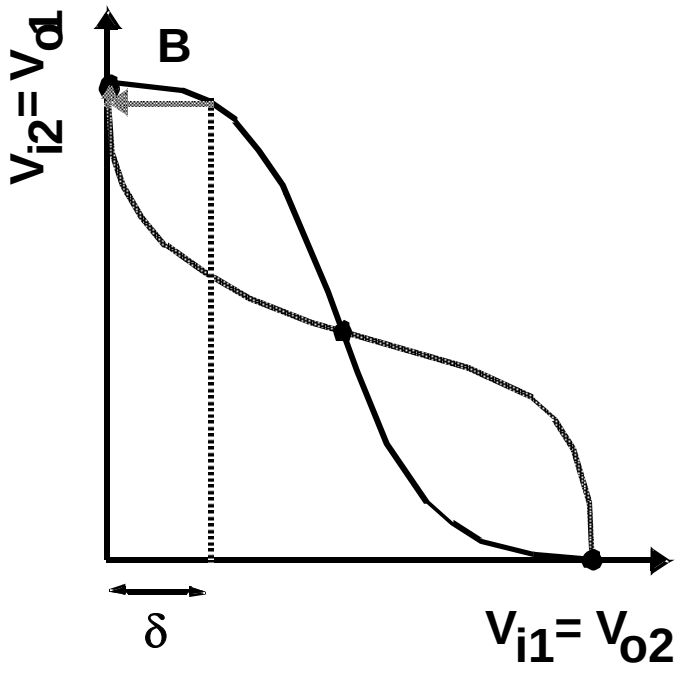
Positive Feedback: Bi-Stability



- **Loop-gain in A,B <<1**
- **A,B: stable points**
- **Loop-gain in C >> 1**
- **C: meta-stable point**

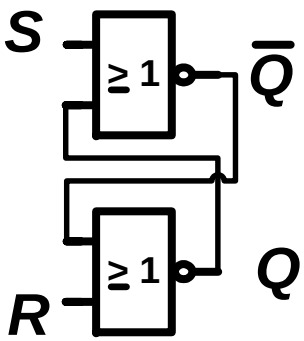
Meta-Stability

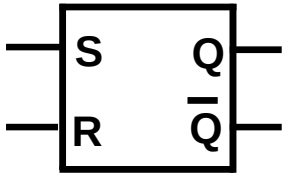




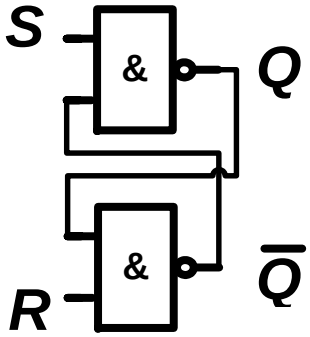
- **Gain should be larger than 1 in the transition region**
- **Smaller than 1 in stable region**

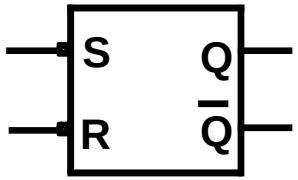
SR-Latch



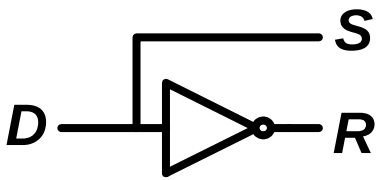


S	R	Q	$\overline{Q}$	
0	0	Q	$\overline{Q}$	
1	0	1	0	
0	1	0	1	
1	1	0	0	← forbidden



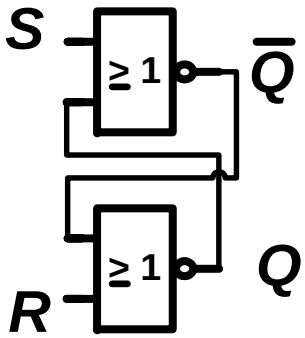


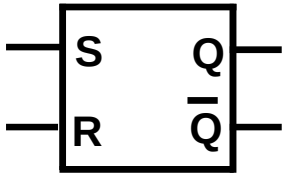
S	R	Q	$\overline{Q}$	
1	1	Q	$\overline{Q}$	
0	1	1	0	
1	0	0	1	
0	0	1	1	← forbidden

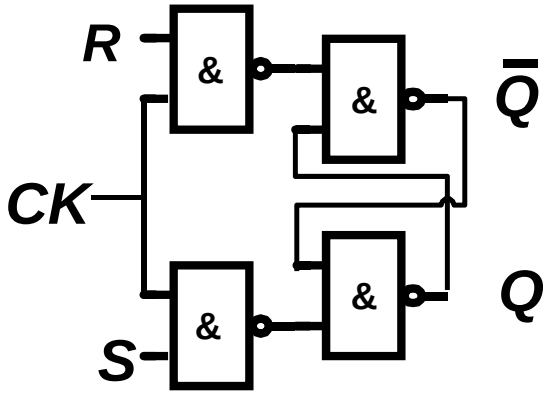


- Construction of D-latch
- D-latch most common in VLSI

Clocked SR-Latch

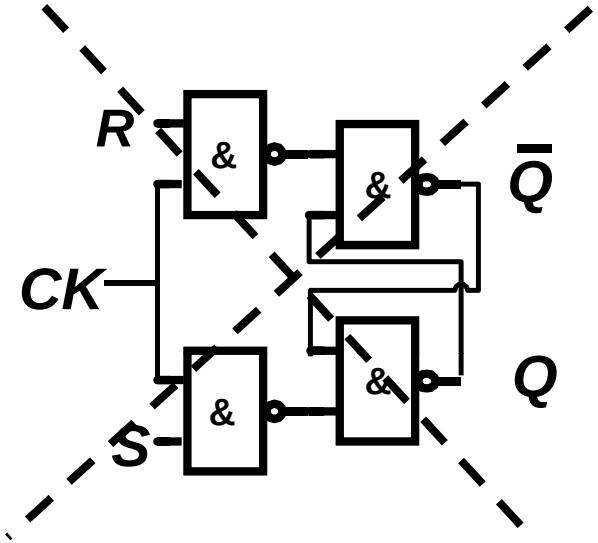


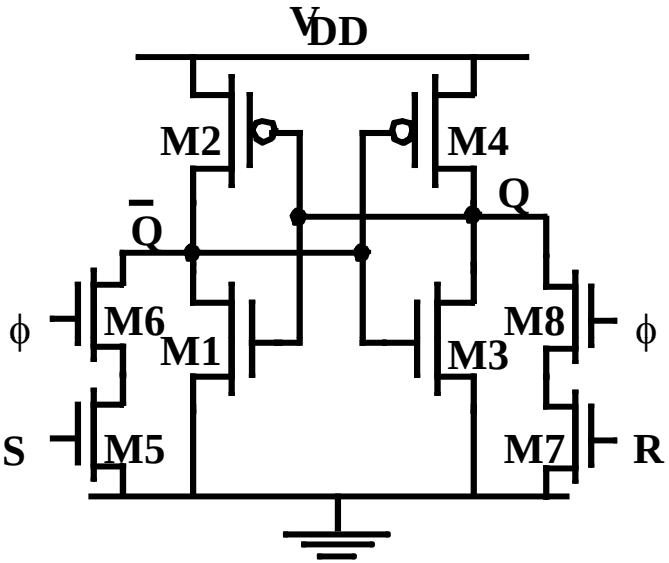




- Naïve implementation
- 16 transistors
- D latch requires 9xN, 9xP
- Master-slave D-register needs 18xN, 18xP
- Larger area, cost, power

CMOS Clocked SR-Latch





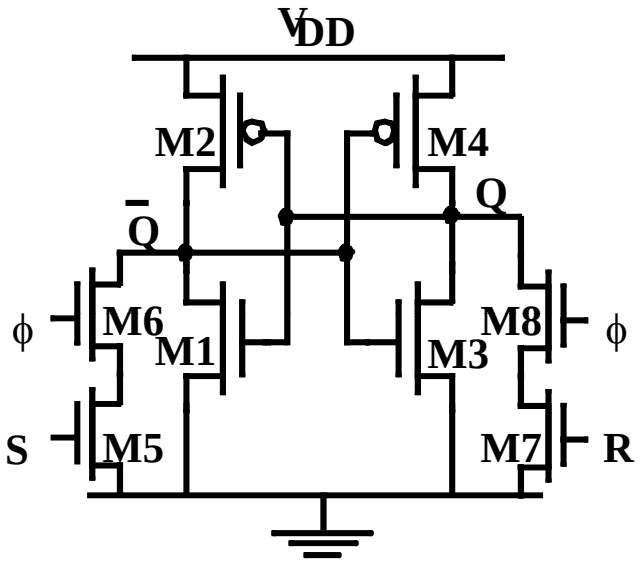
$\phi = CK$

- Save 6 PMOS, 2 NMOS transistors
- D-latch requires 7 x N, 3 x P (instead of 9xN, 9xP)

{TPS}: Is this a ratioed design or not?
Does it consume static power?

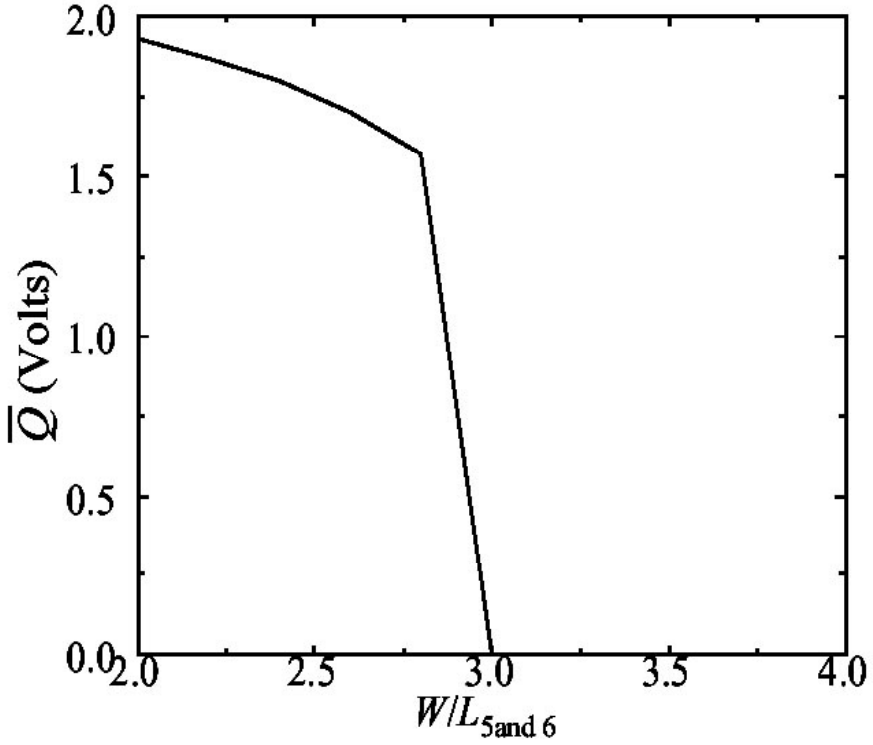
Sizing for 'Set' Action

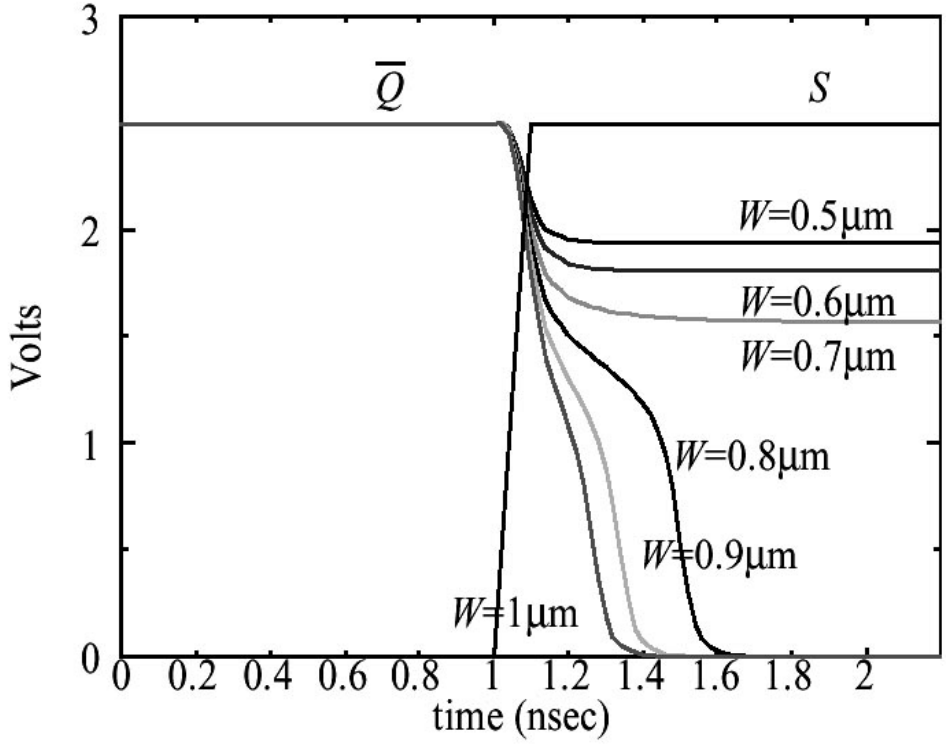
- M_3-M_4 form conventional inverter
- Model M_5-M_6 as one equivalent (double length) transistor M_{56}
- Assume Q = 0 ➔ M_1 is off, M_2 is on
- M_2-M_{56} operate like ratioed pseudo NMOS inverter
- Latch switches when M_{56} pulls input of M_3-M_4 below their switching threshold (assume $V_{DD}/2$)
- Positive feedback amplifies switching
- M_2 and M_{56} both in velocity saturation around $V_O = V_{DD}/2$



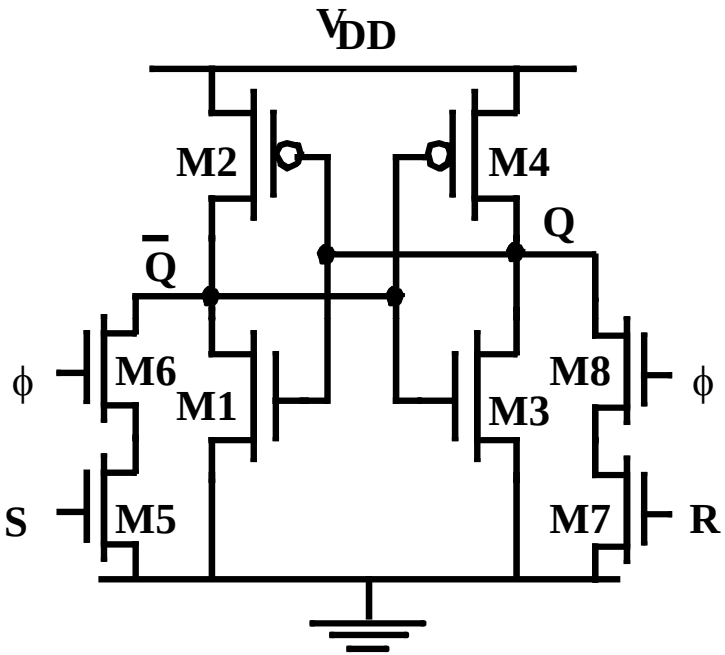
$$k_n'\left(\frac{W}{L}\right)_{5-6}\left((V_{DD}-V_{Tn})V_{DSATn}-\frac{V_{DSATn}^2}{2}\right)=k_p'\left(\frac{W}{L}\right)_2\left((-V_{DD}-V_{Tp})V_{DSATp}-\frac{V_{DSATp}^2}{2}\right)$$

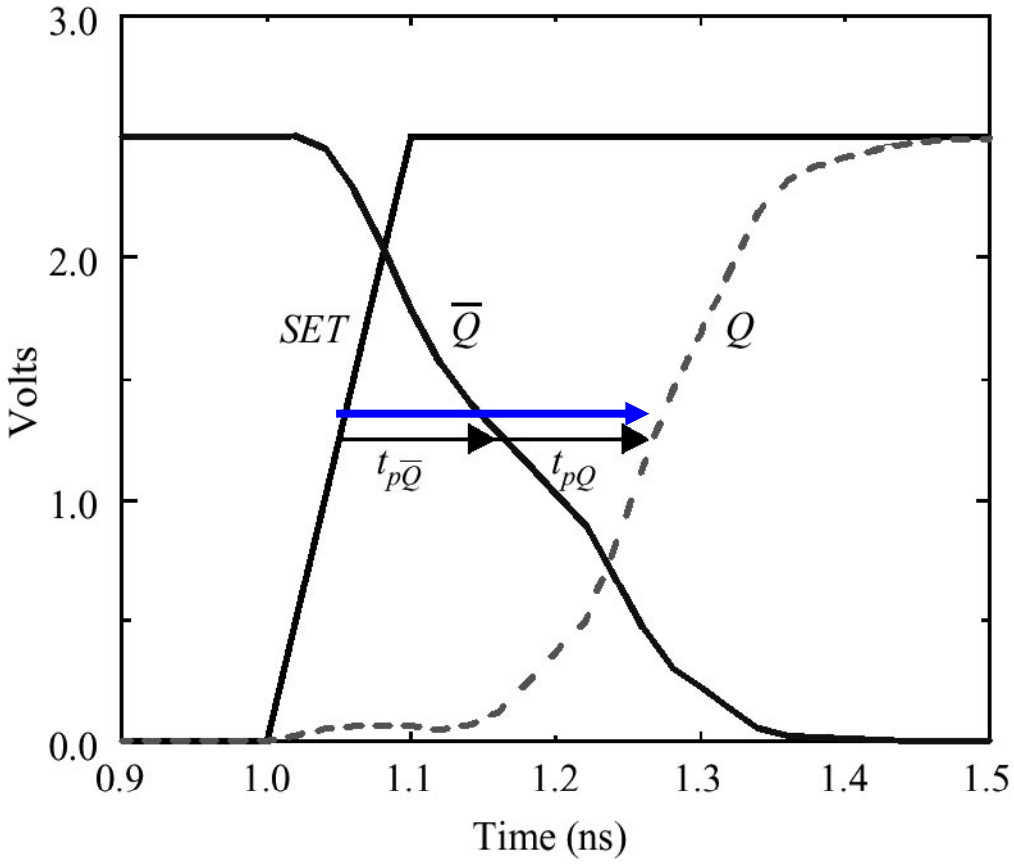
Sizing for ‘Set’ Action



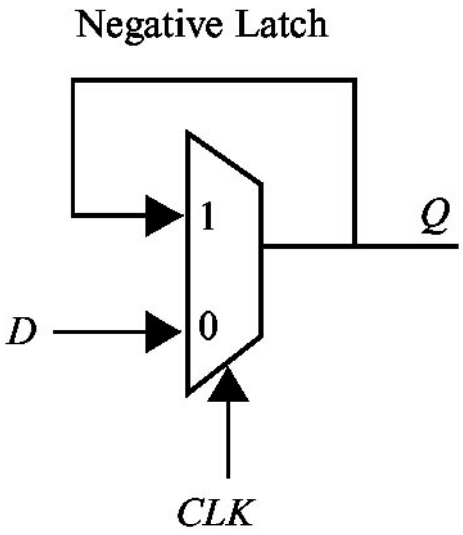


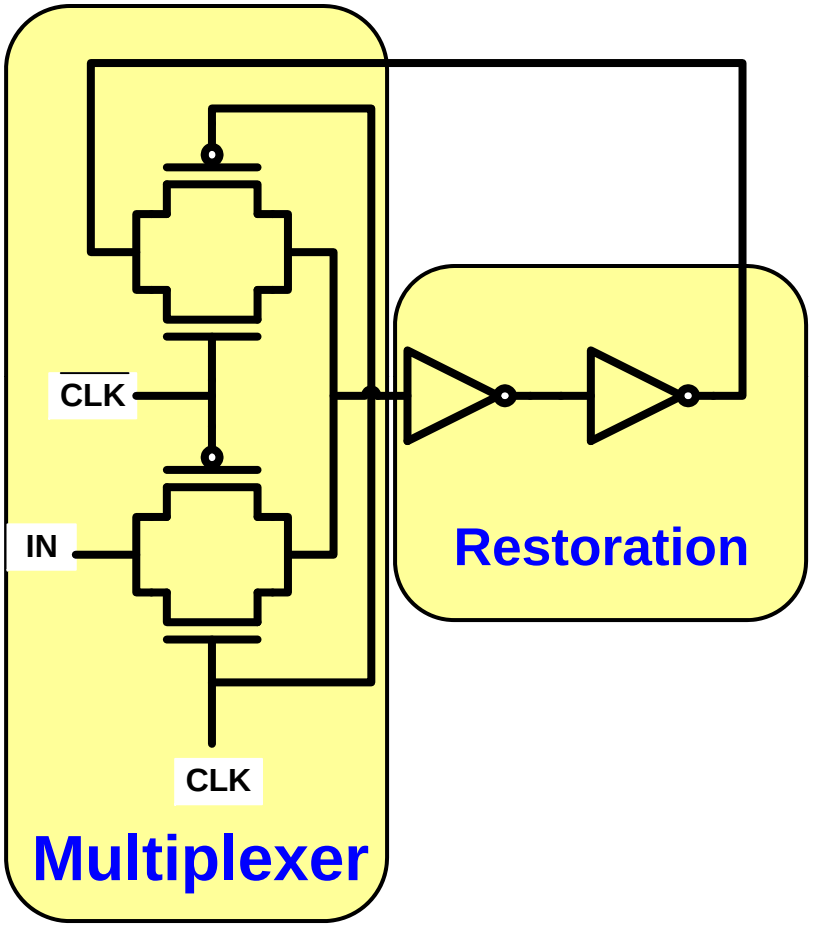
SR Latch Timing





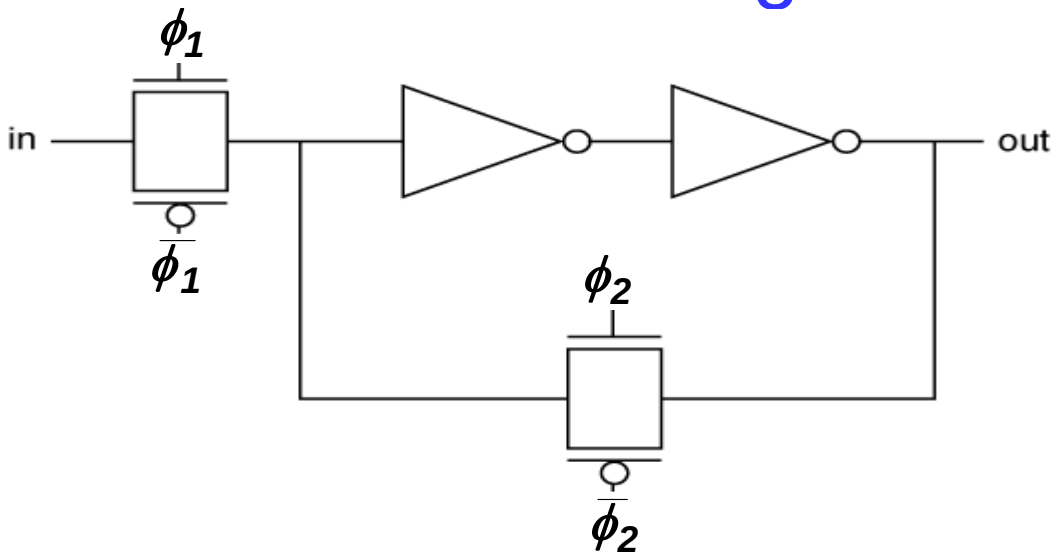
Multiplexer-Based Latches





Mux-based latches much more common in modern dig. IC's

Recirculating latch



- **Quasi-static, static on one phase**
- **Feedback restores value**
- **Requires 4 x N, 4 x P, minimum size**
  (compare 7 x N, 3 x P, non-minimum size)
- **ϕ_1 and ϕ_2 inverse but should be non-overlapping**
- **Definitely not ideal, because… {TPS}**

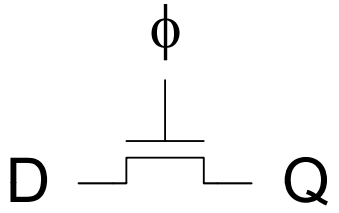
Let's explore TG/PG based latch designs

Latch Design

Pass Transistor Latch

- ☺ Tiny
- ☺ Low clock load

- ☹ V_t drop
- ☹ nonrestoring
- ☹ backdriving
- ☹ output noise sensitivity
- ☹ dynamic
- ☹ diffusion input



Used in 1970's

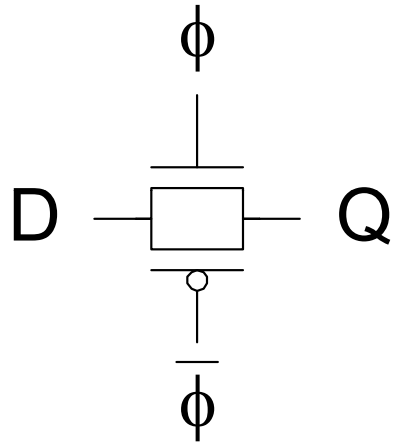
["Latch Design" slides based on Weste & Harris],

Latch Design

Transmission gate

☺ **No V_t drop**

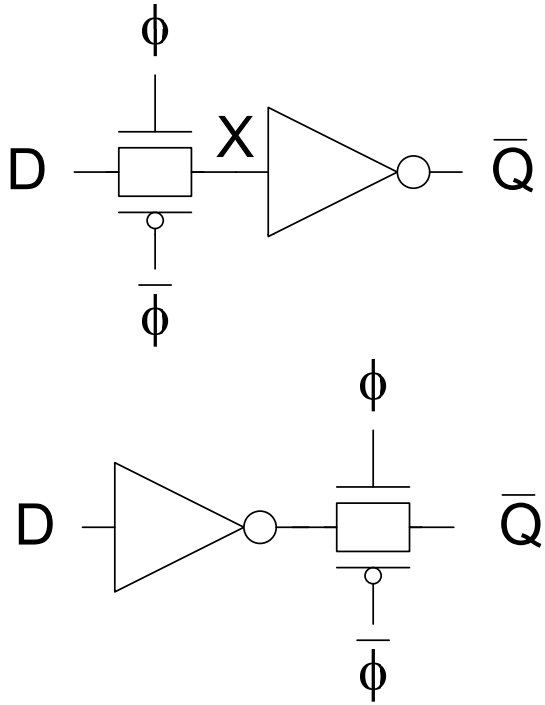
☹ **Requires inverted clock**



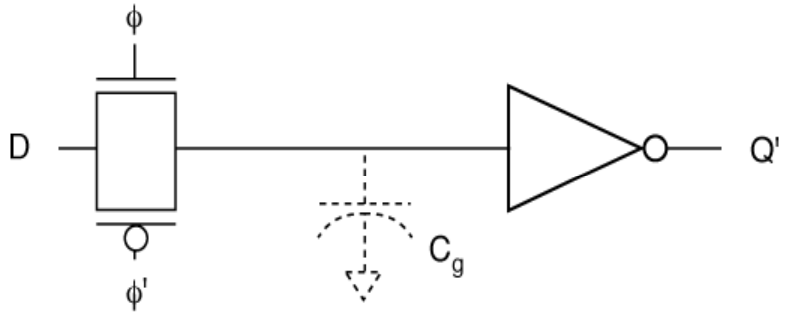
Latch Design

Inverting buffer / Basic Dynamic Latch

- ☺ **Restoring**
- ☺ **No backdriving**
- ☺ **Fixes either**
  - **Output noise sensitivity**
  - **Or diffusion input**
- 😐 **Inverted output**



Dynamic Latch

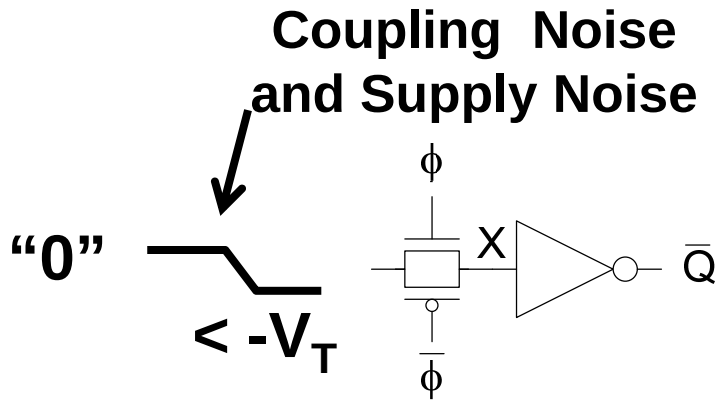


- **Storage capacitance comes primarily from inverter gate capacitance.**
- **Setup and hold times determined by transmission gate—must ensure that value stored on transmission gate is solid.**
- **Stored charge leaks away**
  - **Duration of stored value being good depends on technology**
  - **Worst conditions during burn-in (High VDD, high Temp).**
- **Modern technologies (almost) mandate static latches.**

Noise Sensitivity

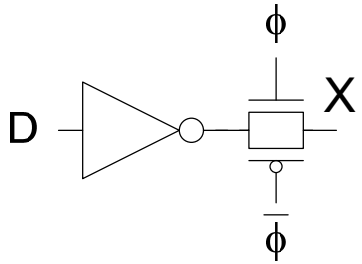
Diffusion Input Noise Sensitivity

- Noise on input can drop node below V_T
- TG NMOS turns on, and X can discharge if it was a “1”
- Similar problems for $V_{in} > V_{DD}$



Output Noise Sensitivity

- State node X is exposed
- Noise spike on output can corrupt the state

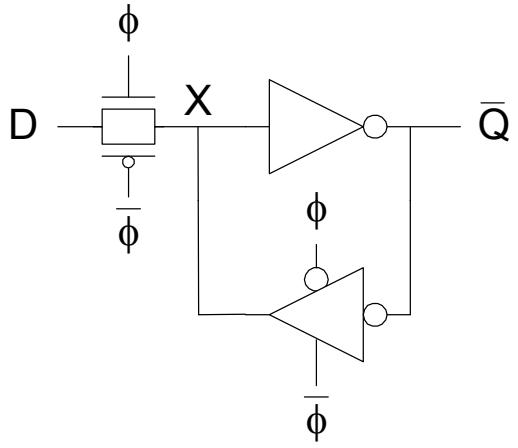


Latch Design

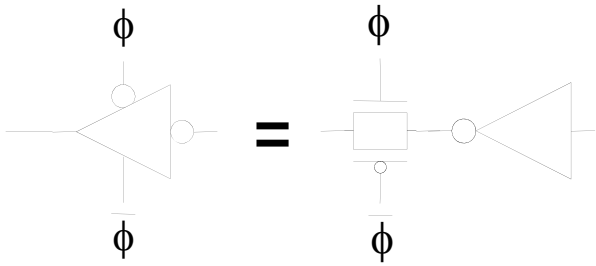
Tristate feedback

☺ **Static**

☹ **Backdriving risk**



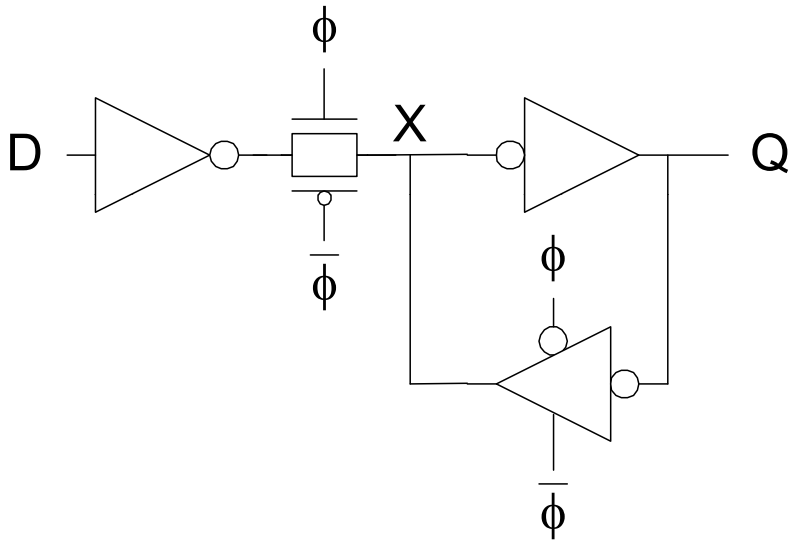
- **Static latches are now essential because of leakage**
  - **If only during burn-in (@ high VDD, T)**



Latch Design

Buffered input

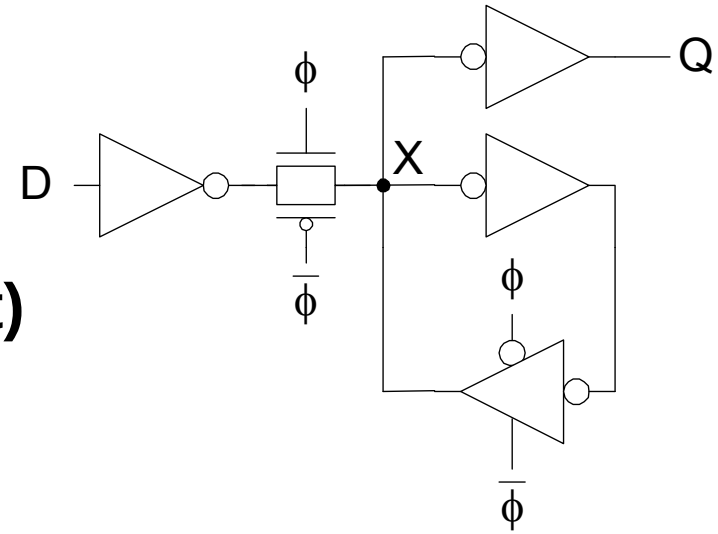
- ☺ **Fixes diffusion input**
- ☺ **Noninverting**



Latch Design

Buffered output

Widely used in standard cells



- ☺ **No backdriving**
- ☺ **Very robust (most important)**

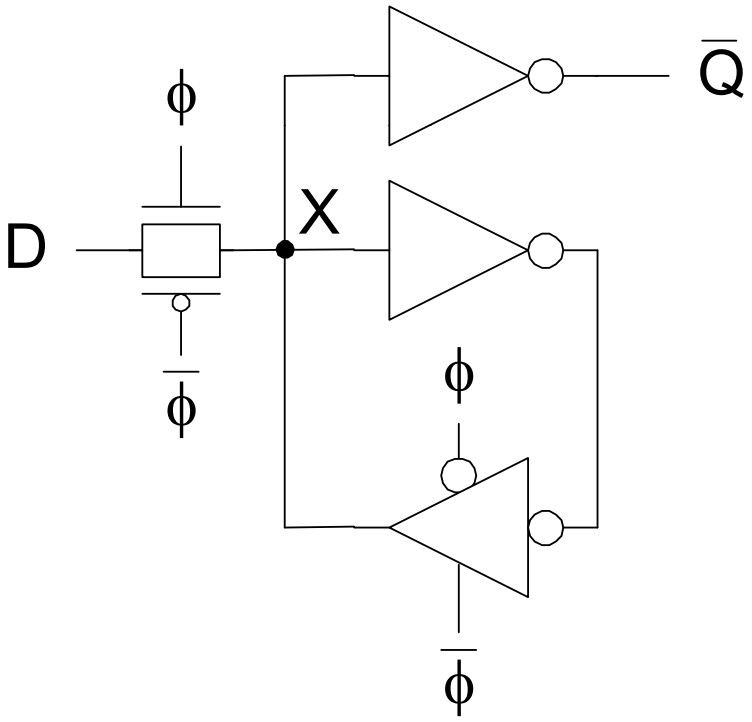
- ☹ **Rather large**
- ☹ **Rather slow (1.5 – 2 FO4 delays)**
- ☹ **High clock loading**

Latch Design

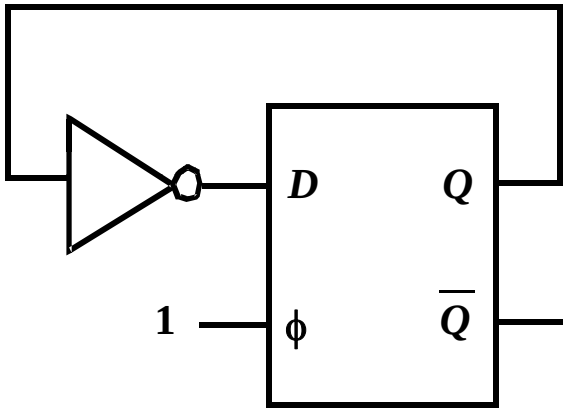
Datapath latch

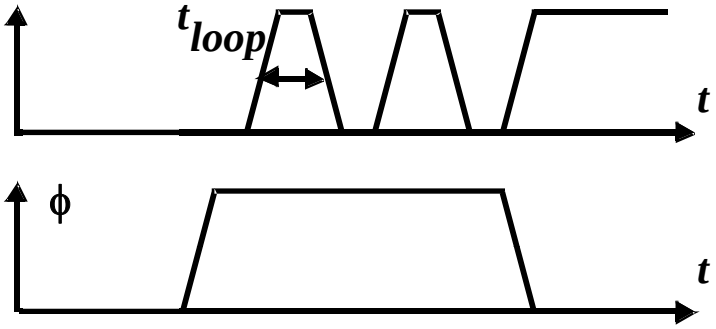
(only use in noise-controlled environments)

- ☺ **Smaller**
- ☺ **Faster**
- ☹ **Unbuffered input**



Latch Designs can Suffer from Race Problems

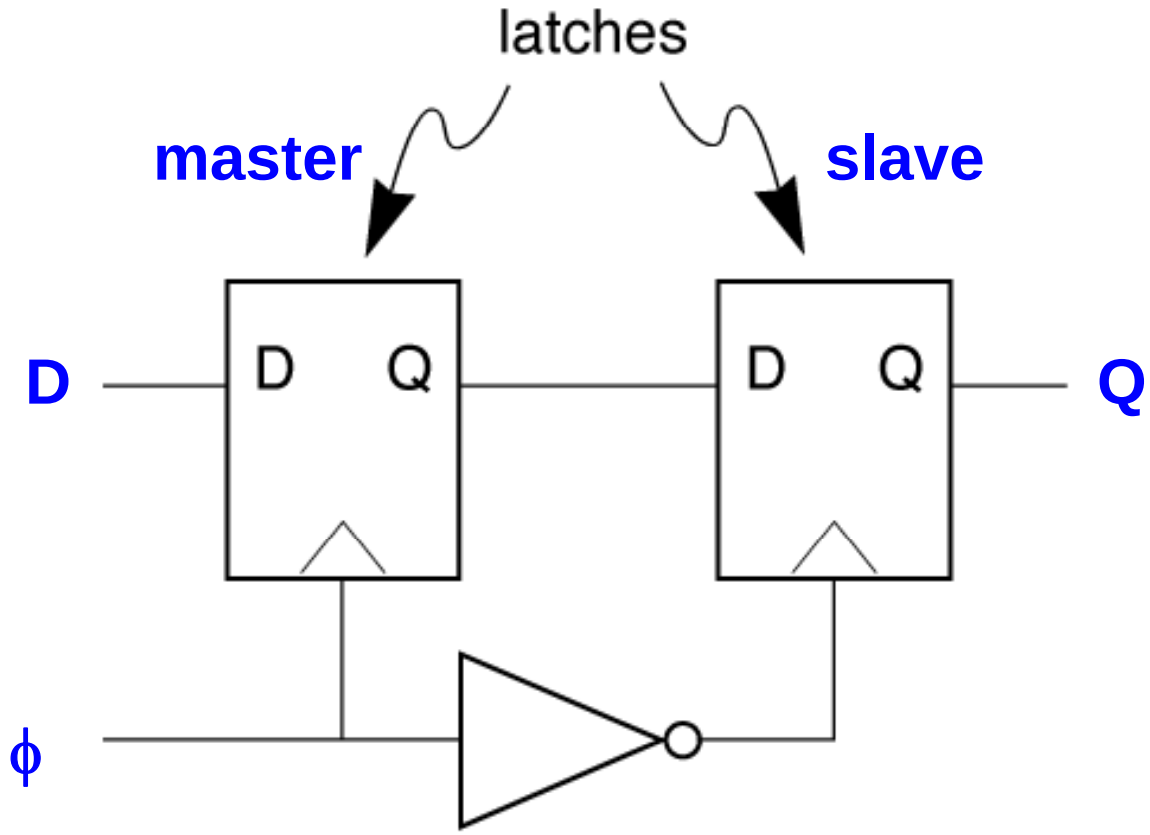




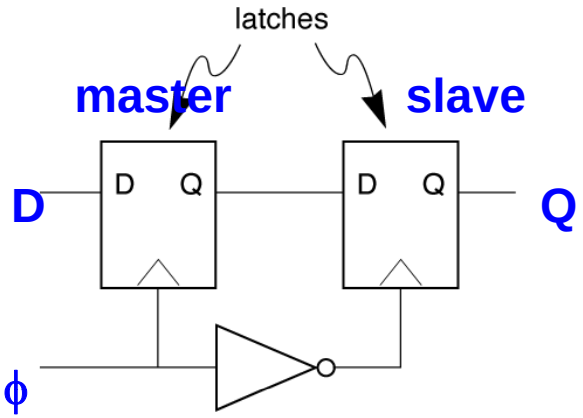
Signal can race around during $\phi = 1$

Registers

- **Not transparent—use multiple storage elements to isolate output from input.**
- **Master-slave, edge triggered principle**



Master-slave operation



ϕ = 0:

- **master latch is disabled;**
- **slave latch is enabled,**
- **but master latch output is stable,**
- **so output does not change.**

ϕ = 1:

- **master latch is enabled,**
- **loading value from input;**
- **slave latch is disabled,**
- **maintaining old output value.**

ϕ = 1 → 0:

- **Slave latch copies current value of master, and master stops changing**

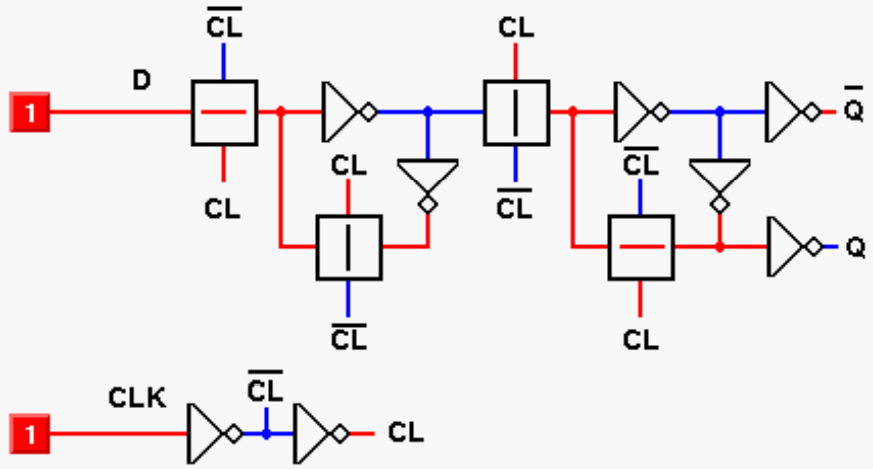
CMOS Flip-Flop Construction

CMOS technology allows a very different approach to flip-flop design and construction. Instead of using logic gates to connect the clock signal to the master and slave sections of the flip-flop, a CMOS flip-flop uses *transmission gates* to control the data connections. (See the CMOS gate electronics page for a closer look at the transmission gate itself.)

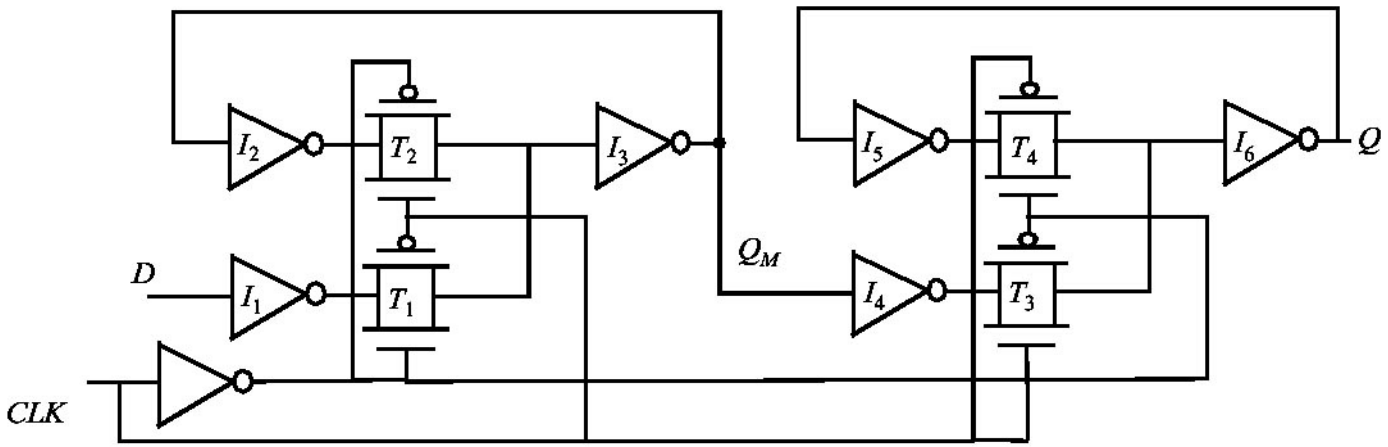
The result is that a controllable flip-flop can be built with only inverters and transmission gates — a very small and simple structure for an IC.

The basic CMOS D flip-flop is shown below.

clickable

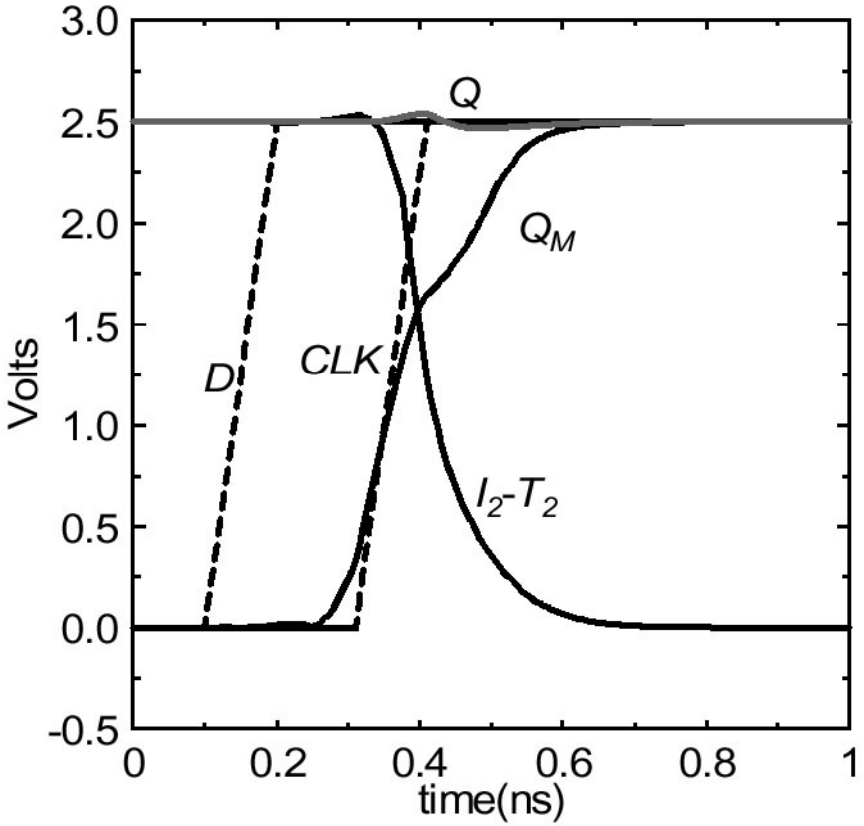


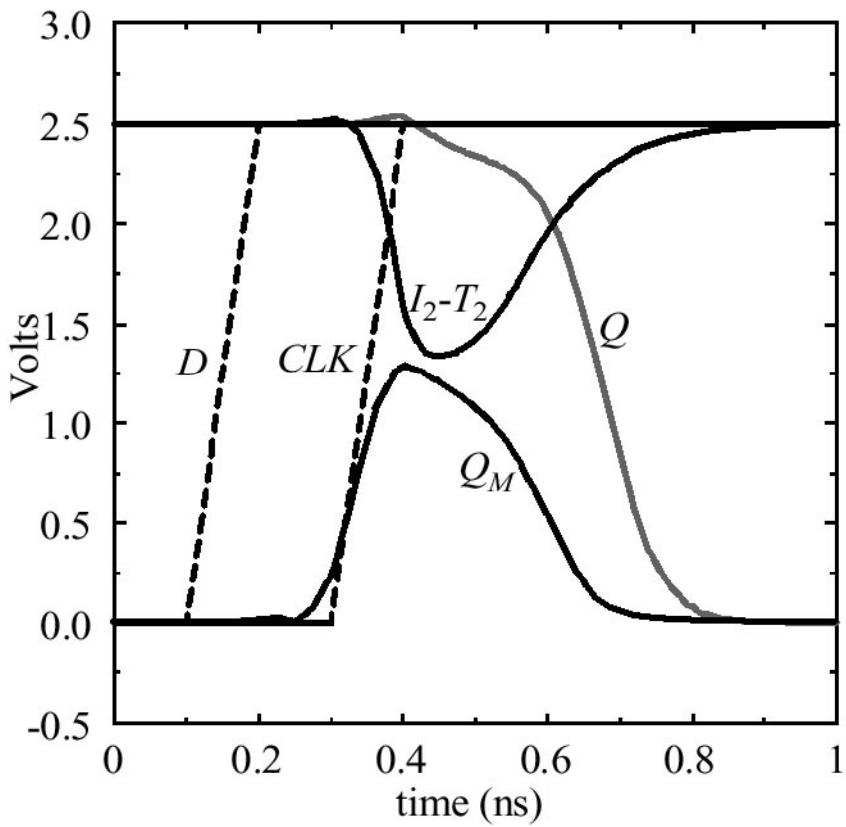
Transistor Level Master Slave Positive Edge Triggered Register



- **Robust Design**
- **Can eliminate I_1 and I_4, however, they make design more robust (avoid charge sharing, robust input) (see next)**
- **High Clock Load (8 x)**

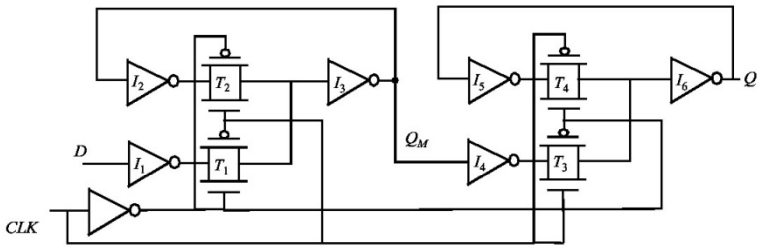
Set-up Time Simulation

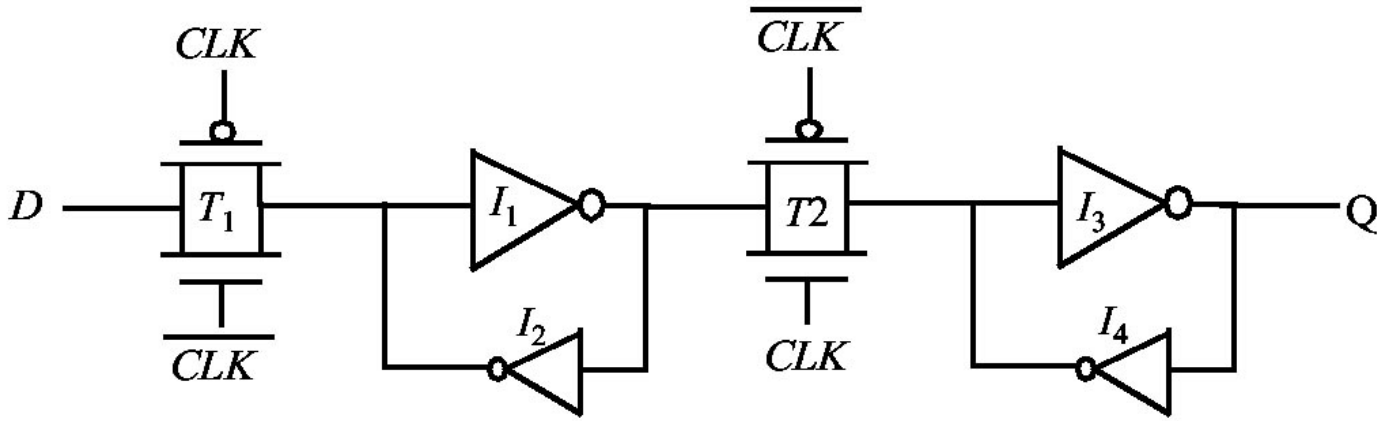




Slightly smaller delay between D and CLK

Ratioed Reduced Clock Load Register

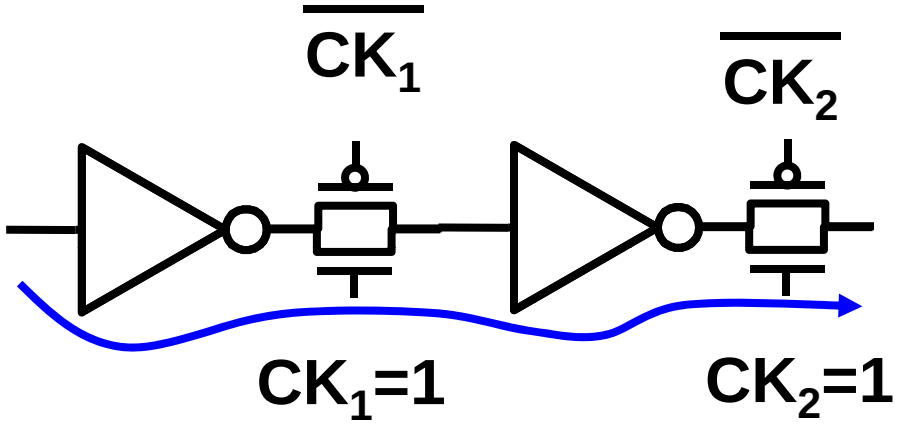




- **I_2 and I_4 are small, even long**
- **Lower clock load**
- **Increased design complexity**
- **Reduced robustness (reverse conduction / backdriving)**

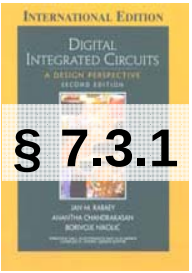
Simple Master-Slave Register

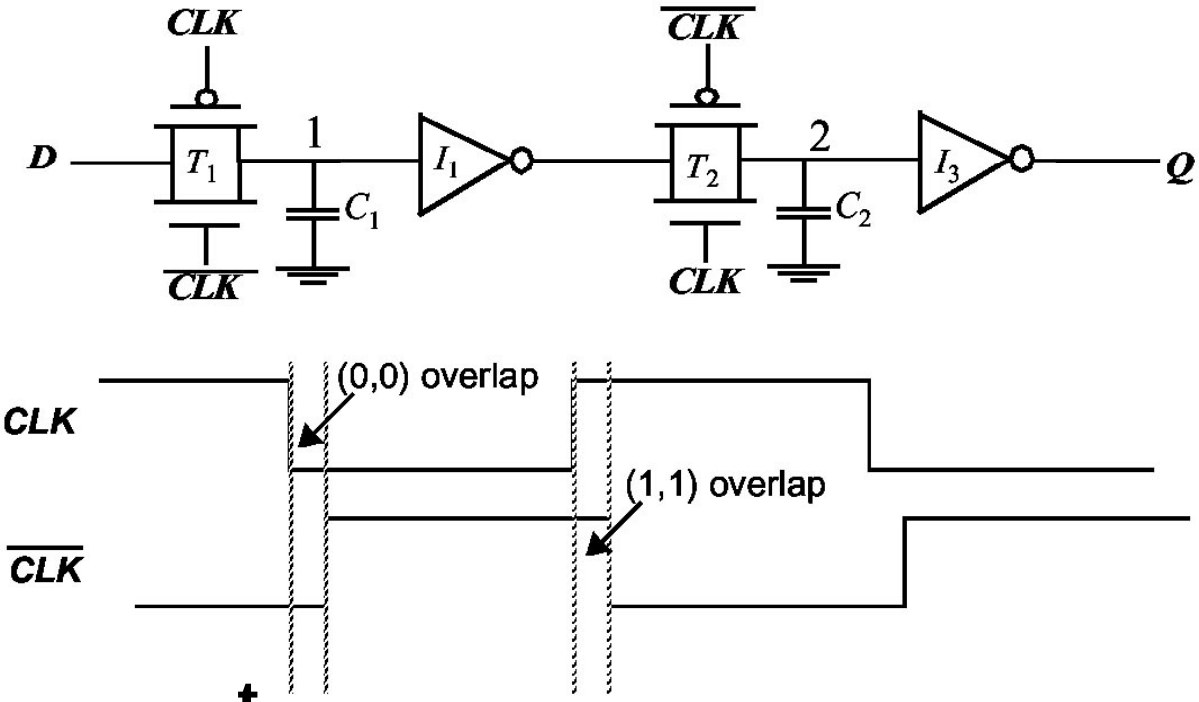
Clock phase overlap is important design problem



- **Similar problem for 0-0 overlap**

Dynamic Edge Triggered Register





t_{su}	t_{T1}
t_{hold}	approximately zero
t_{cq}	$t_{I1} + t_{T2} + t_{I3}$
$t_{overlap\ 0\text{-}0}$	$< t_{T1} + t_{I1} + t_{T2}$ Prevent race through T_1, I_1, T_2
t_{hold}	$> t_{overlap\ 1\text{-}1}$ Enforce hold-time constraint

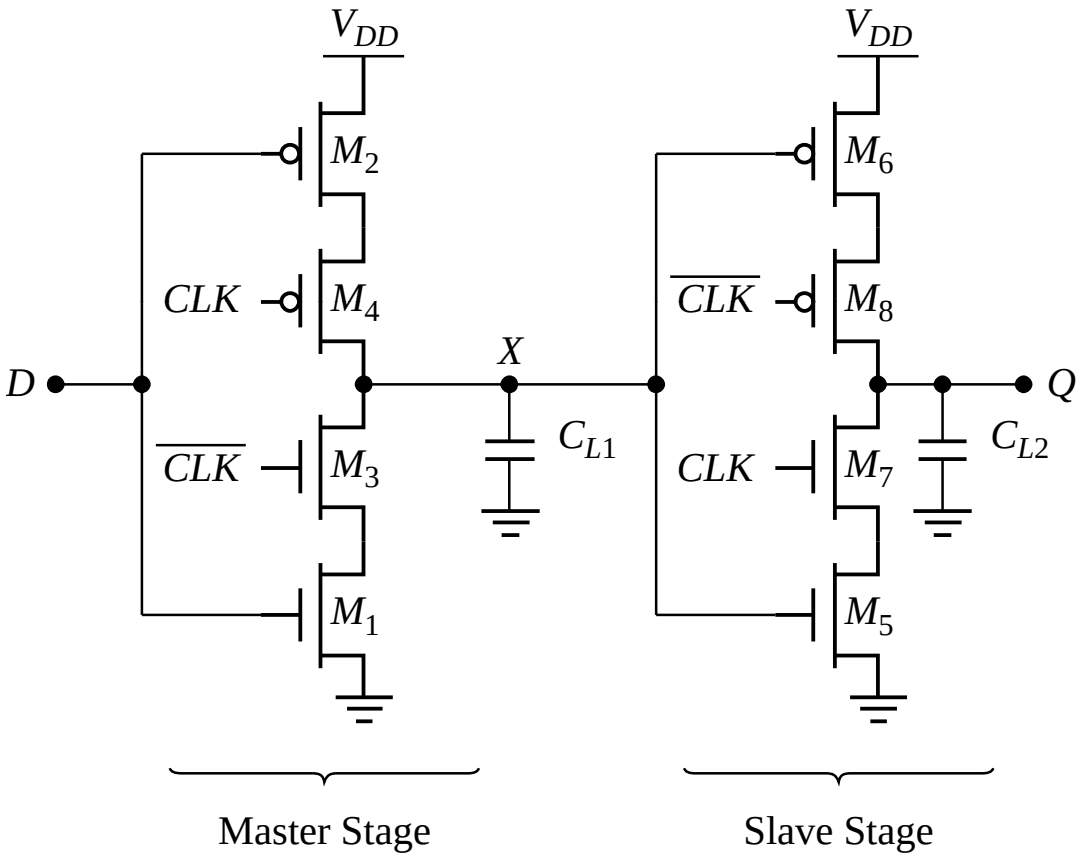
Clocking and CSE problems

- **Clock overlap with multiple phases great problem in view of clock skew**
- **Power – in some cases, around 50% of total power**
- **Delay – setup time, clk-to-Q, ..**
- **Robustness**
- **Noise-sensitivity of element, specifically for dynamic elements**

Next: advanced CSEs to combat some of the above

Other Latches/Registers: C²MOS

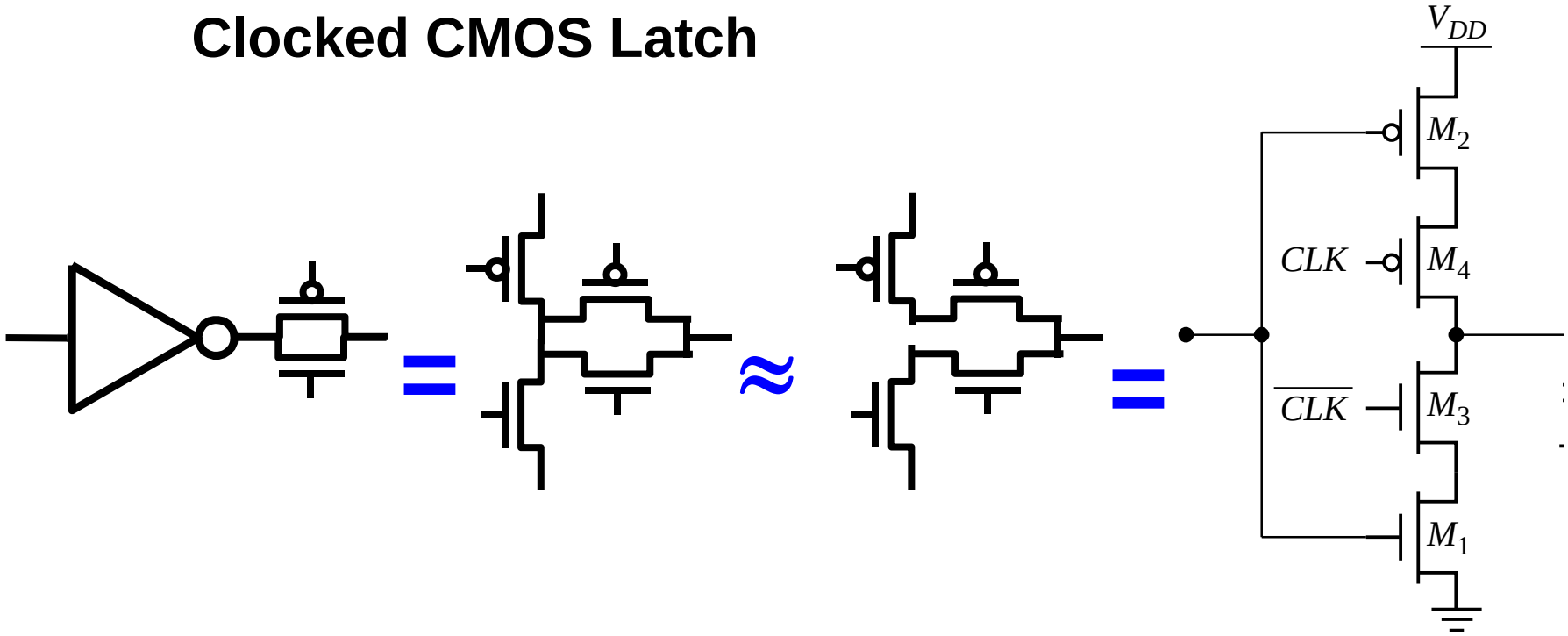
Clocked CMOS Register – insensitive to overlap



"Keepers" can be added to make circuit pseudo-static

C²MOS Latch

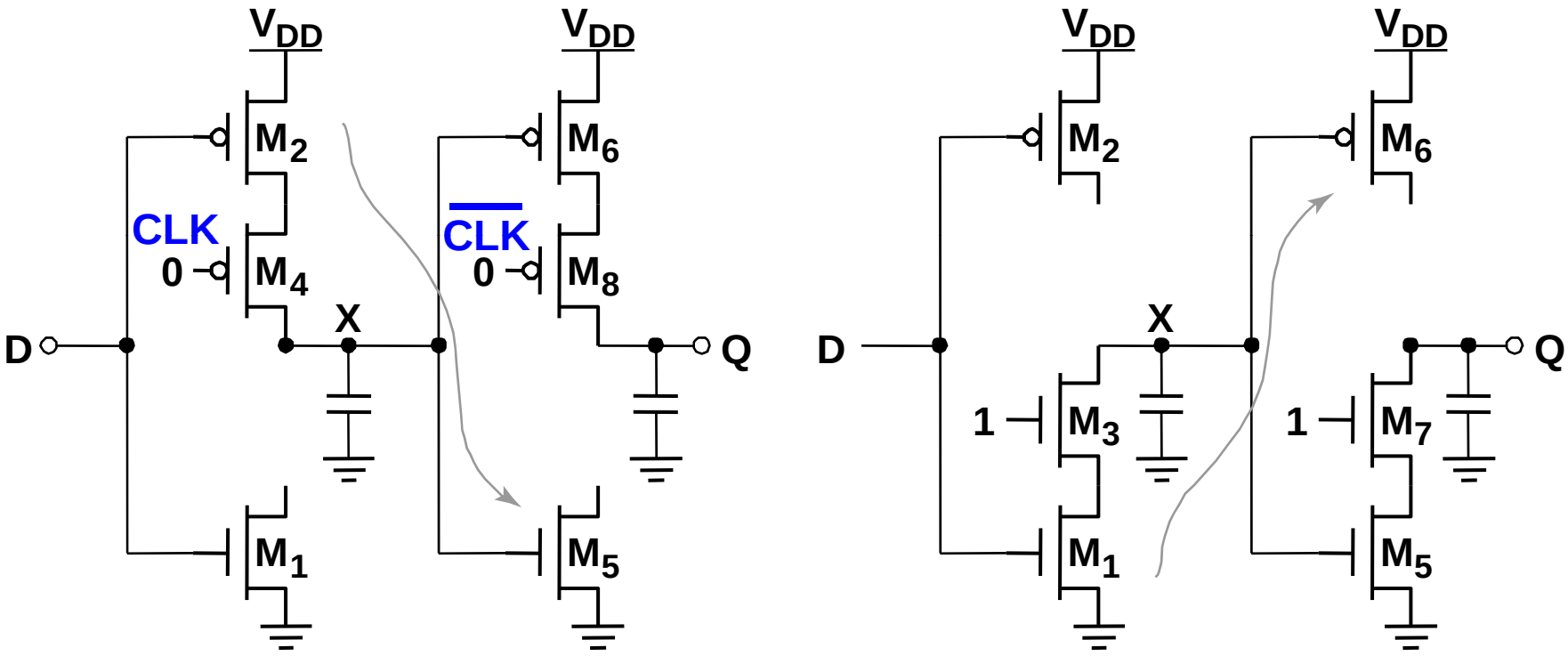
Clocked CMOS Latch



Slightly … Slower

Slightly … Smaller

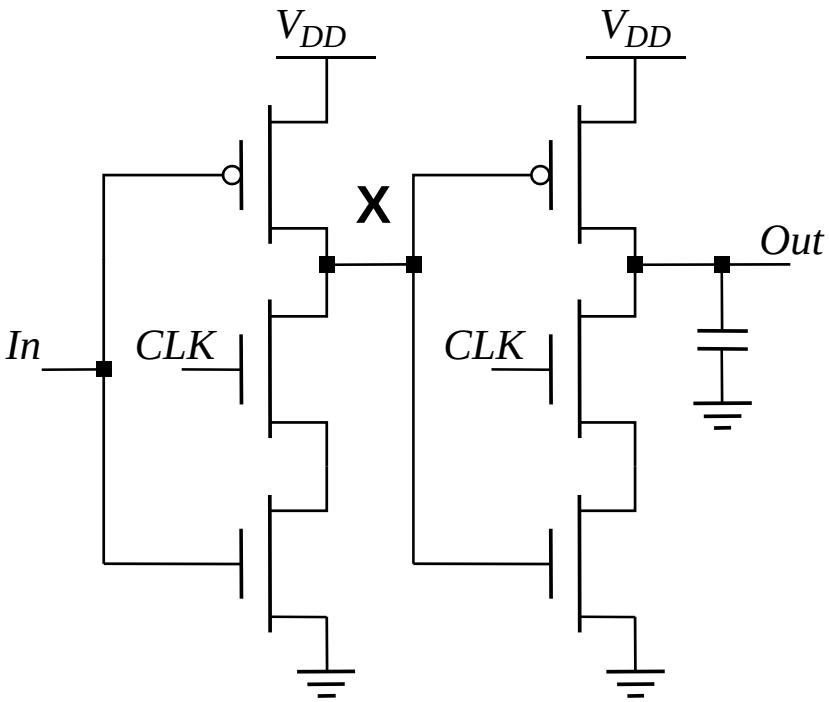
Insensitive to Clock-Overlap

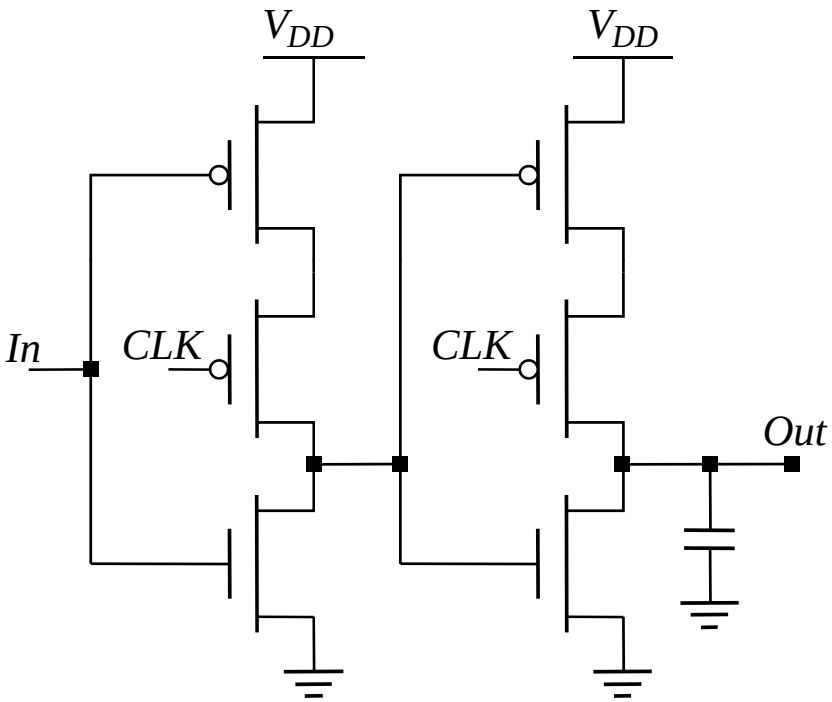


Output always decoupled from input, even with overlap

Other Latches/Registers: TSPC

True Single Phase Clocking



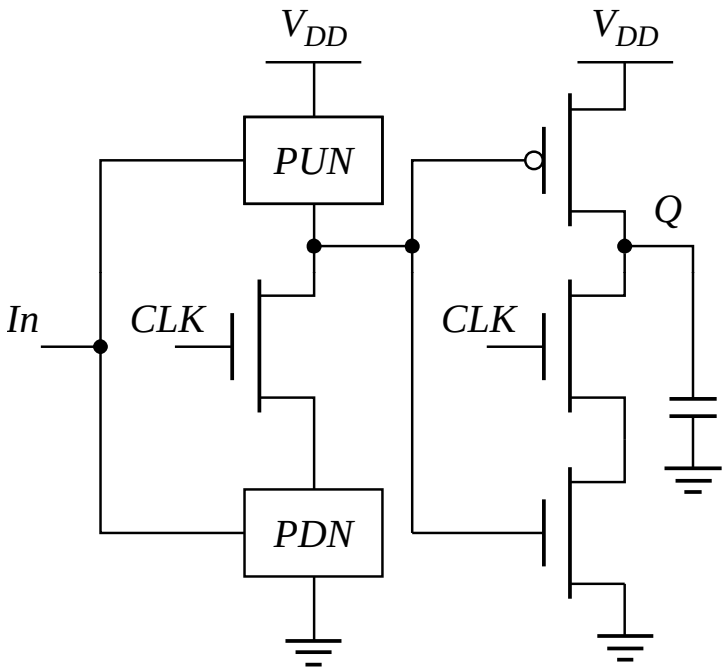


Positive latch
(transparent when CLK= 1)

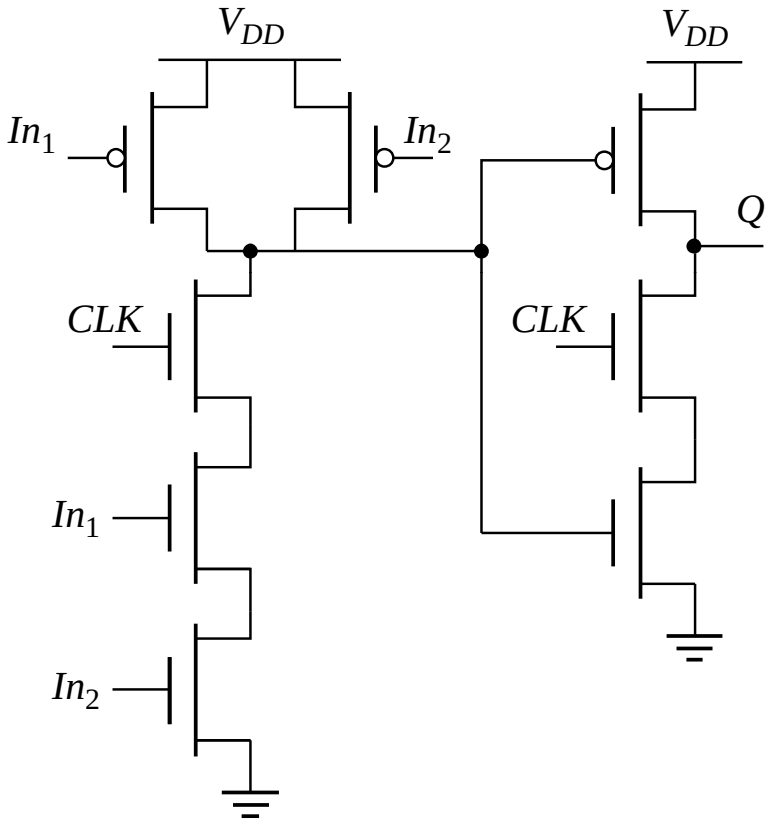
Negative latch
(transparent when CLK= 0)

TSPC register: positive and negative latch in cascade

Including Logic in TSPC

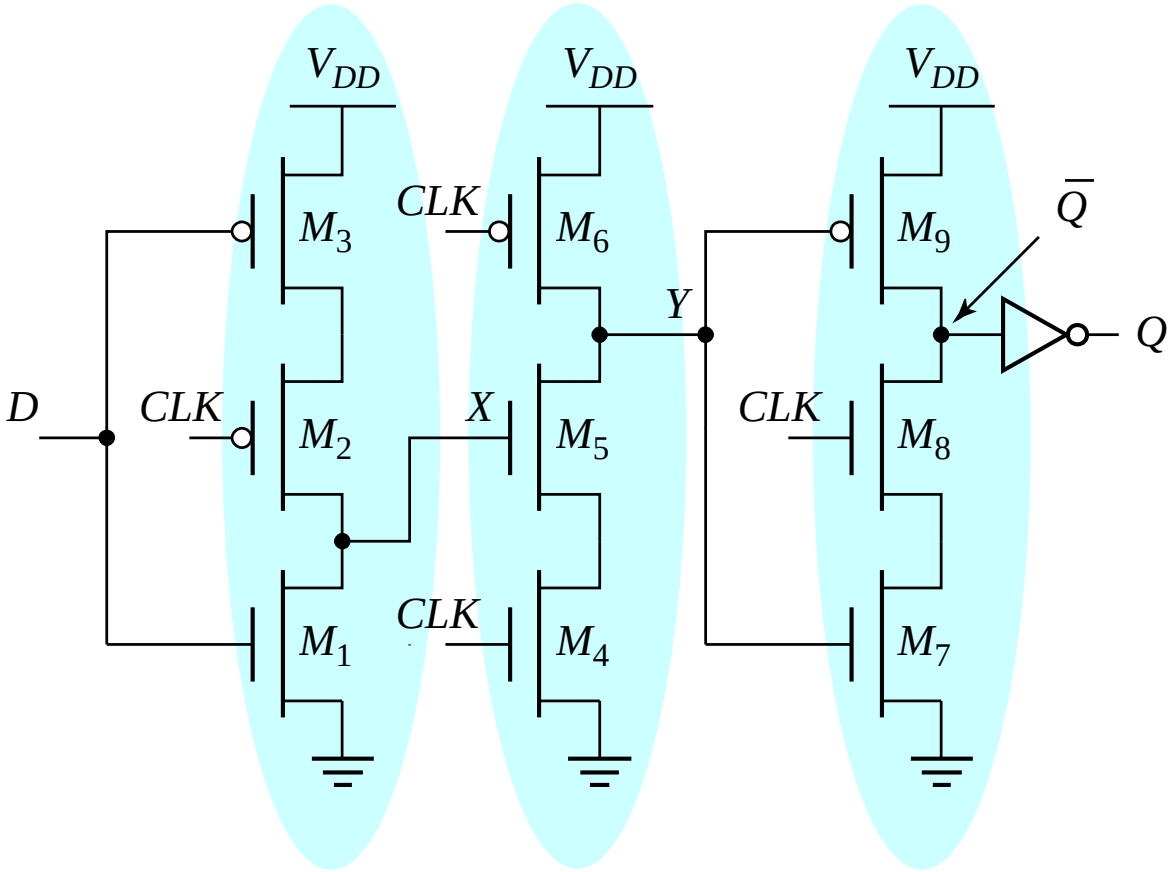


Example: logic inside the latch



AND latch

Alternative TSPC Register



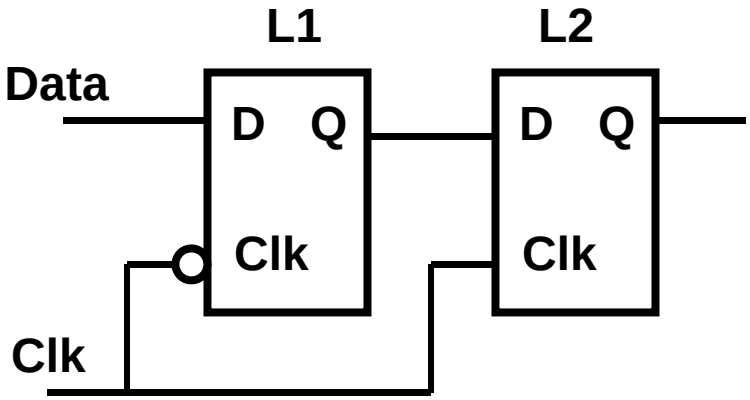
Clk = 0	D sampled on X	Precharge Y to VDD	Y ↑ doesn't make Q' ↓
Clk ↑		Evaluate based on X	
Clk = 1	Falling X can't charge Y	Y is stable	Y sampled on Q

Pulse-Triggered Latches
An Alternative Approach

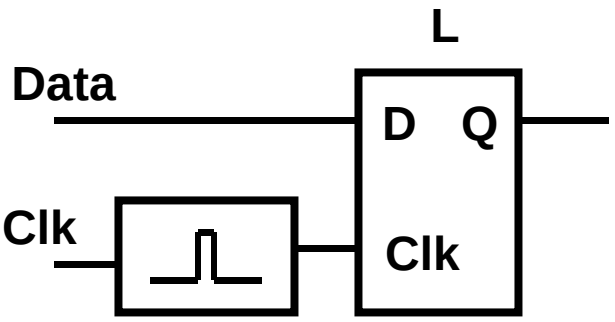
Remember: Use registers to avoid race-around problem.

Two latches in cascade: master-slave

Alternative: Pulse triggered latches

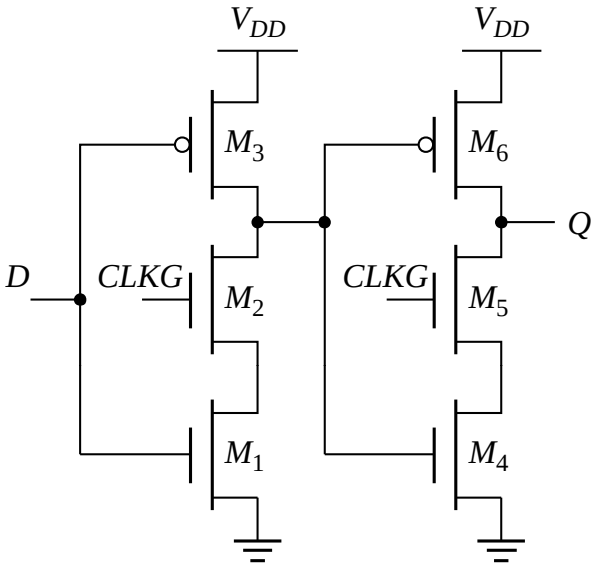


Master-Slave Latches

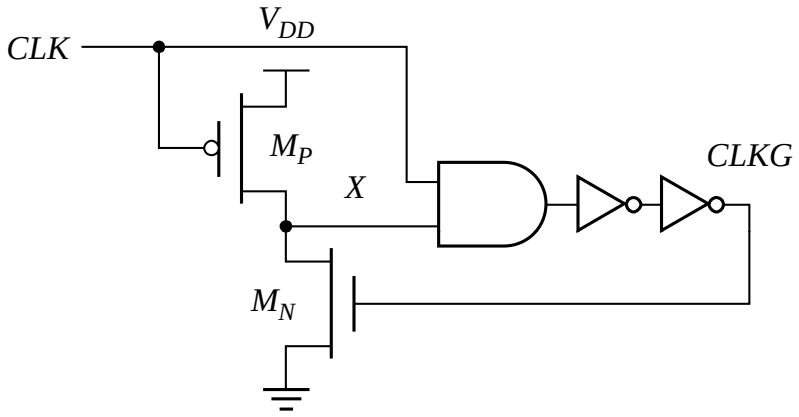


Pulse-Triggered Latch

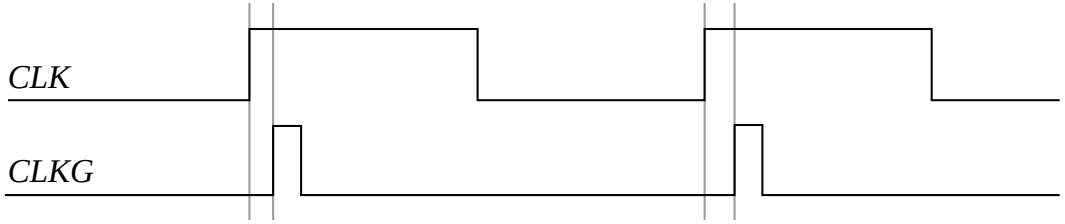
Pulsed Latches



(a) register



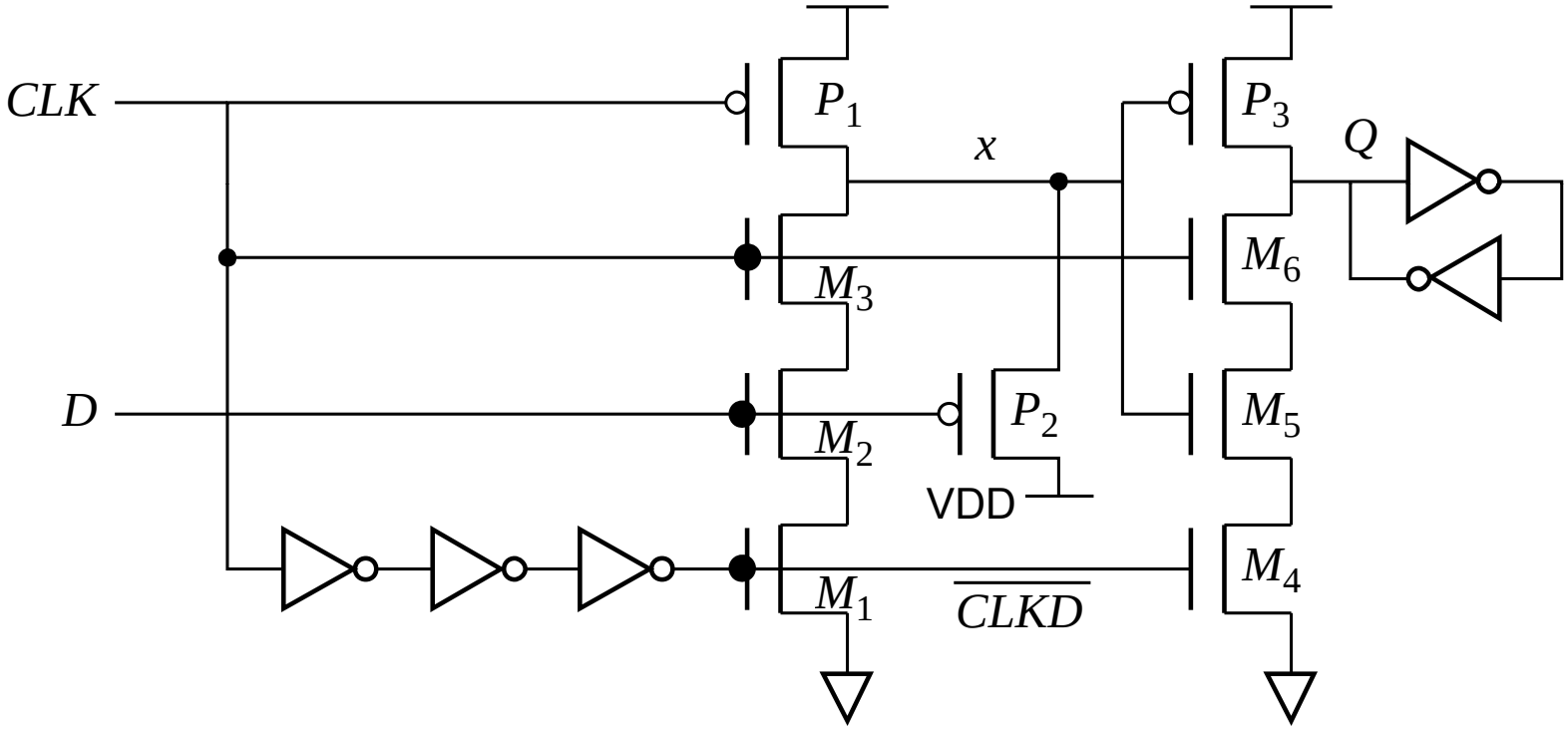
(b) glitch generation



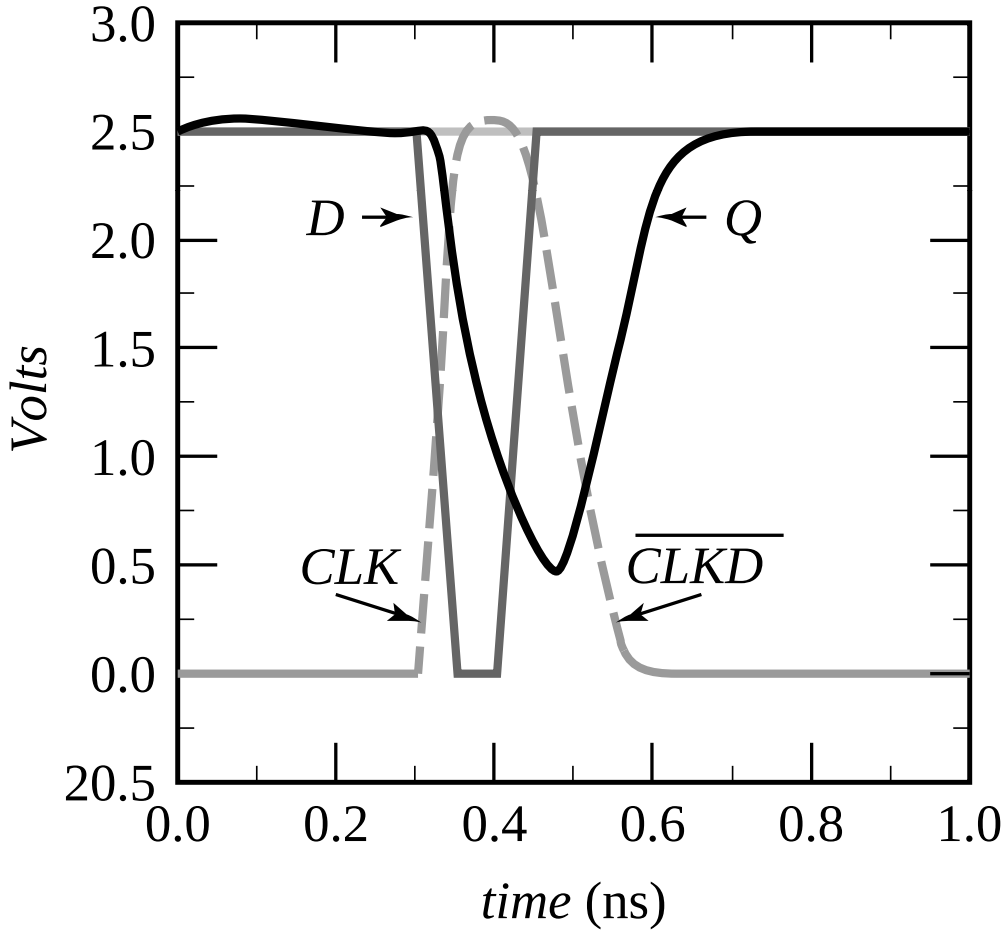
(c) glitch clock

Pulsed Latches

Hybrid Latch – Flip-flop (HLFF), AMD K-6 and K-7 :



Hybrid Latch-FF Timing



More Topics

- **Reset and enable inputs**
- **Scan-enabled CSEs**
- **Sense-amplifier based CSEs**
- **Double-edge triggered**
- **Low-leakage sleep mode**
- **(Ultra) Low voltage**
- **Soft-error / SEU tolerance**

- **Timing of CSEs**

Summary

- **Background**
  - **Timing, terminology, classification**
- **Static CSEs**
  - **Latches**
  - **Registers**
- **Dynamic CSEs**
  - **Latches**
  - **Registers**