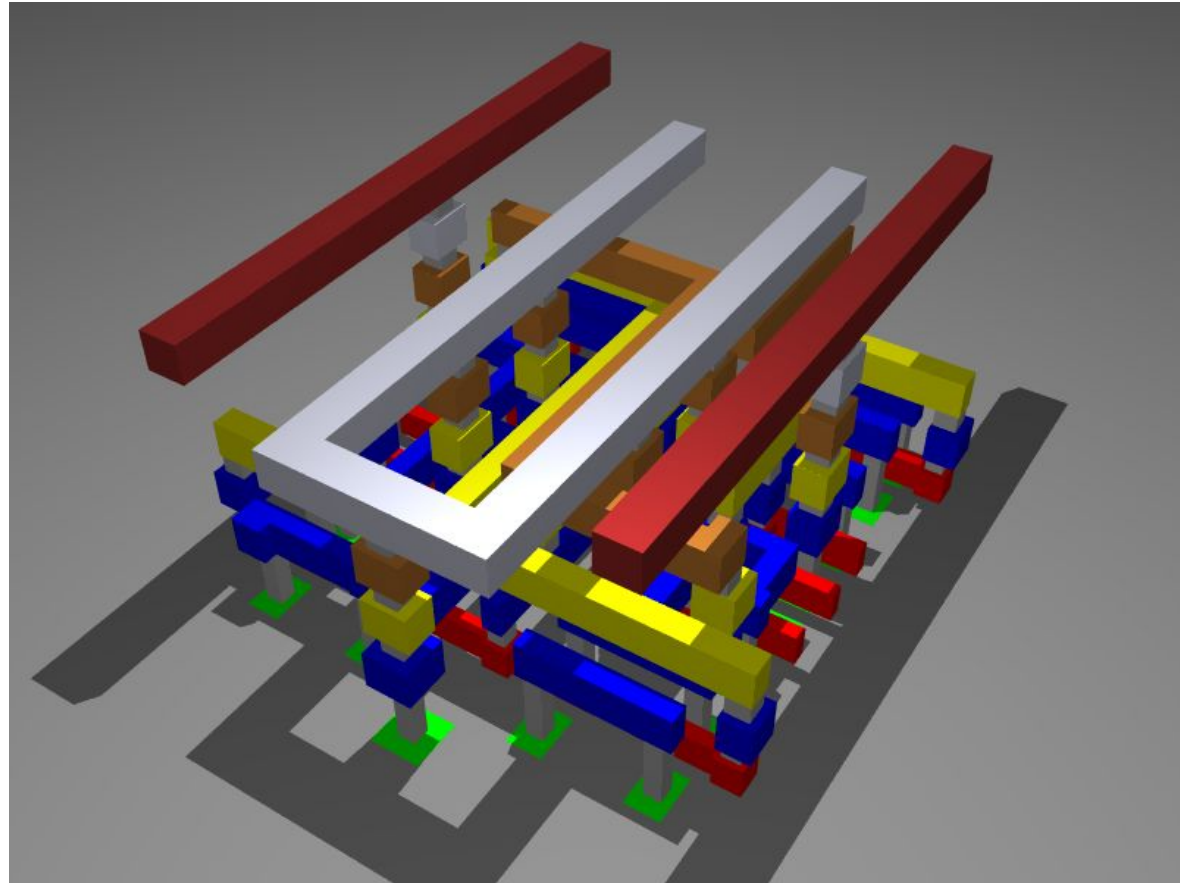


## ***MODULE 6***



## ***SEQUENTIAL ELEMENTS***

<b>P</b>	<b>7.1</b>	<b>Introduction</b>	<b>326 – 327</b>
<b>P</b>	<b>7.1.1</b>	<b>Timing Metrics for Sequential Circuits</b>	<b>327 – 328</b>
<b>P</b>	<b>7.1.2</b>	<b>Classification of Memory Elements</b>	<b>328 – 330</b>
	<b>7.2</b>	<b>Static Latches and Registers</b>	<b>330</b>
<b>P</b>	<b>7.2.1</b>	<b>The Bistability Principle</b>	<b>330 – 332</b>
<b>P</b>	<b>7.2.2</b>	<b>Multiplexer-Based Latches</b>	<b>332 – 333</b>
<b>P</b>	<b>7.2.3</b>	<b>Master-Slave Edge-Triggered Register</b>	<b>333 – 335</b>
<b>I</b>		<b>Timing Properties of Multiplexer-Based Master ....</b>	<b>335 – 339</b>
<b>O</b>	<b>7.2.4</b>	<b>Low-Voltage Static Latches</b>	<b>339 – 341</b>
<b>P</b>	<b>7.2.5</b>	<b>Static SR Flip-Flops — Writing Data by Pure Force</b>	<b>341 – 344</b>
<b>I</b>	<b>7.3</b>	<b>Dynamic Latches and Registers</b>	<b>344</b>
<b>I</b>	<b>7.3.1</b>	<b>Dynamic Transmission-Gate Edge-Triggered Registers</b>	<b>344 – 346</b>
<b>O</b>	<b>7.3.2</b>	<b>C2MOS — A Clock-Skew Insensitive Approach</b>	<b>346 – 350</b>
<b>O</b>	<b>7.3.3</b>	<b>True Single-Phase Clocked Register (TSPCR)</b>	<b>350 – 354</b>
<b>O</b>	<b>7.4</b>	<b>Alternative Register Styles</b>	<b>354 – 358</b>
<b>P</b>	<b>7.5</b>	<b>Pipelining: An Approach to Optimize Sequential Circuits</b>	<b>358 – 360</b>
<b>P</b>	<b>7.5.1</b>	<b>Latch versus register based pipelines</b>	<b>360</b>
<b>O</b>	<b>7.5.2</b>	<b>NORA-CMOS A logic style for pipelined circuits</b>	<b>361 – 363</b>
	<b>7.6</b>	<b>Nonbistable Sequential Circuits</b>	
<b>P</b>	<b>7.6.1</b>	<b>The Schmitt Trigger</b>	<b>364 – 367</b>
<b>O</b>	<b>7.6.2</b>	<b>Monostable Sequential Circuits</b>	<b>367 – 368</b>
<b>I</b>	<b>7.6.3</b>	<b>Astable Circuits</b>	<b>368 – 370</b>
<b>O</b>	<b>7.7</b>	<b>Perspective: Choosing a Clocking Strategy</b>	<b>370 – 371</b>
<b>P</b>	<b>7.8</b>	<b>Summary</b>	<b>371 – 372</b>

## § 7.5 Will be discussed with module 08 timing design

# Sequential Elements - Outline

## ■ Background

- Timing, terminology, classification

## ■ Static Flipflops

- Latches
- Registers

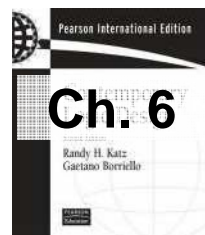
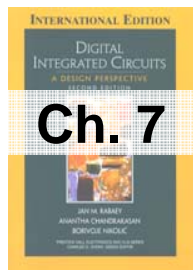
## ■ Dynamic Flipflops

- Latches
- Registers

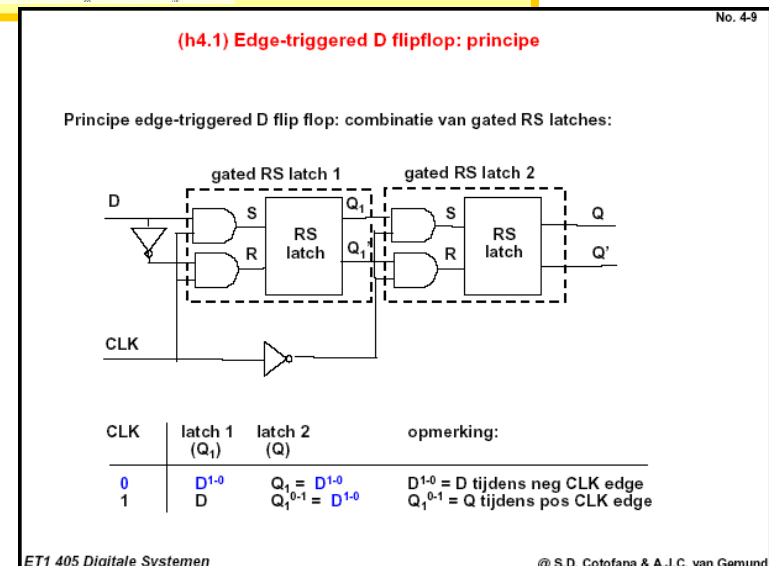
## ■ Non-bistable elements

- Schmitt Trigger

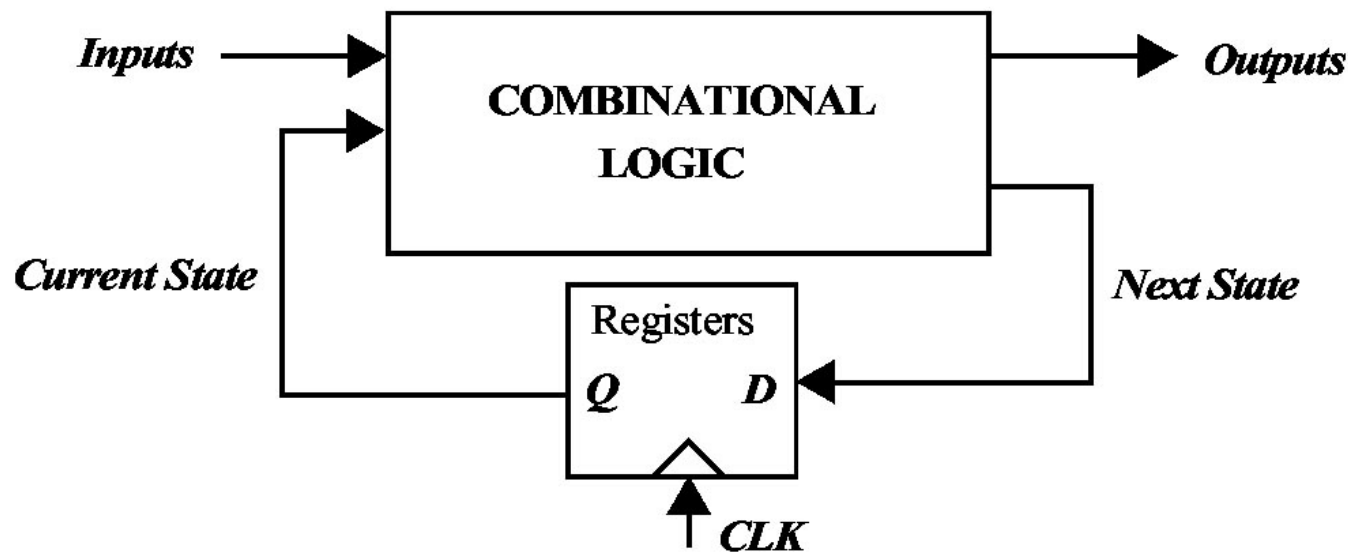
- Much of this material has already been covered in **DS1 (v Genderen), Katz Section 6.1**
- Here we will add **transistor-level** implementation, **dynamic flipflops**



[Sorin Cotofana]



# FSM with Positive Edge Triggered Registers



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

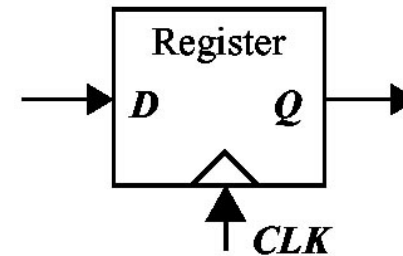
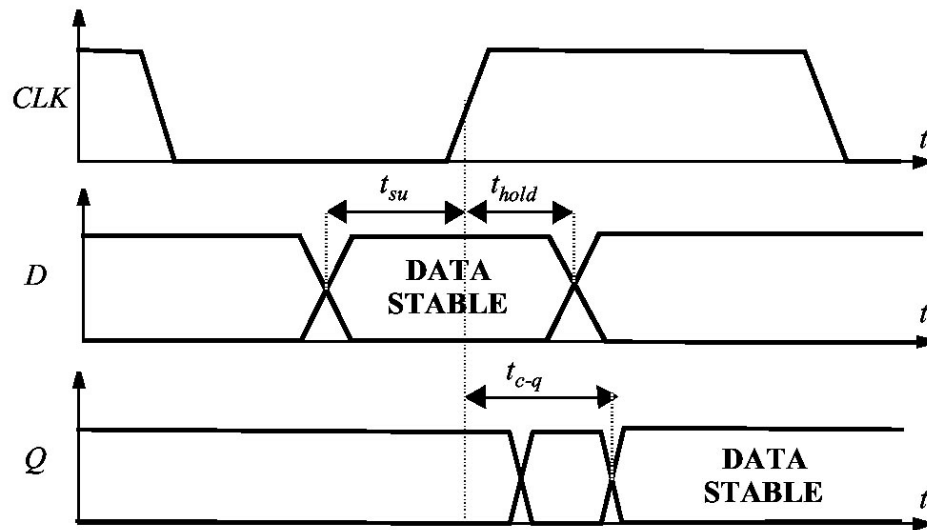
# Memory elements

- Store a **temporary value**, remember a **state**
- 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\\_\(electronics\)](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.**

# Timing Metrics Reminder



$t_{c-q}$  : 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

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

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

# Nomenclature

## Beware for confusion

	Katz (CS1)	Rabaey
Latch	Level sensitive storage element	Level sensitive storage element
Register	Group of storage elements	Edge triggered storage element
Flip Flop	Edge triggered storage element	Bistable element using feedback

$$CLK = CK = \phi$$



# Latches vs. Registers

## 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**

# 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)

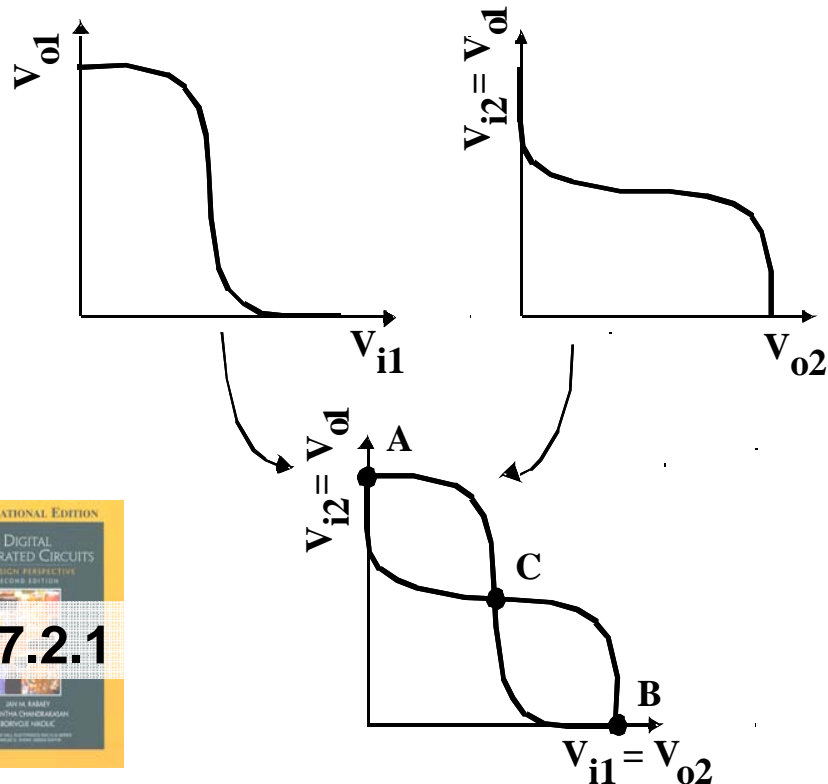
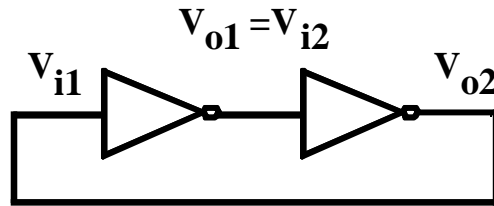
**Faster, smaller**

Rabaey: bistable elements are called Flip Flops

# Static Latches and Registers

- Latches → can be gated or not
- Registers

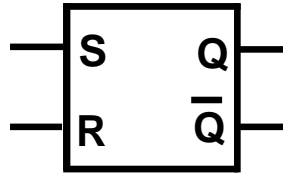
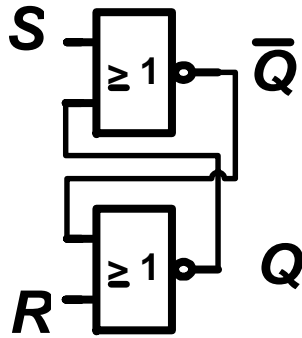
# Positive Feedback: Bi-Stability



- |Loop-gain| in A,B  $\ll 1$
- A,B: **stable** points
- |Loop-gain| in C  $\gg 1$
- C: **meta-stable** point

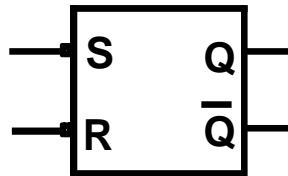
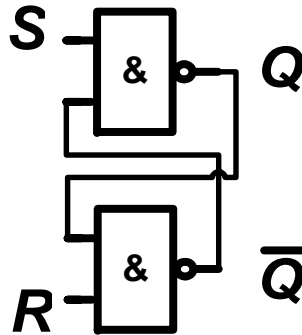


# SR-Latch



S	R	Q	Q̄
0	0	Q	Q̄
1	0	1	0
0	1	0	1
1	1	0	0

← forbidden

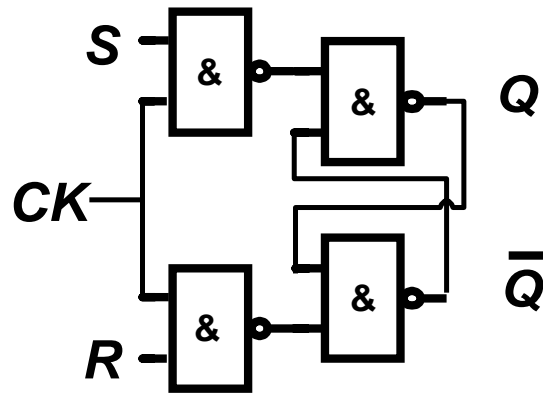


S	R	Q	Q̄
1	1	Q	Q̄
0	1	1	0
1	0	0	1
0	0	1	1

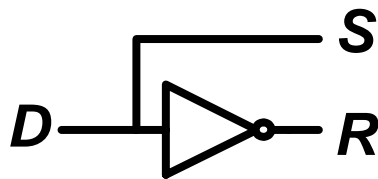
← forbidden



# Clocked SR-Latch

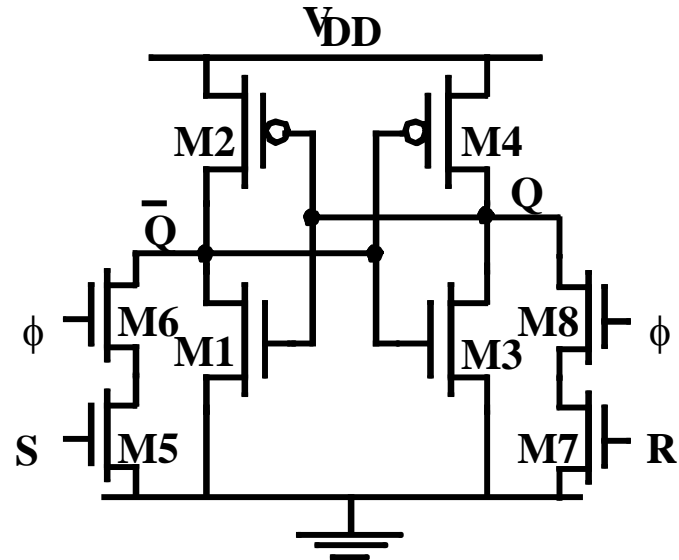
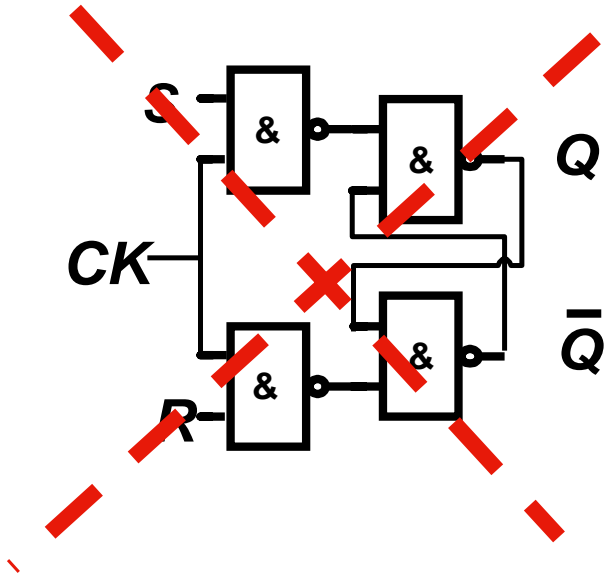


- Katz: gated latch
- **Positive** latch (active on CK high)
- Naïve implementation
- 16 transistors
- D latch requires **9xN, 9xP**
- Larger **area, cost, power**



- Construction of **D-latch**
- D-latch, D-register most **common** in **VLSI**

# CMOS Clocked SR-Latch



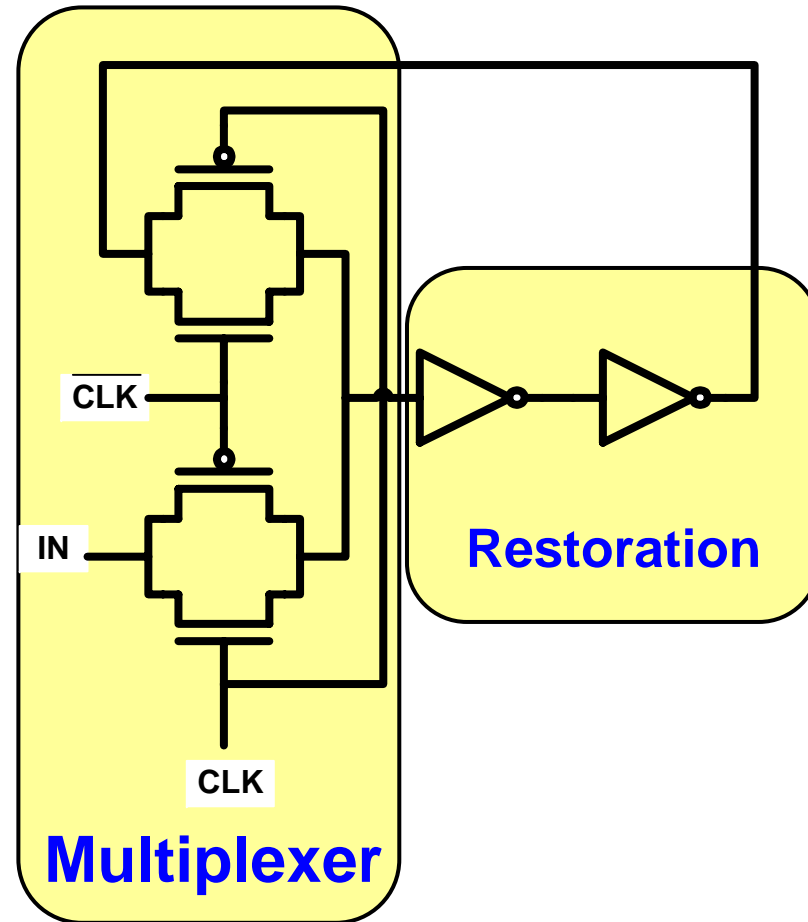
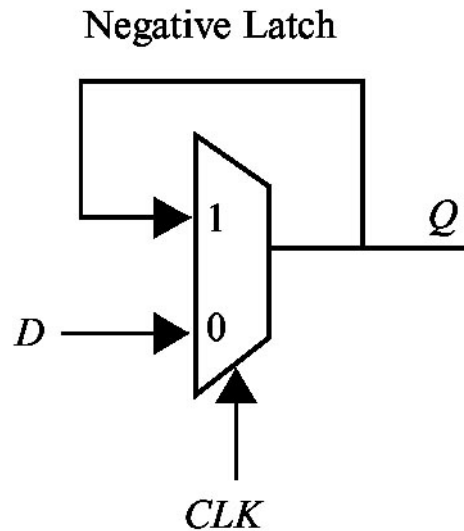
$$\phi = CK$$

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

**Q:** Is this a **ratioed** design or not?  
Does it consume static power?



# Multiplexer-Based Latches



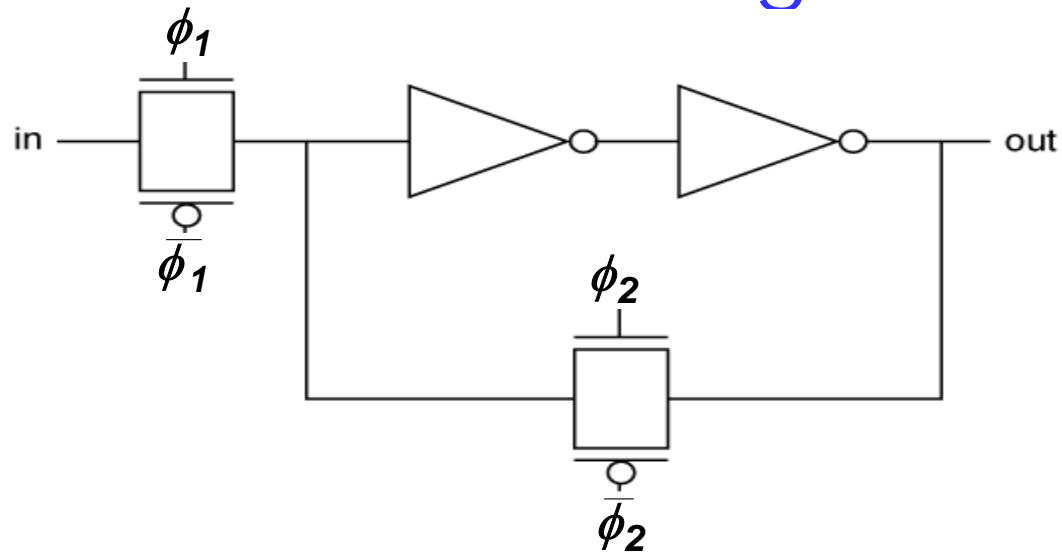
## Recirculating latch



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

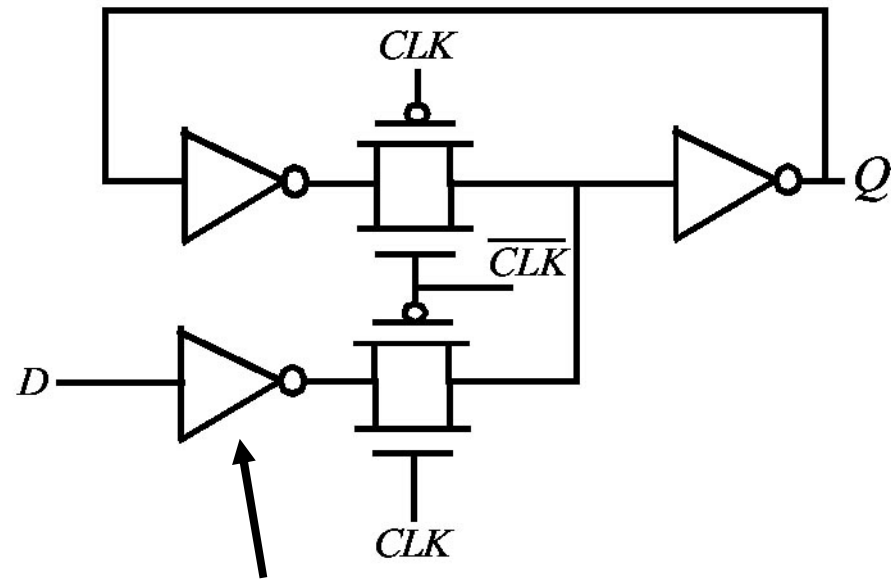
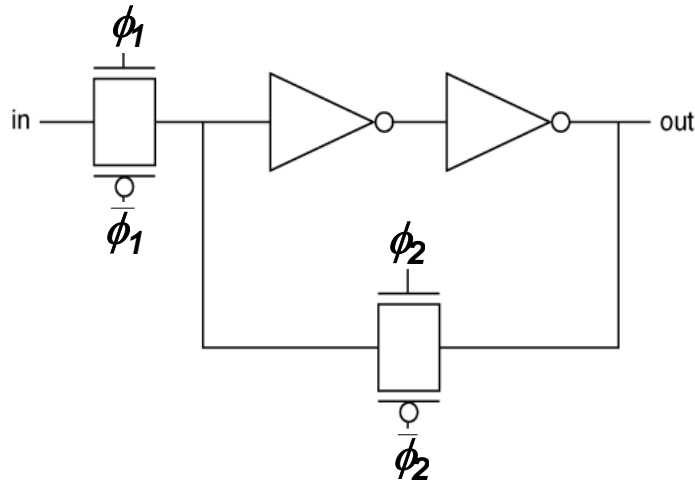


# Recirculating latch



- **Quasi-static**, static on one phase
- Feedback **restores** value
- Requires  $4 \times N$ ,  $4 \times P$ , minimum size  
(compare  $7 \times N$ ,  $3 \times P$ , non-minimum size)
- $\phi_1$  and  $\phi_2$  inverse but should be non-overlapping
- Can suffer from **charge sharing** (when  $\phi$  not non-overlapping)  
 $C_{in}$  and  $C_{load}$  form communicating vessels when Output connected directly to input

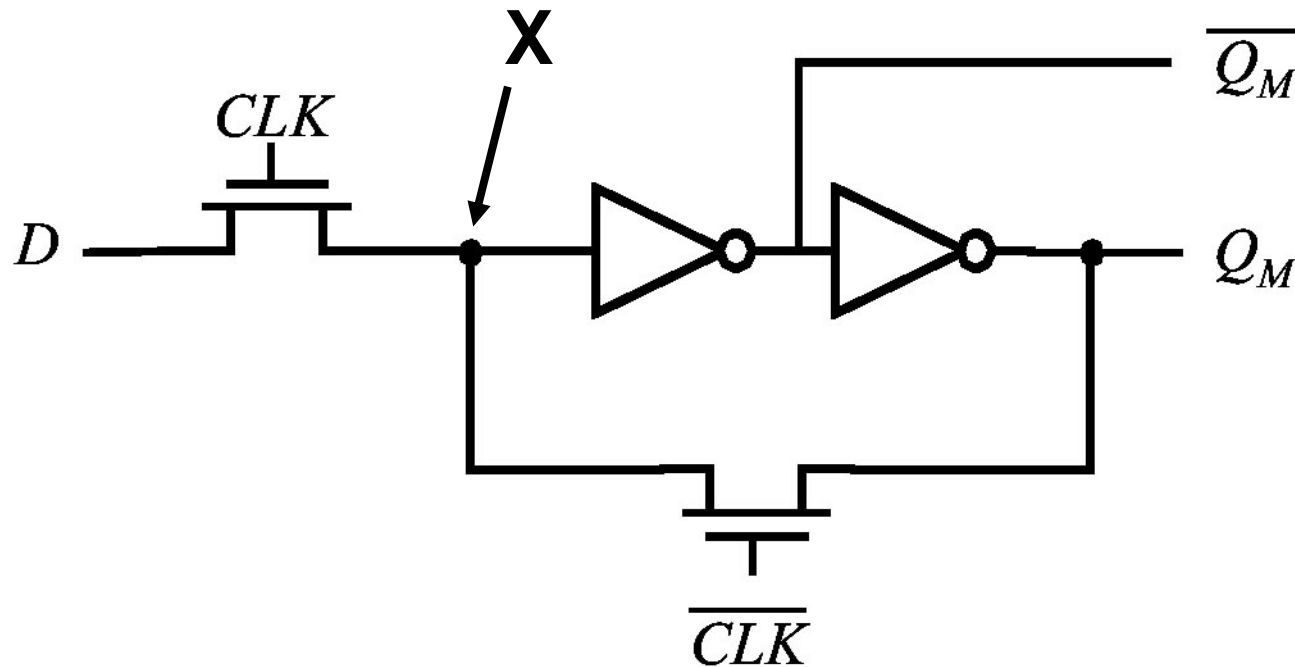
# In insensitive for Charge Sharing



Uni-directionality of this inverter prevents coupling between Q and D

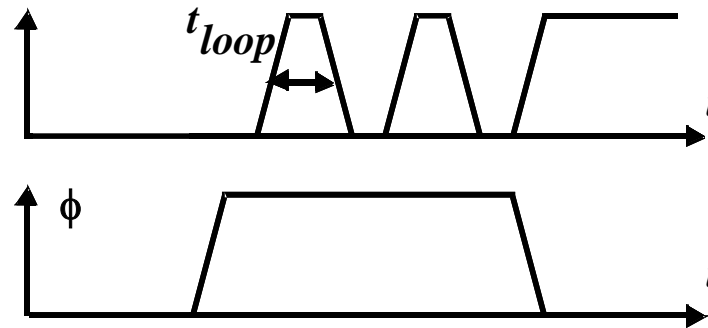
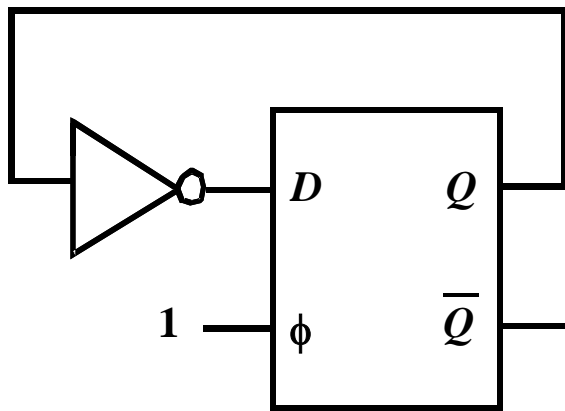
- Non ratioed
- High load to CLK

# Recirculating NMOS Latch.



- Degraded 1 at X
- Lower noise margin, higher delay, power

# Latch Designs can Suffer from Race Problems

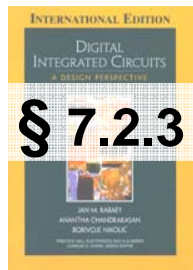
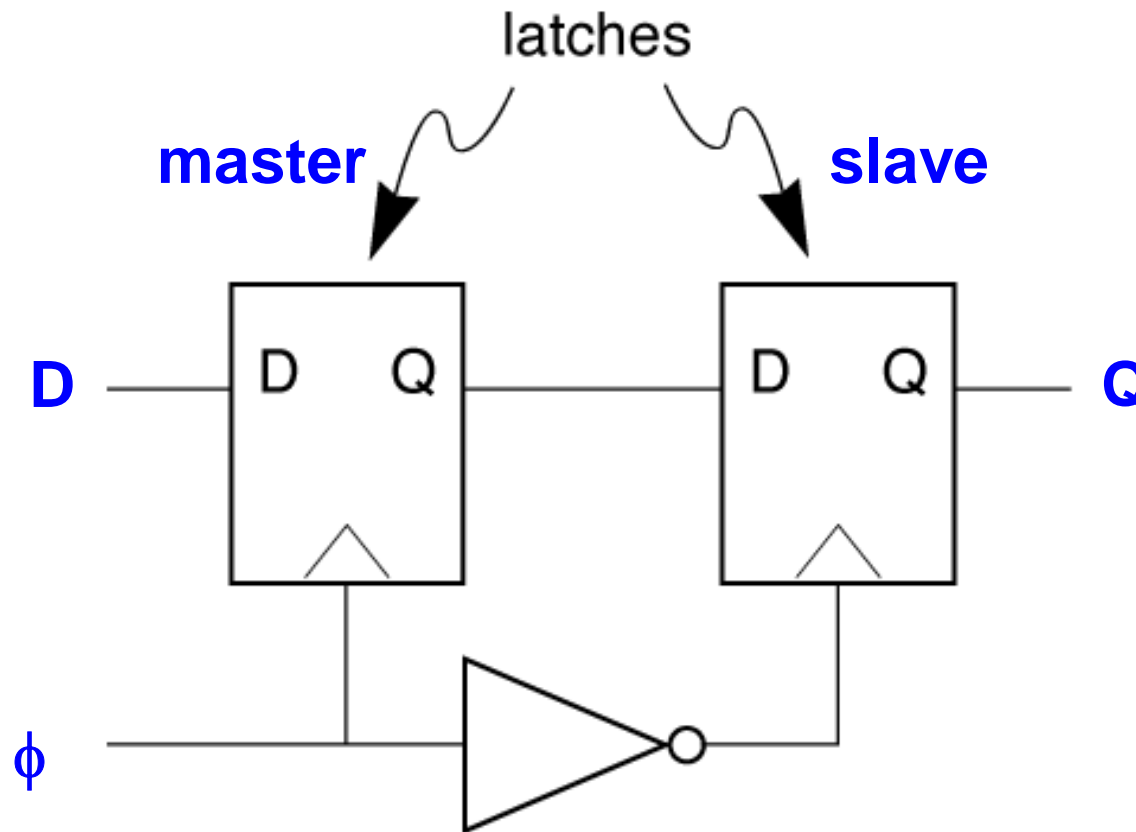


Signal can **race around** during  $\phi = 1$

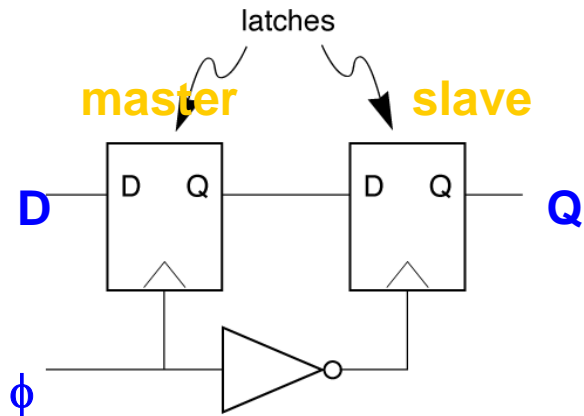


# Registers

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



# Master-slave operation



$\phi = 0$ :

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

$\phi = 1$ :

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

# CMOS Flip-Flop Construction

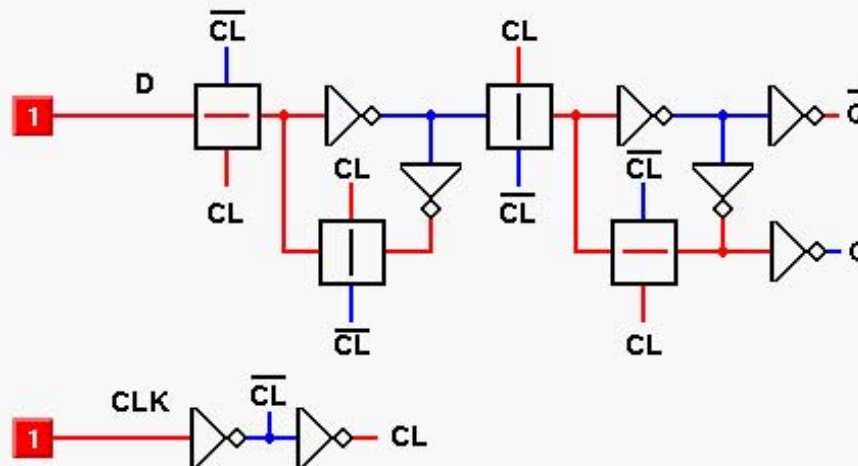
CMOS technology allows a very different approach to flip-flop design and construction. Instead of using logic gates to connect the clock signal to the master and slave sections of the flip-flop, a CMOS flip-flop uses *transmission gates* to control the data connections. (See the [CMOS gate electronics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

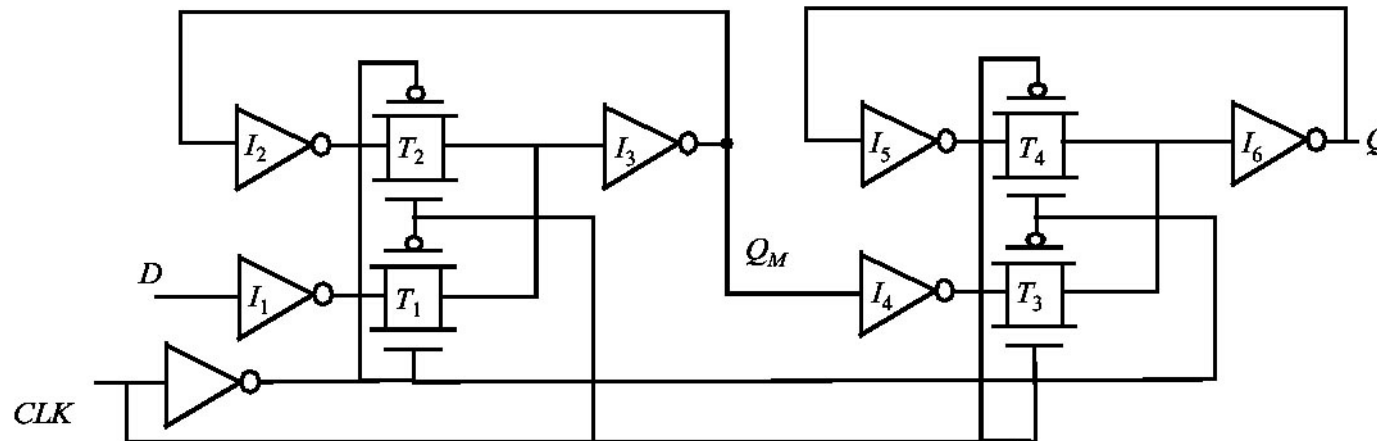
[www.play-hookey.com](http://www.play-hookey.com)

clickable



link

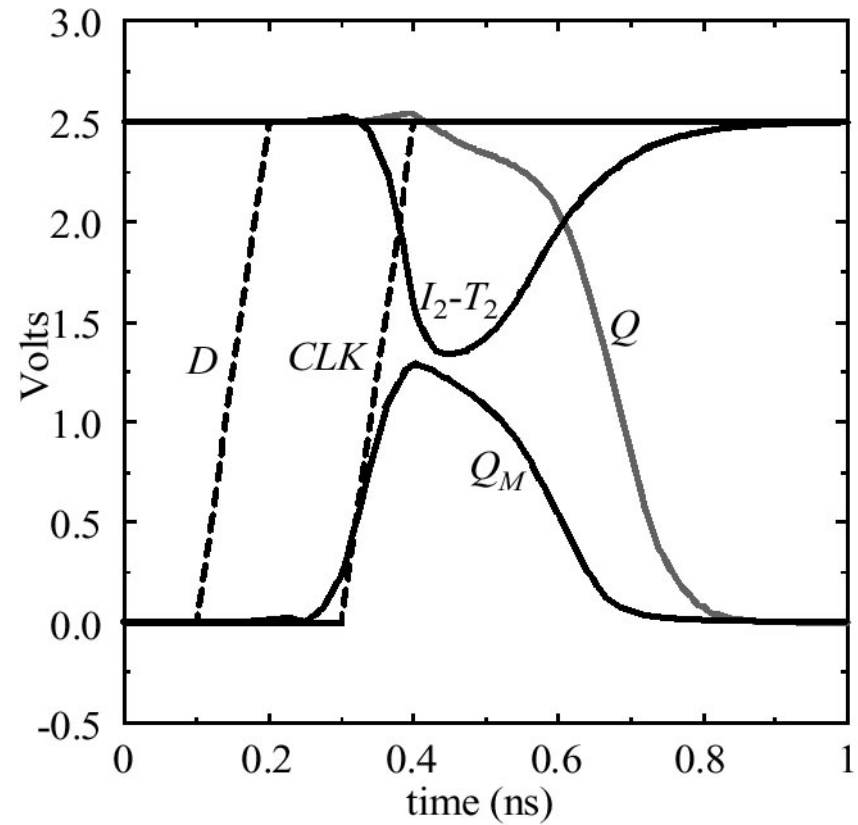
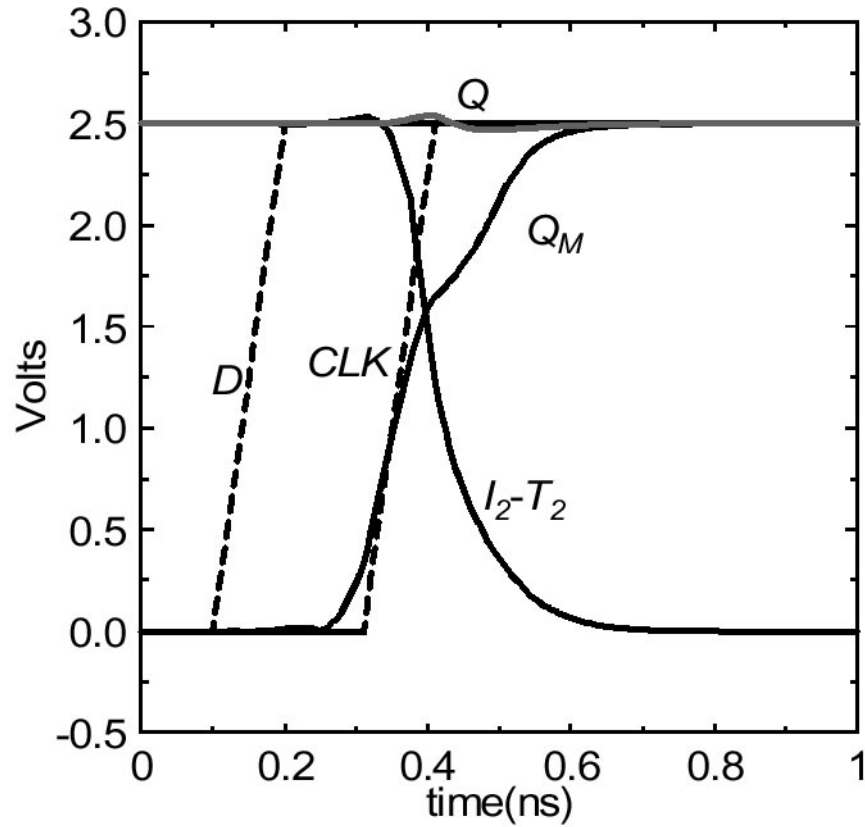
# Transistor Level Master Slave Positive Edge Triggered Register



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

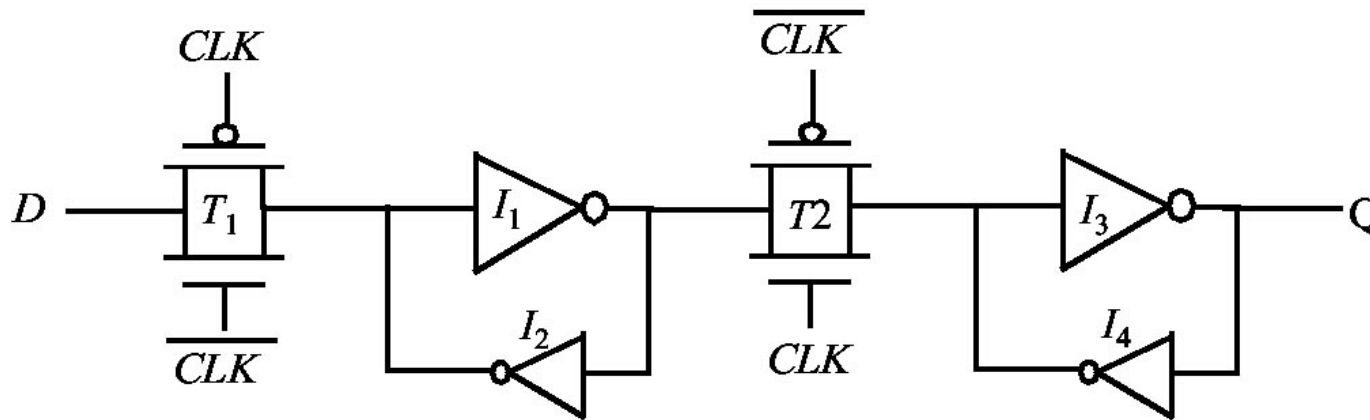
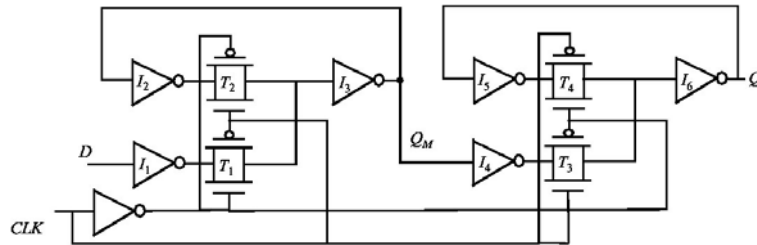


# Set-up Time Simulation



**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)

# Non-bistable Elements

## ■ Schmitt Trigger

Adobe Acrobat Professional - [schmitt\_trigger\_20070410[1].pdf]

File Edit View Document Tools Advanced Window Help

Open Save Print Email Search Create PDF Review & Comment Secure Sign Advanced Editing

2007/10/04 P-LAB Robot Project Kaichouhi, A

**Elektronische Schakelingen Exercise 6**  
**ET1505-D2, 2007**  
**The CMOS Standard and Schmitt-trigger Logic Inverters**

**Objectives:**

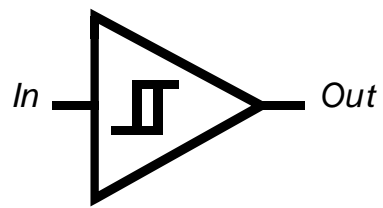
- Measuring characteristics of a Schmitt-trigger inverter and a standard CMOS logic inverter.
- Understanding why hysteresis of the Schmitt-trigger is useful when interfacing sensors to digital electronic circuits.

The first section of this exercise is an introduction to the Schmitt-trigger variant of this inverter. Concepts such as the voltage-transfer curve, threshold voltage, gain, logic levels, and noise margins are explained, and will be characterized experimentally. A photo-transistor experiment is used to test the behaviour of both standard and Schmitt logic inverter types.

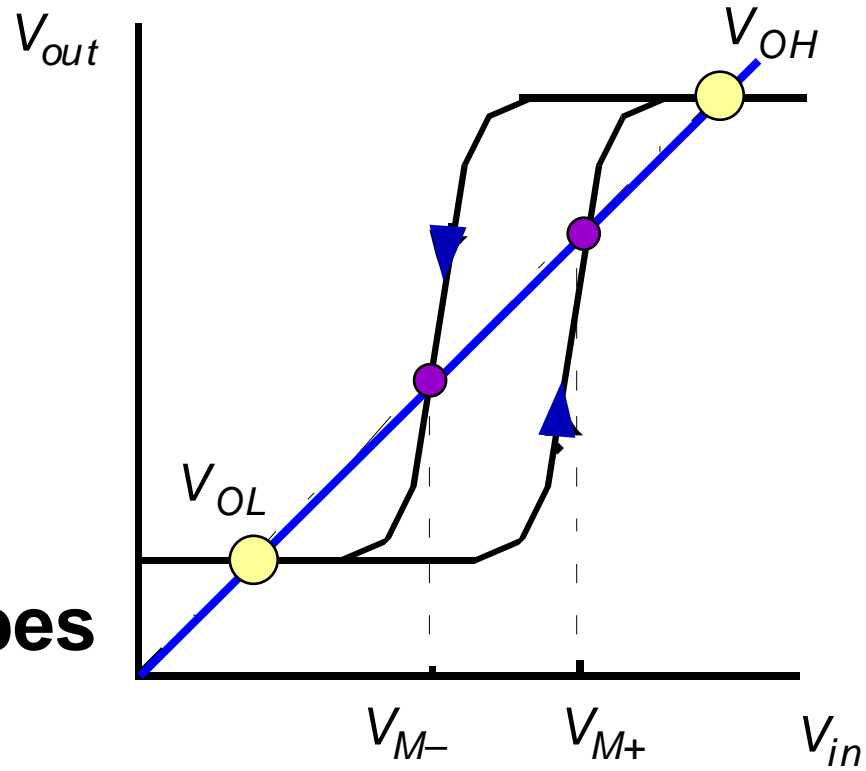
**Was discussed in P-lab**

215.9 x 279.4 mm 1 of 14

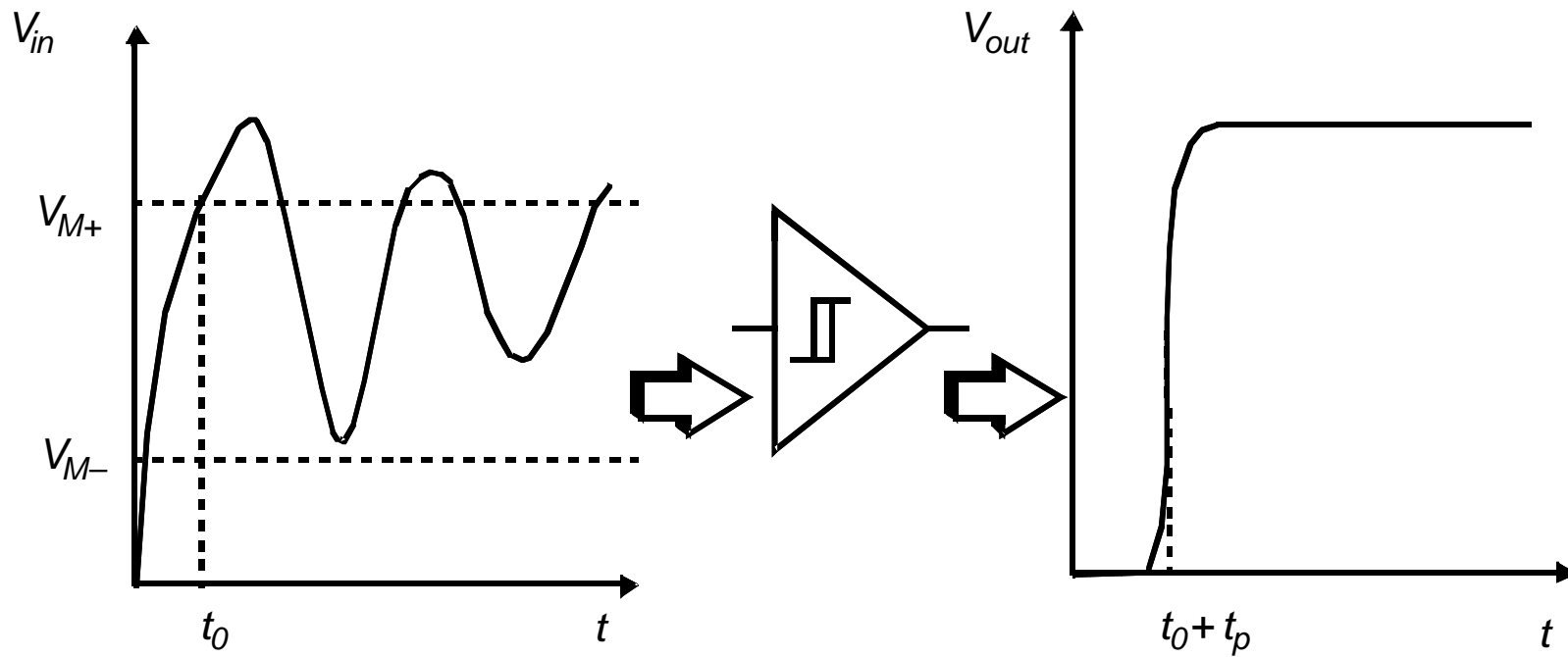
# Schmitt Trigger



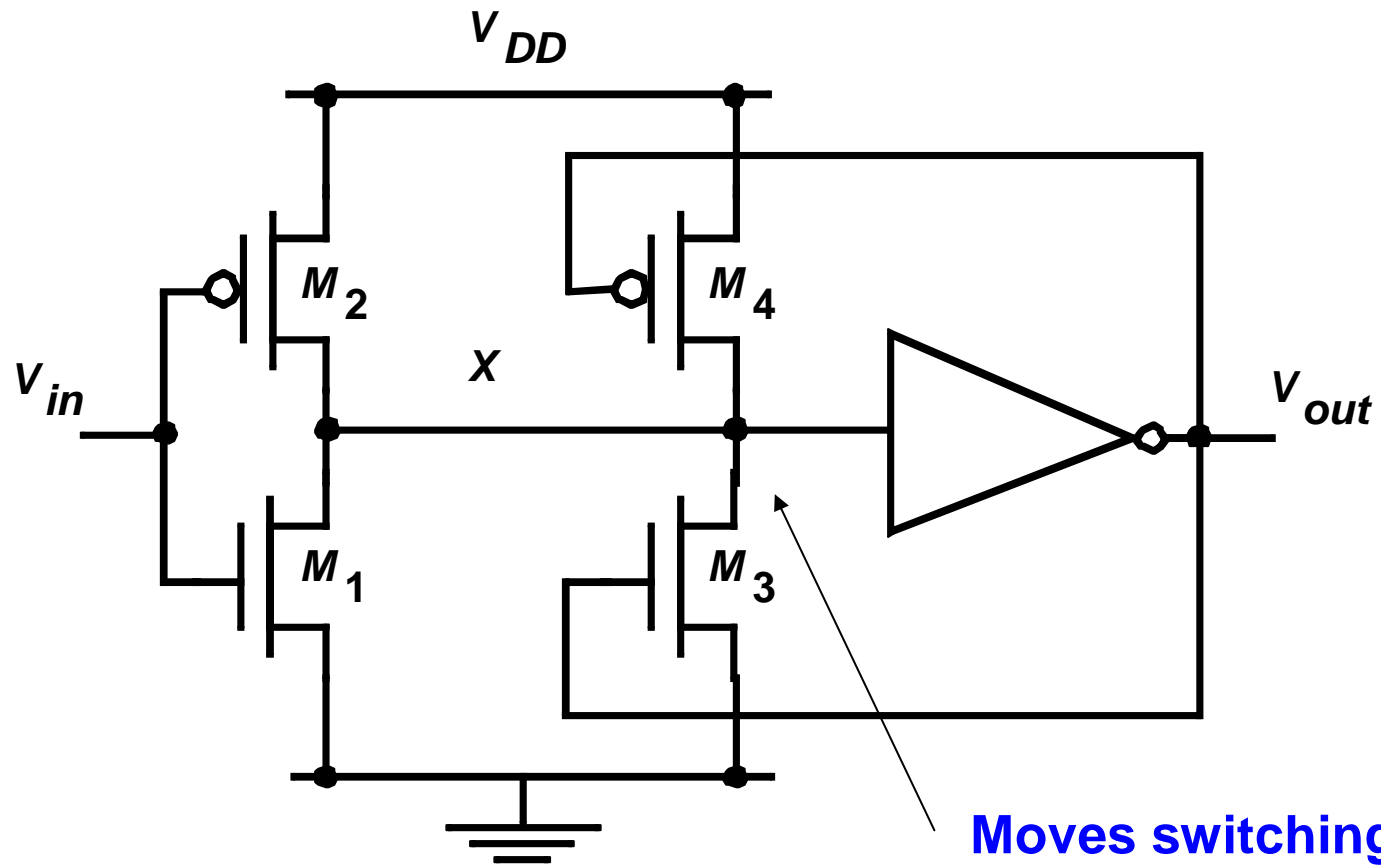
- VTC with hysteresis
- Restores signal slopes



# Noise Suppression using Schmitt Trigger

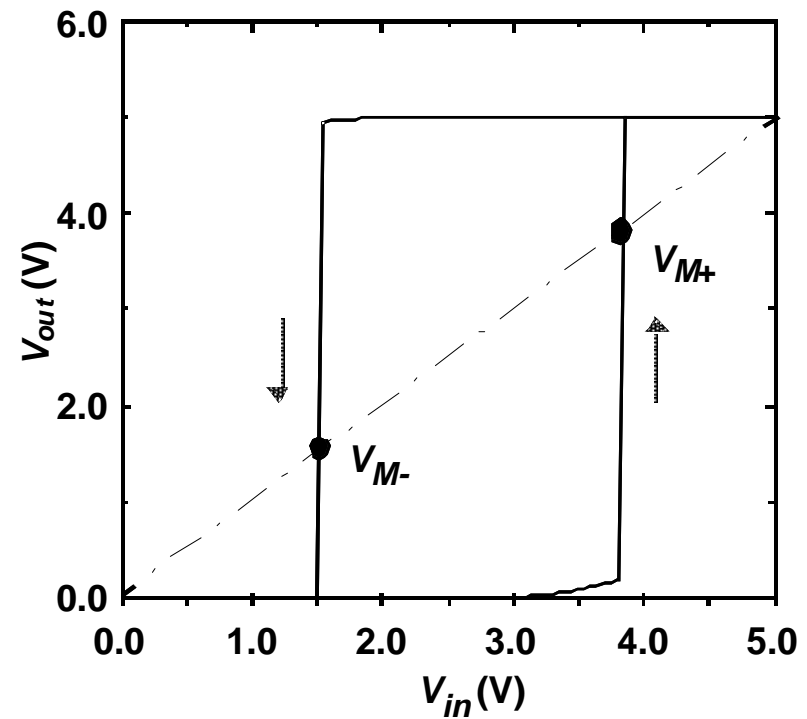
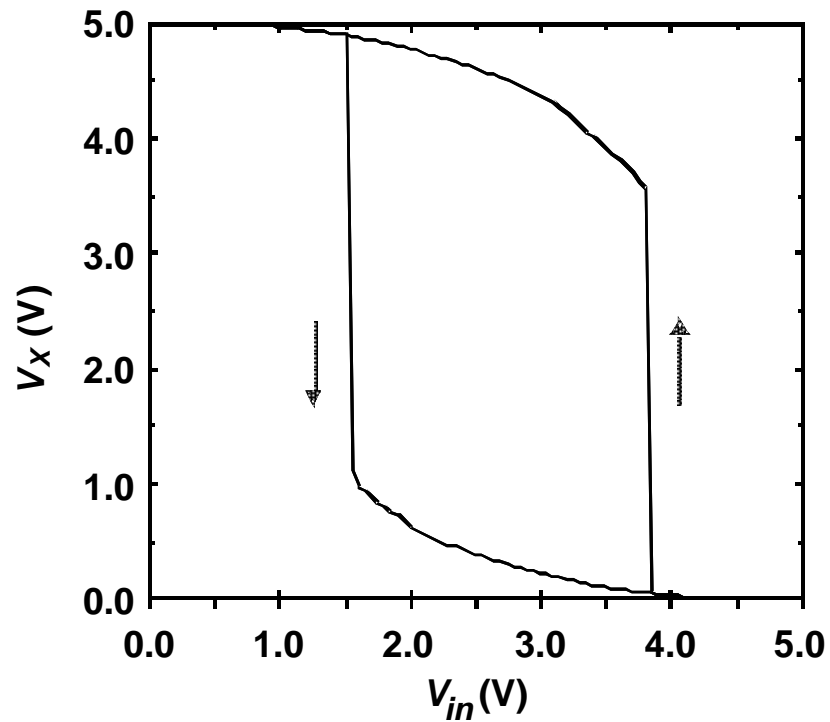


# CMOS Schmitt Trigger

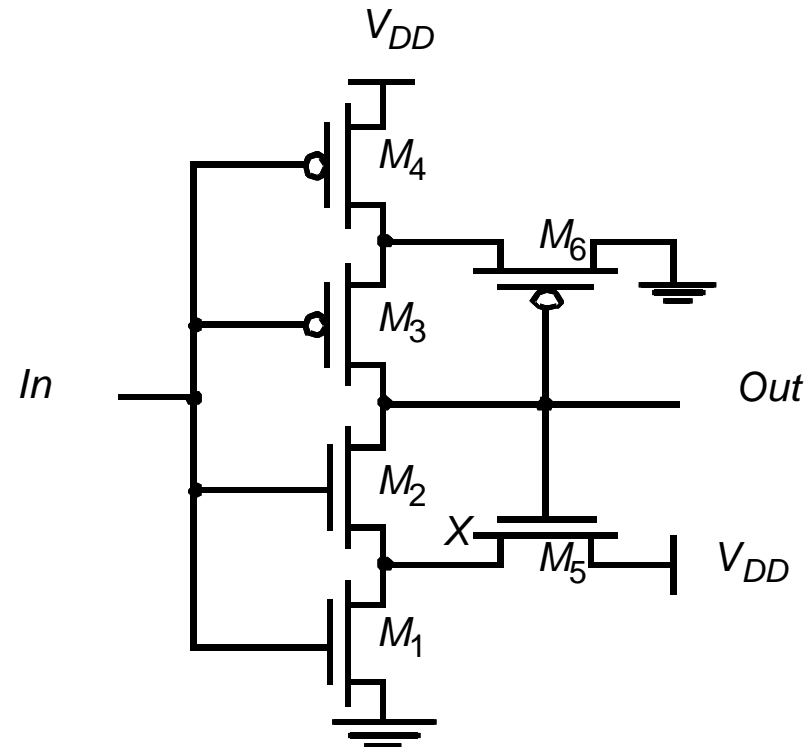


**Moves switching threshold  
of first inverter**

# Schmitt Trigger Simulated VTC



# CMOS Schmitt Trigger (2).





# Summary

- **Background**

- **Timing, terminology, classification**

- **Static Flipflops**

- **Latches**

- **Registers**

- **Dynamic Flipflops**

- **Latches**

- **Registers**

- **Non-bistable elements**

- **Schmitt Trigger**