Digital Integrated Circuits

Designing Sequential Circuits

Fuyuzhuo

Sequential Logic Circuits

- **Introduction**
- **Timing**
- **Static Latches and Registers**
- **Dynamic Latches and Registers**

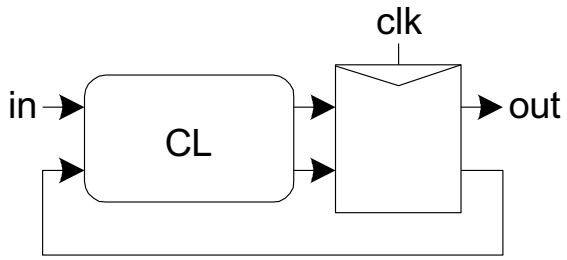
Sequential Logic Circuits

- ***Introduction***
- Timing
- Static Latches and Registers
- Dynamic Latches and Registers

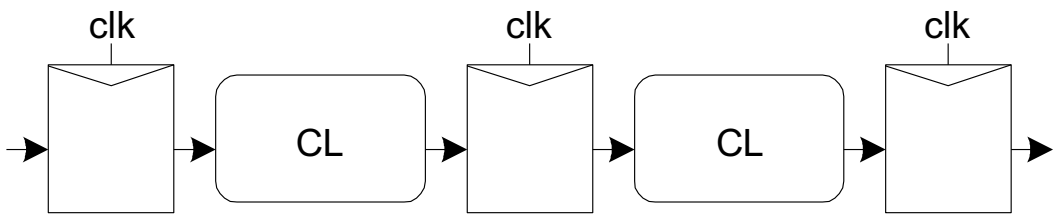
Comb. or ***Seq.***

Sequencing

- *Combinational logic*
  - output depends on current inputs
- *Sequential logic*
  - output depends on current and previous inputs
  - Requires separating previous, current, future
  - Called *state* or *tokens*
  - Ex: FSM, pipeline



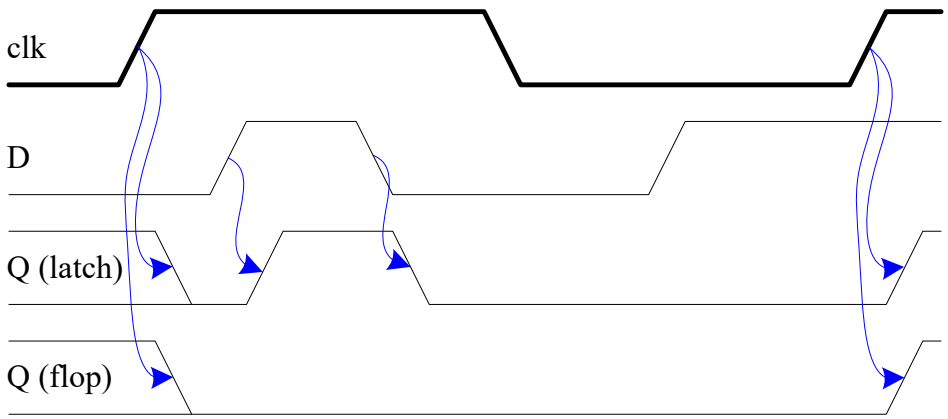
Finite State Machine



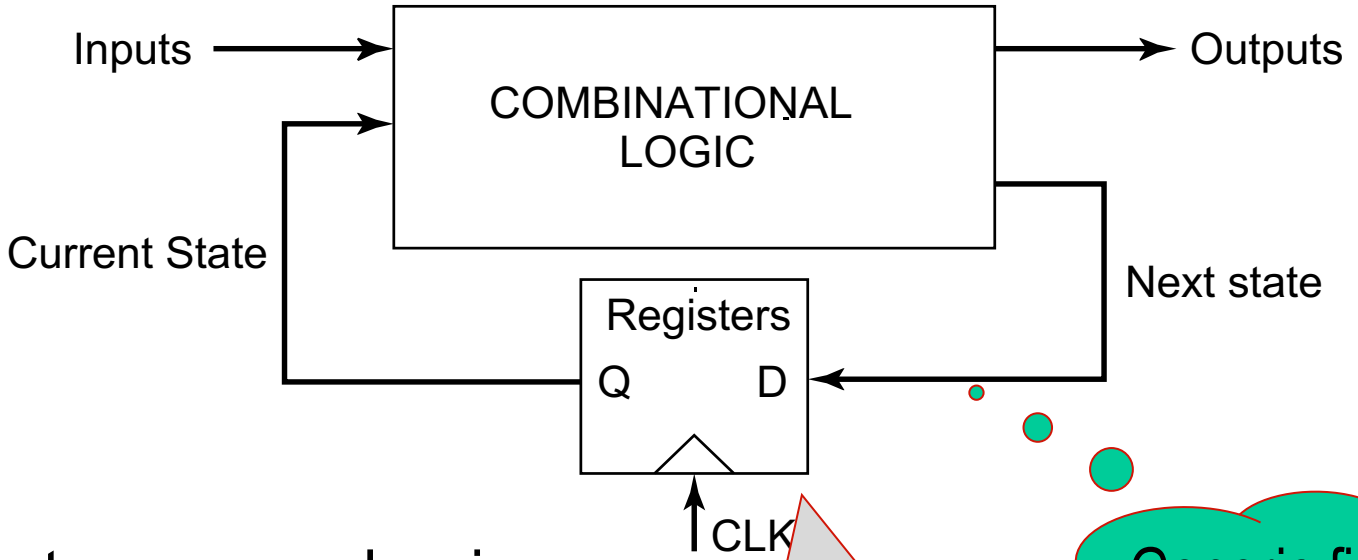
Pipeline

Sequencing Elements

- **Latch**: Level sensitive
  - a.k.a. transparent latch, D latch
- **Flip-flop**: edge triggered
  - A.k.a. master-slave flip-flop, D flip-flop, D register
- Timing Diagrams
  - Transparent
  - Opaque
  - Edge-trigger



Sequential Logic



2 storage mechanisms
- positive feedback
- charge-based

Generic finite-state machine

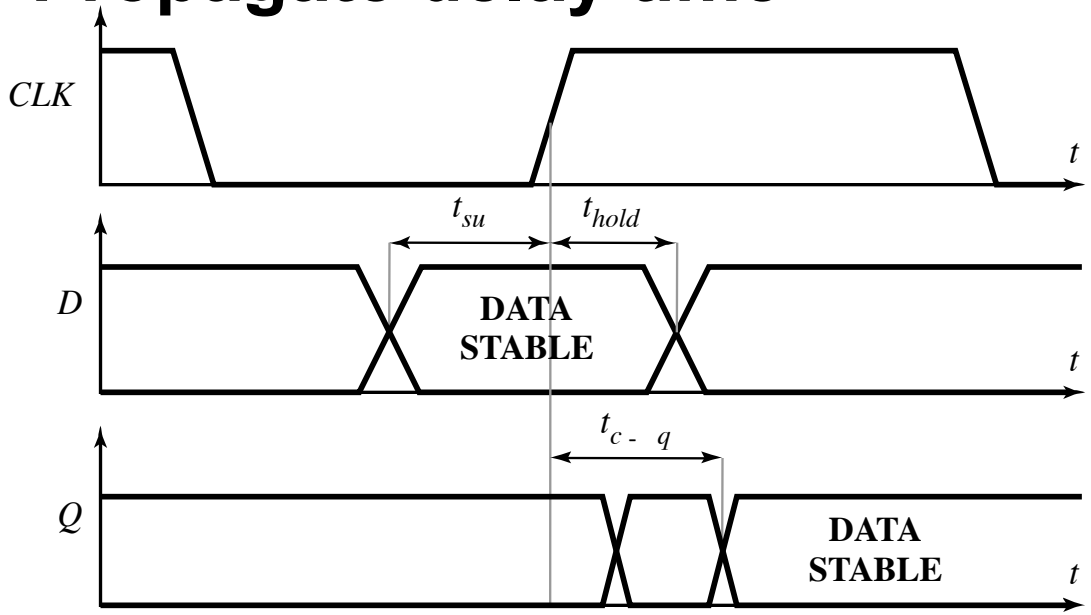
Synchronous,rising/positive edge triggered,falling edge/negative edge triggered

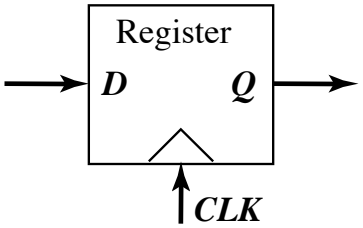
Timing Definitions

❶ **Setup time:** the time that the data inputs must be valid before clock transition

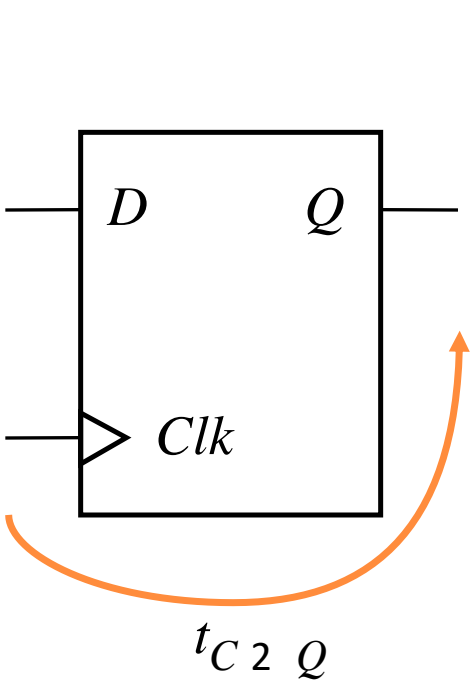
❷ **hold time:** the time that the data must remain valid after the clock transition

❸ **Propagate delay time**

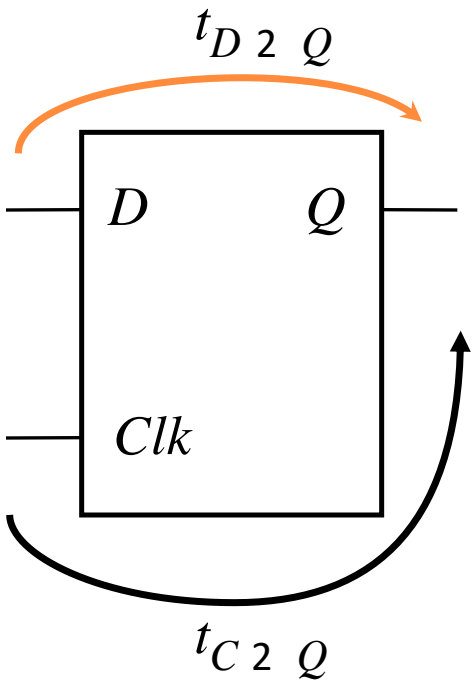




Characterizing Timing



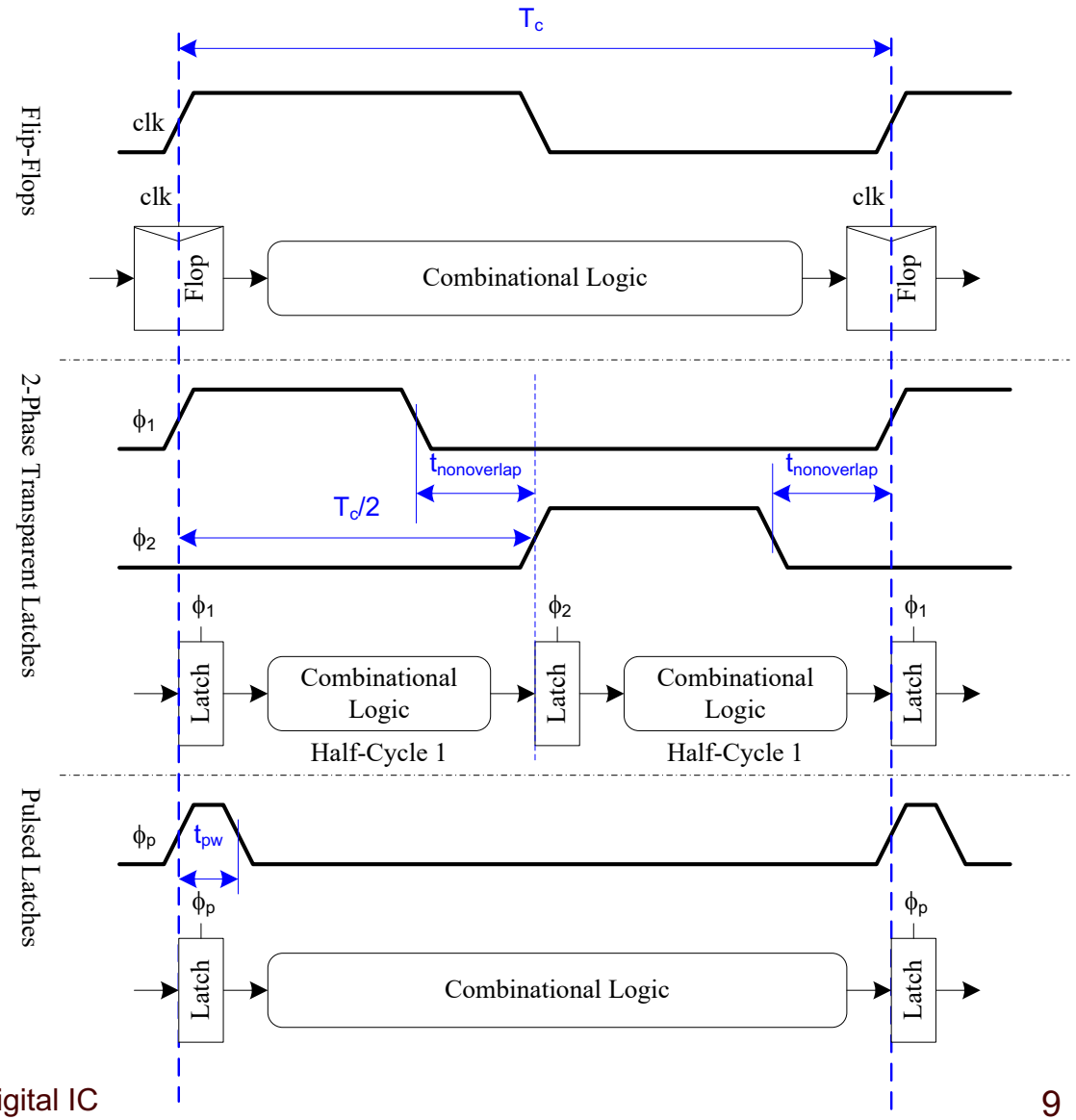
Register



Latch

Sequencing Methods

- Flip-flops
- 2-Phase Latches
- Pulsed Latches

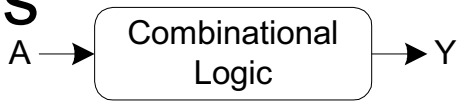


Design Sequential Logic Circuits

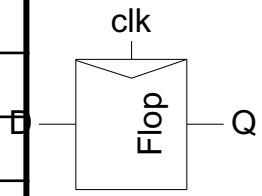
- Introduction
- ***Timing***
- Static Latches and Registers
- Dynamic Latches and Registers

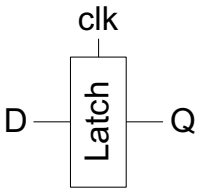
Timing Diagrams

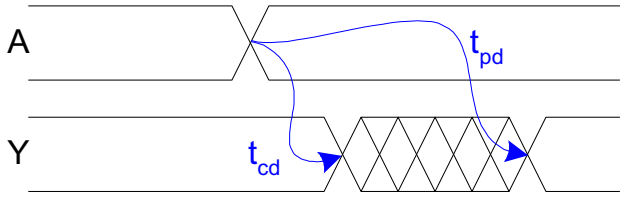
Contamination and **Propagation** Delays

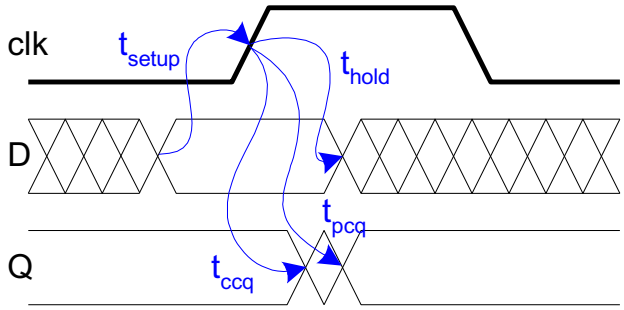


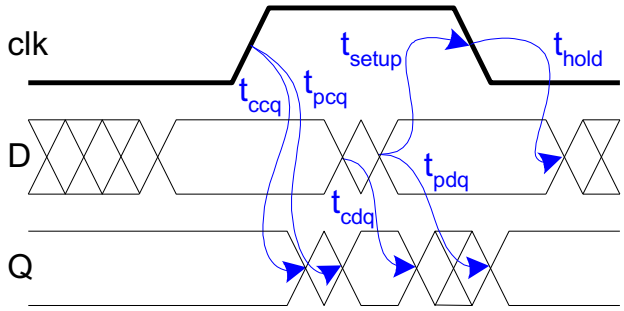
t_{pd}	Logic Prop. Delay
t_{cd}	Logic Cont. Delay
t_{pcq}	Latch/Flop Clk-Q Prop Delay
t_{ccq}	Latch/Flop Clk-Q Cont. Delay
t_{pdq}	Latch D-Q Prop Delay
t_{pcq}	Latch D-Q Cont. Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time





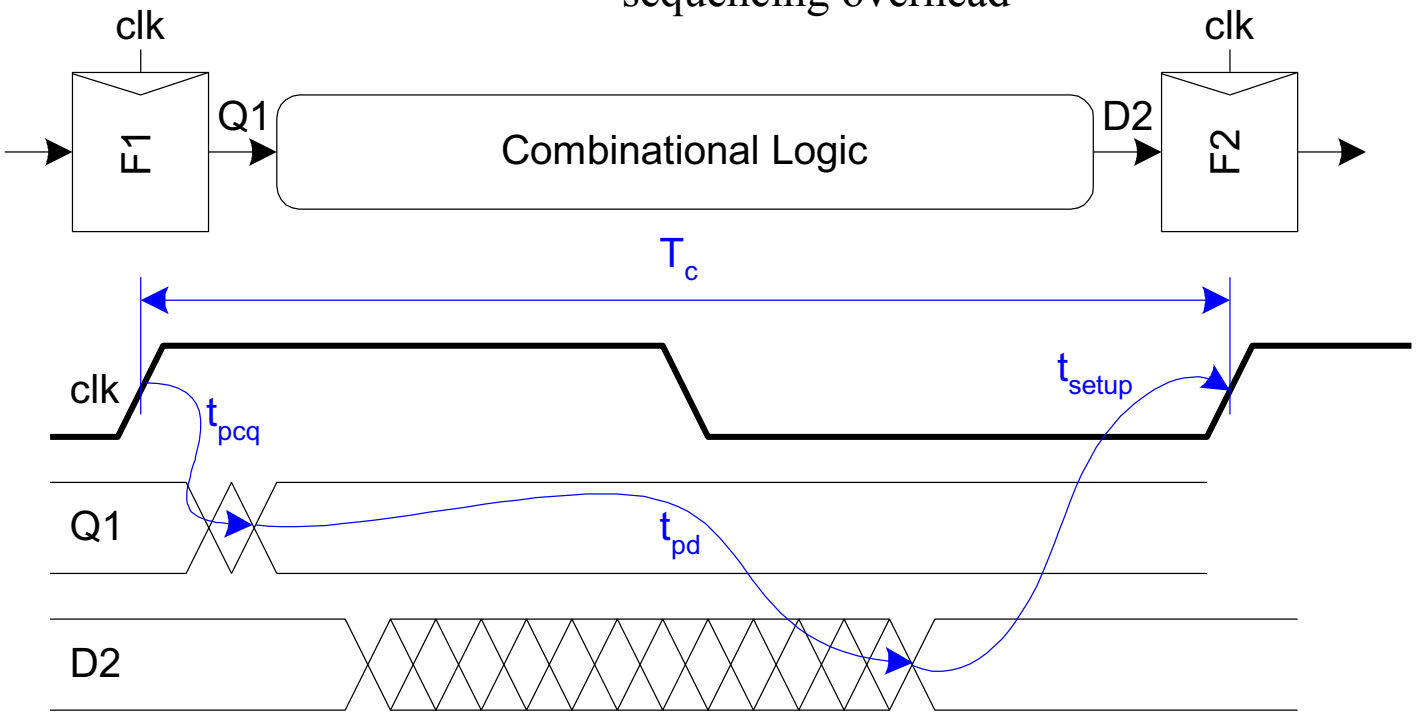






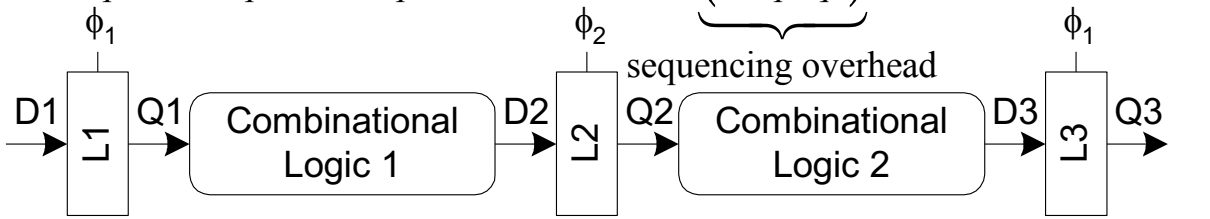
Max-Delay: Flip-Flops

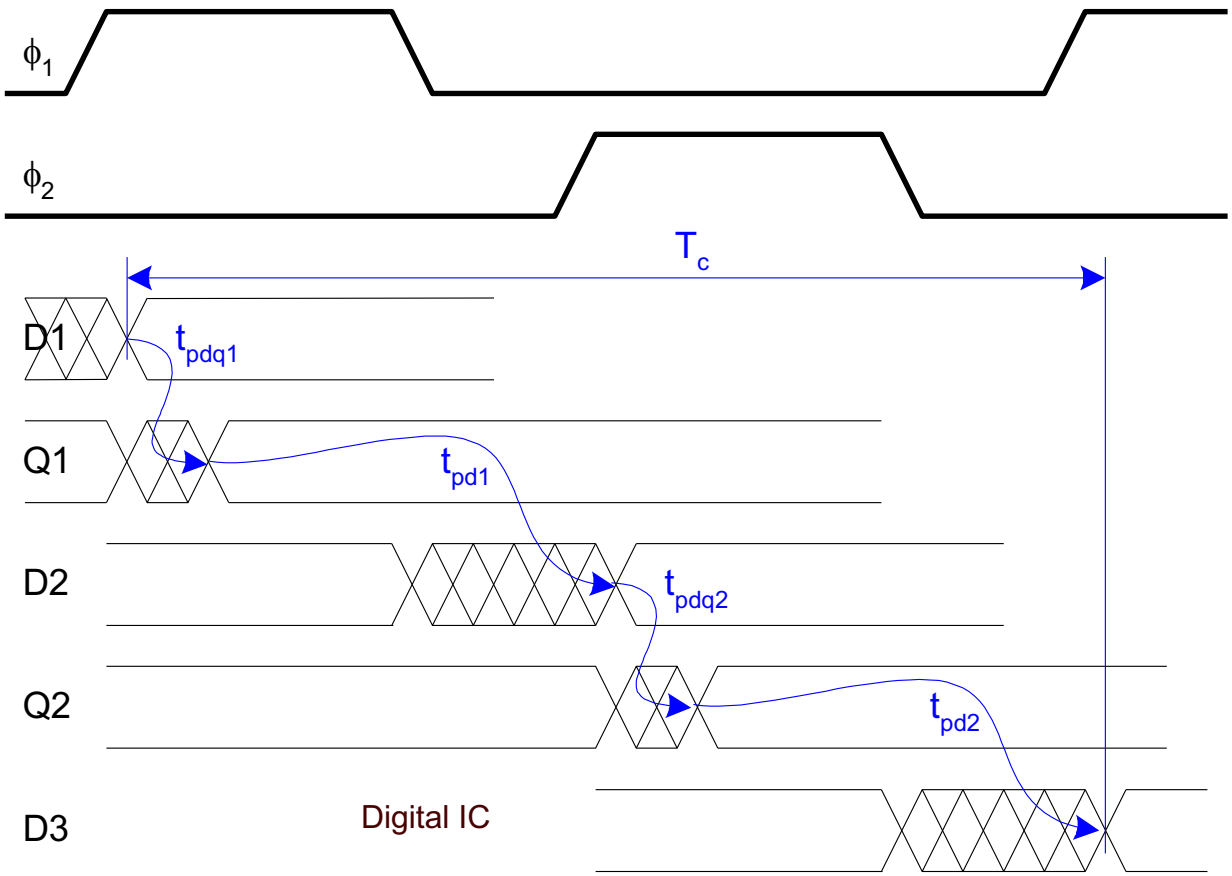
$$t_{pd} \uparrow < T_c \uparrow - \underbrace{\left(t_{\text{setup}} + t_{pcq} \right)}_{\text{sequencing overhead}}$$



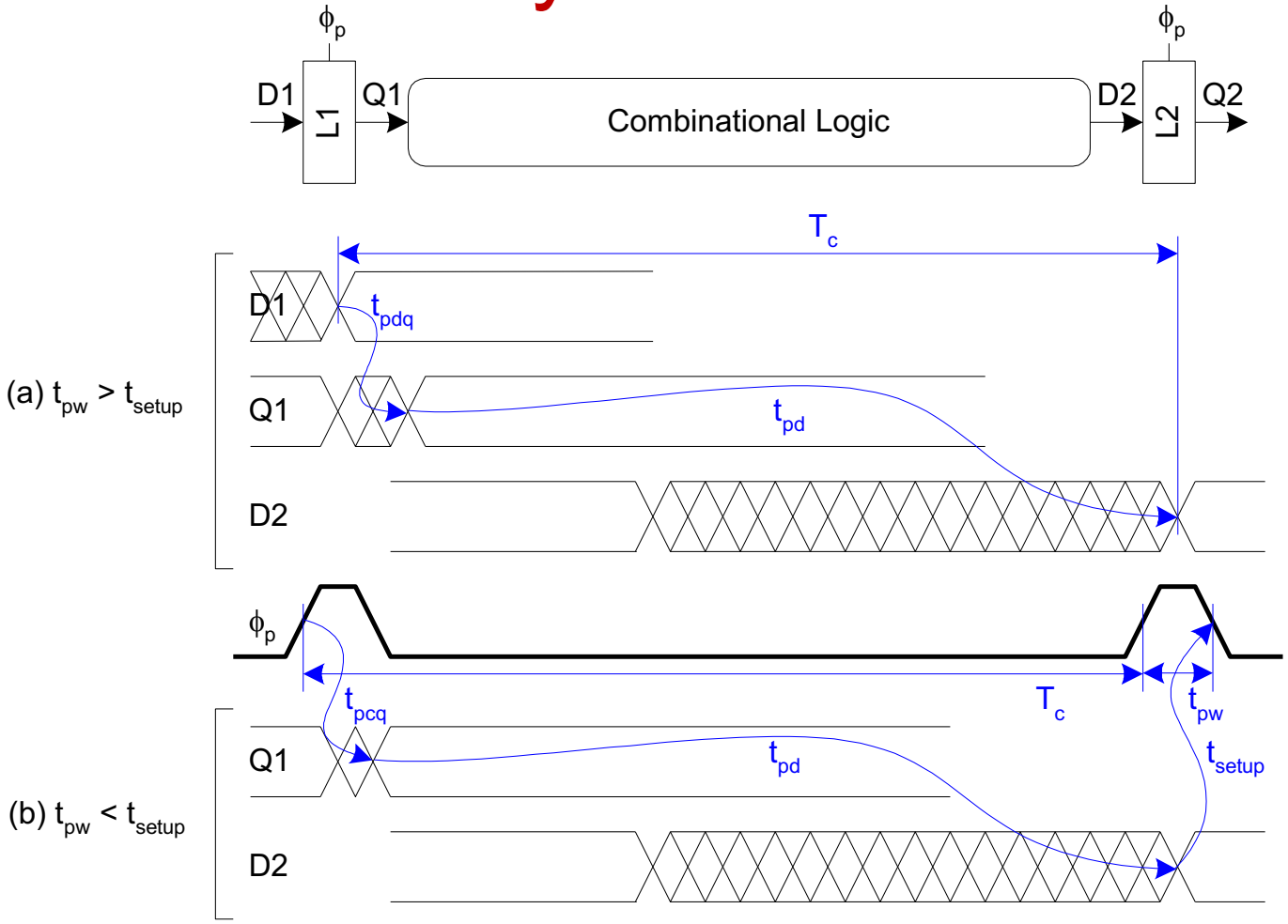
Max Delay: 2-Phase Latches

$$t_{pd} = t_{pd1} + t_{pd2} < T_c - \underbrace{(2t_{pdq})}_{\text{sequencing overhead}}$$





Max Delay: Pulsed Latches

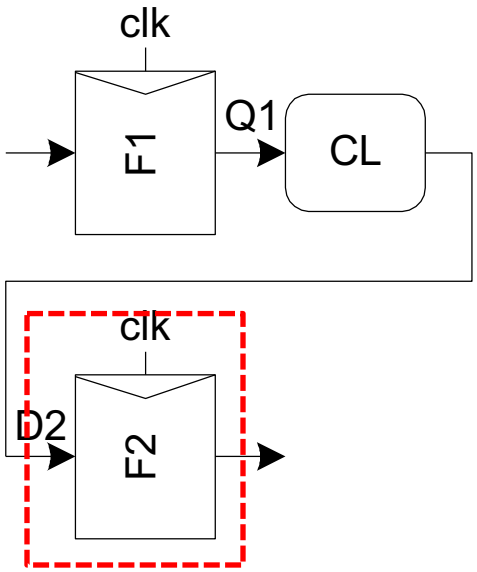


$$T_c \geq \max(t_{pdq} + t_{pd}, t_{pcq} + t_{pd} + t_{setup} - t_{pw})$$

$$t_{pd} = T_c - \max(t_{pdq}, t_{pcq} + t_{setup} - t_{pw})$$

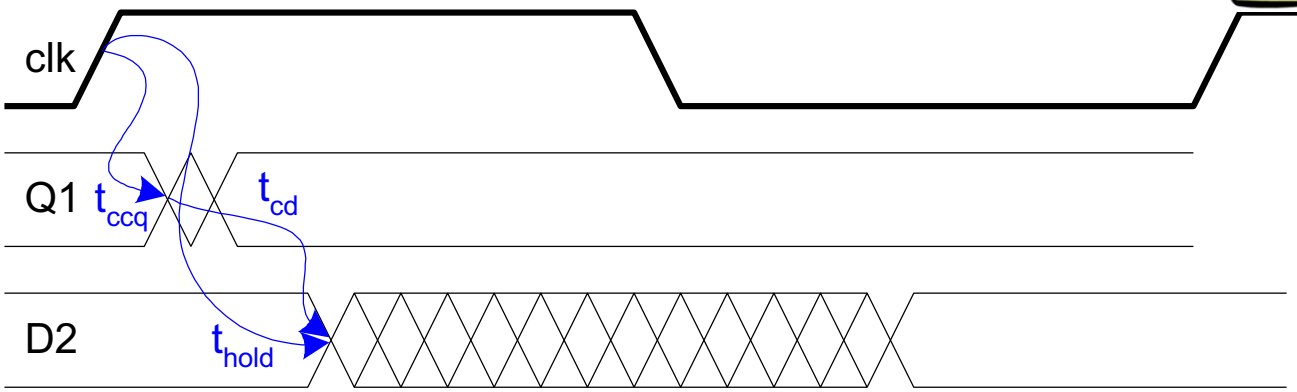
Sequencing overhead

Min-Delay: Flip-Flops



Race condition
Hold time failure
min-delay failure

$$t_{cd} \downarrow \geq t_{\text{hold}} - t_{ccq}$$

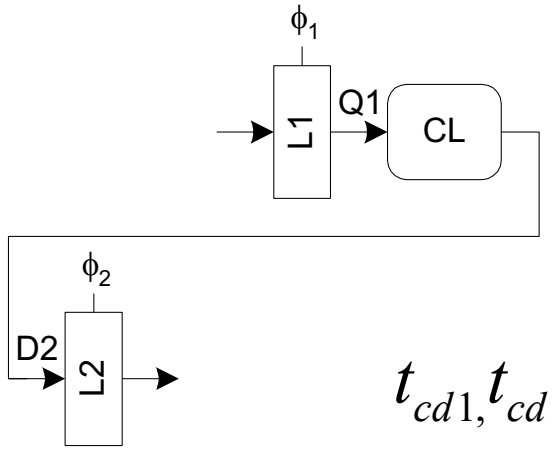


Min-Delay: 2-Phase Latches

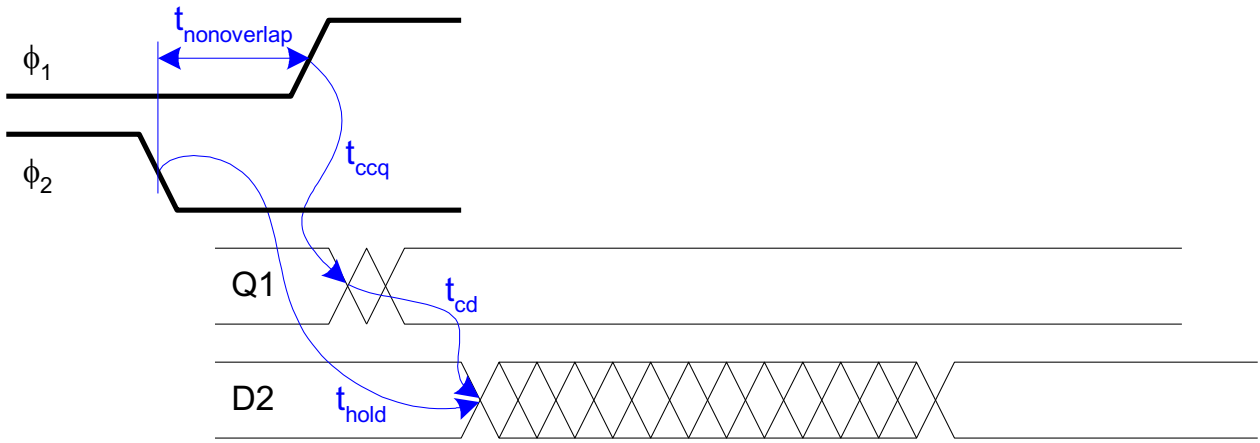
Hold time reduced by nonoverlap

Paradox: hold applies twice each cycle, vs. only once for flops.

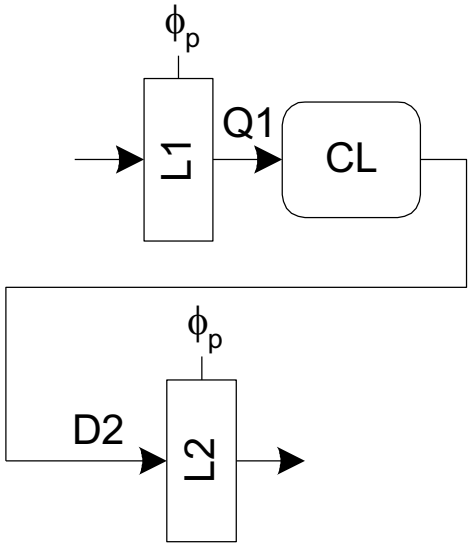
But a flop is made of two latches!



$$t_{cd1}, t_{cd2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}}$$

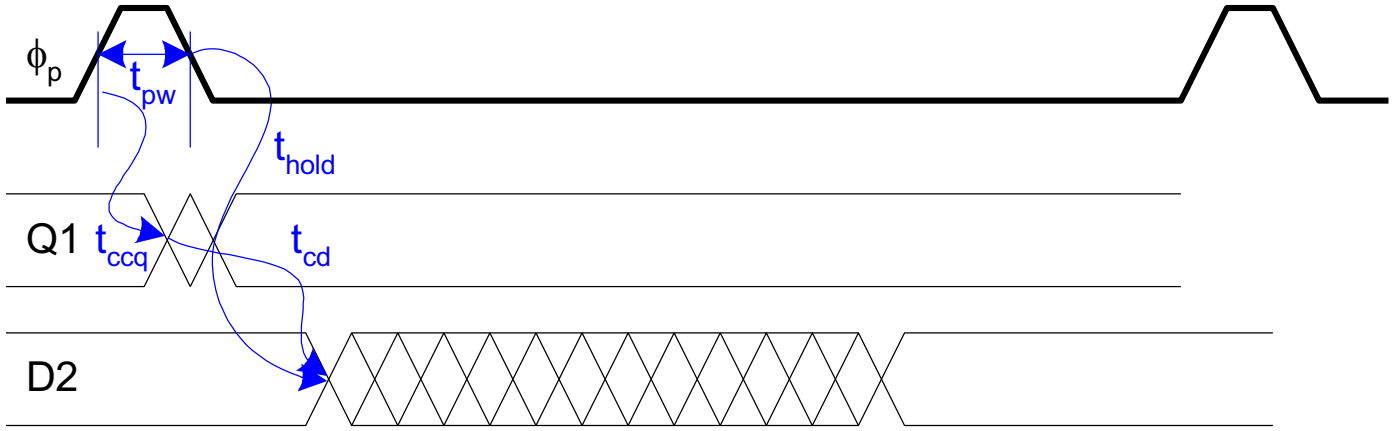


Min-Delay: Pulsed Latches



Hold time increased by pulse width

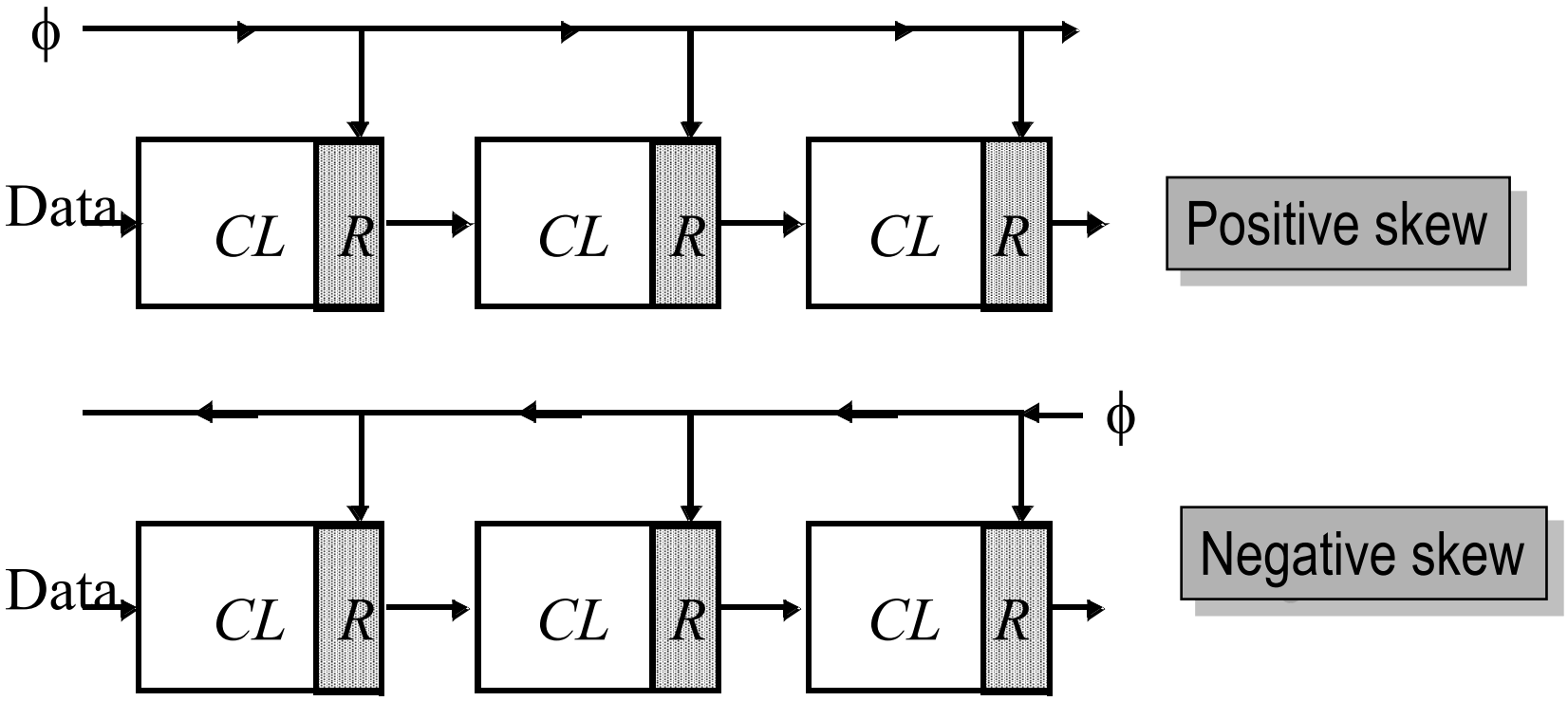
$$t_{cd} \geq t_{\text{hold}} - t_{ccq} + t_{pw}$$



Clock skew

- Definition
  - The spatial variation in arrival time of a clock transition on an integrated circuit
- Sources
  - Static mismatches in the clock paths and differences in the clock load

Positive/negative skew



Clock skew about FF

***negative* skew**

$$t_{pdq} \le T_c - (t_{pcq} + t_{setup} + t_{skew})$$

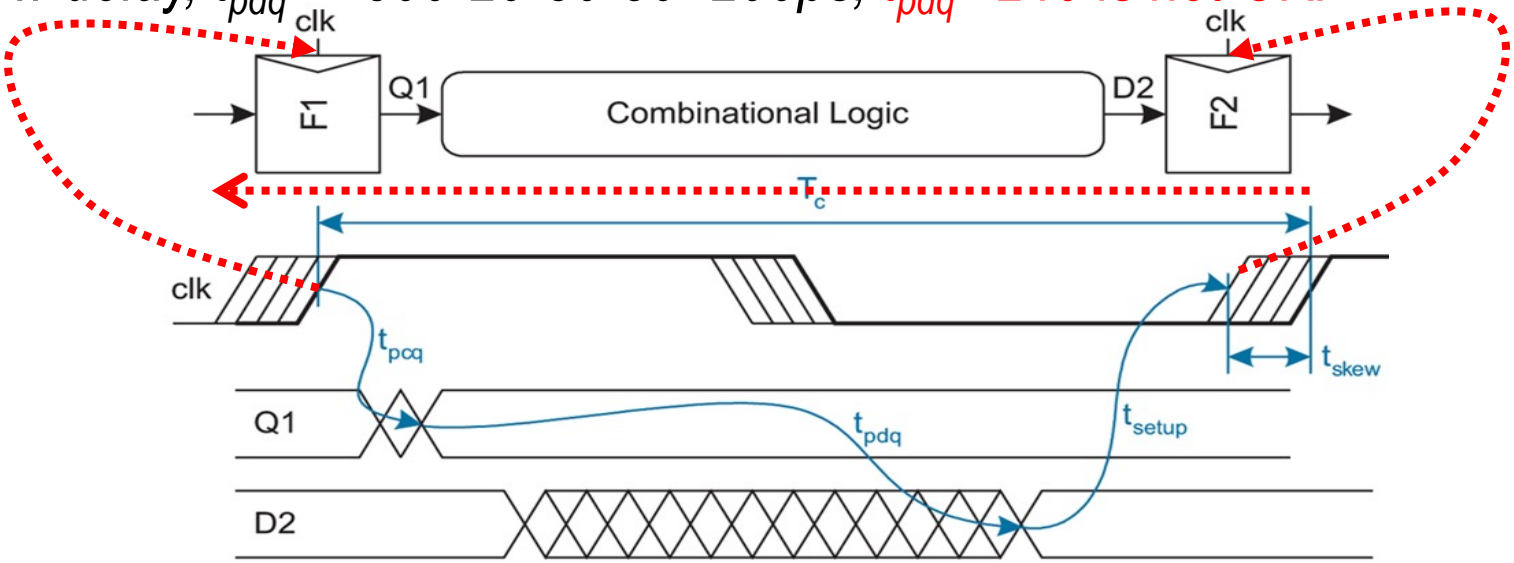
***Positive* skew for *better* condition**

Large is better !

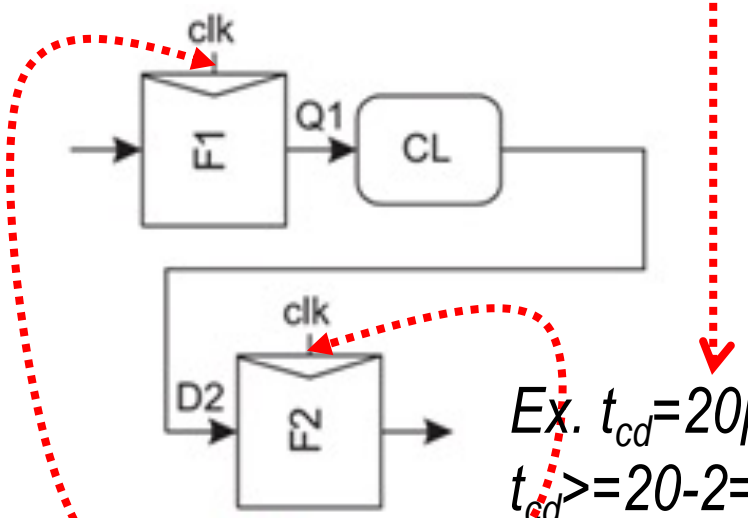
Ex. t_{pcq}=20ps,t_{setup}=30ps, T_c=300ps,t_{skew}=50ps

t_{pdq}<=300-20-30=250 , t_{pdq} =240 is OK

Including skew delay, t_{pdq} <=300-20-30-50=200ps, t_{pdq} =240 is not OK!



Clock skew about FF



Small is better !

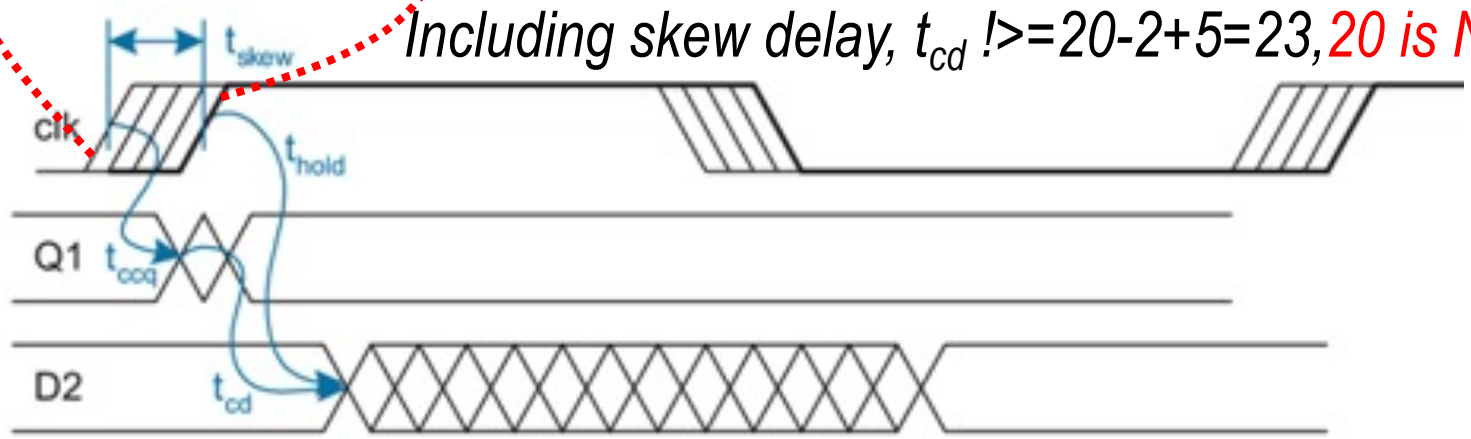
$$t_{cd} \geq t_{hold} - t_{ccq} + t_{skew}$$

***Positive* skew for *worse* condition**

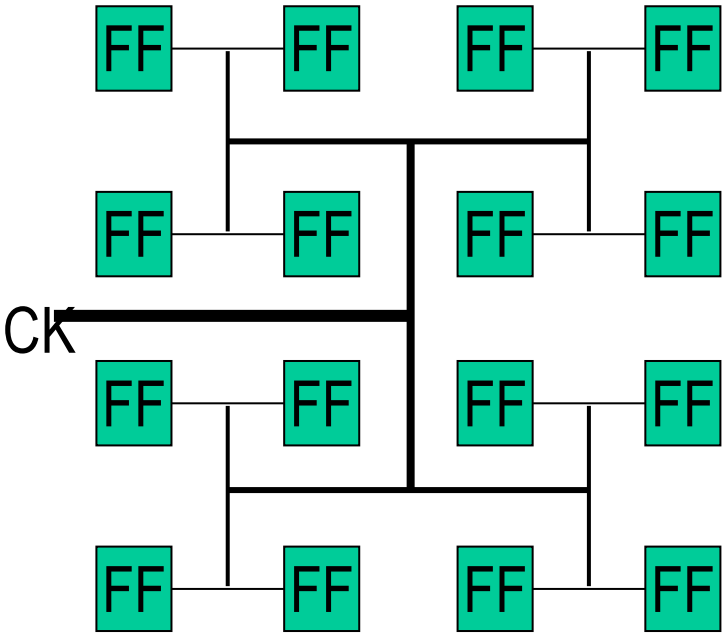
Ex. t_{cd}=20ps,t_{hold}=20ps, t_{ccq}=2ps,t_{skew}=5ps

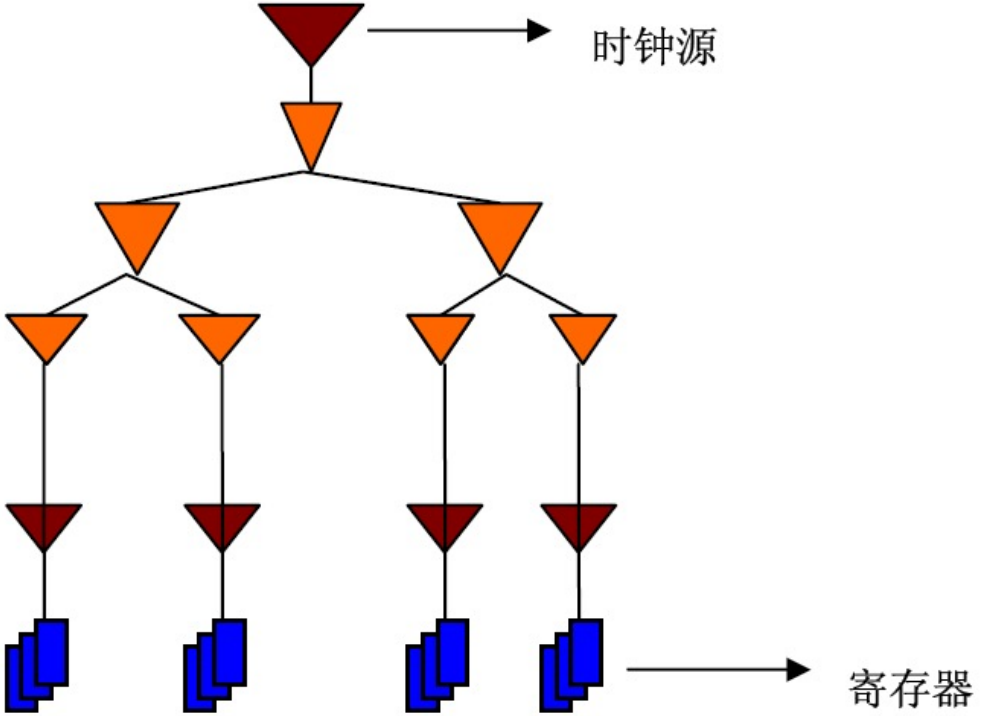
t_{cd}>=20-2=18, t_{cd}=20 is OK!

Including skew delay, t_{cd} !>=20-2+5=23,20 is Not OK



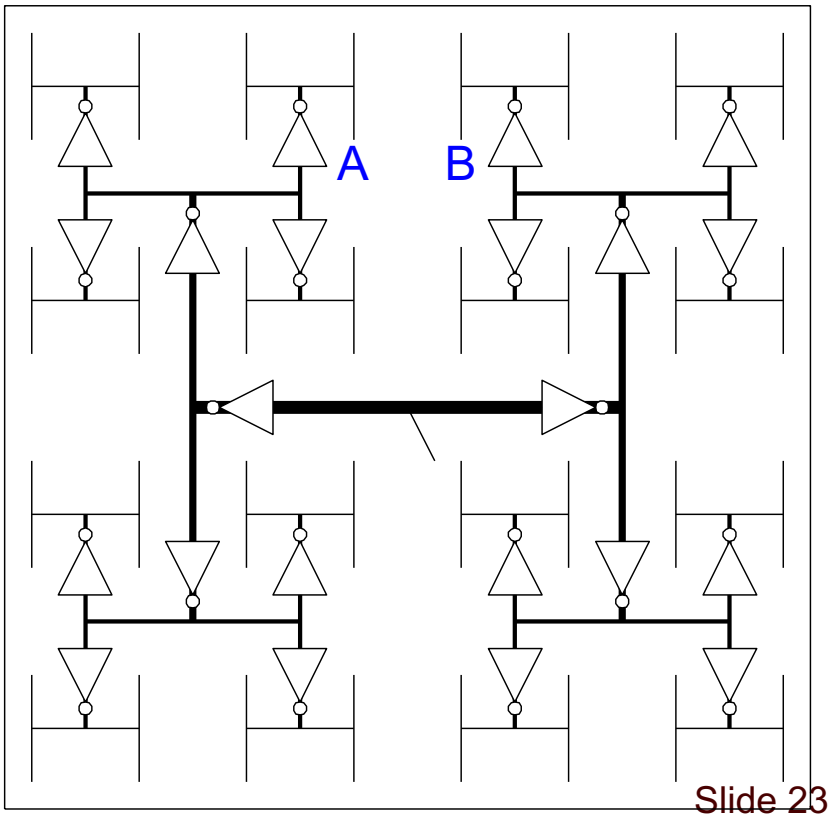
Clock tree





H-Trees

- Fractal structure
  - Gets clock arbitrarily close to any point
  - Matched delay along all paths
- Delay variations cause skew
- A and B might see big skew



Design Sequential Logic Circuits

- Introduction
- Timing
- ***Static Latches and Registers***
- Dynamic Latches and Registers

Static vs Dynamic Storage

- Static storage
  - preserve state as long as the power is on
  - have positive feedback (**regeneration**)
  - useful when updates infrequent (clock gating)
- Dynamic storage
  - store state on parasitic capacitors
  - only hold state for short periods of time (ms)
  - require periodic refresh
  - usually simpler, higher speed , lower power

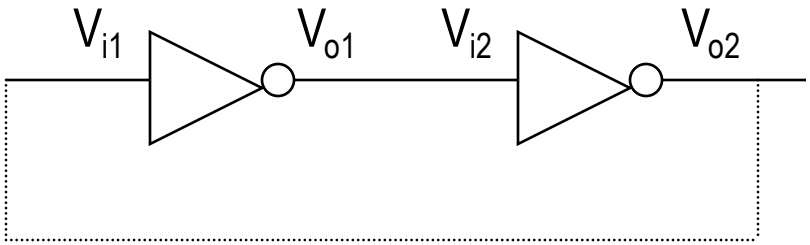
Static Latches and Registers

- The bistability principle
- Multiplexer-based latches
- Master-slave edge-triggered register
- Low-voltage static latches
- Static SR Flip-flops-writing data by pure force

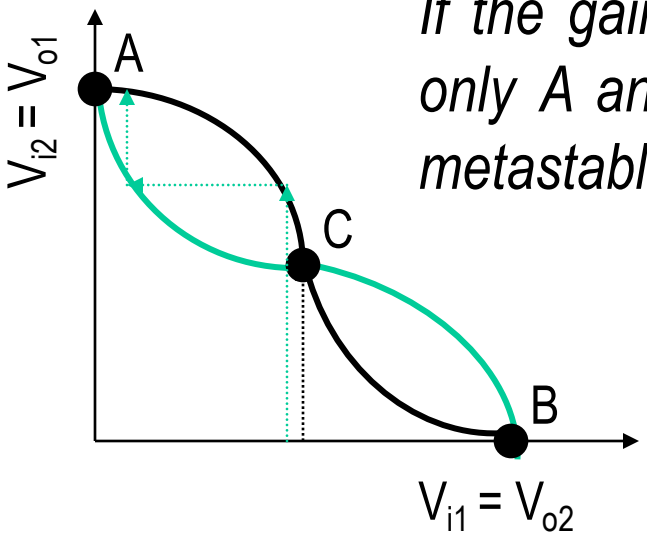
Static Latches and Registers

- ***The bistability principle***
- Multiplexer-based latches
- Master-slave edge-triggered register
- Low-voltage static latches
- Static SR Flip-flops-writing data by pure force

Review: The Regenerative Property



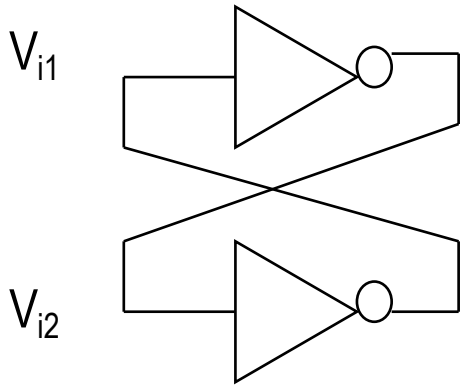
cascaded inverters



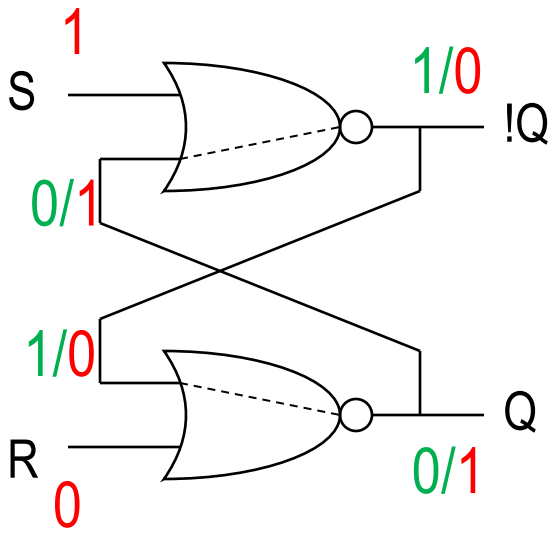
If the gain in the transient region is larger than 1, only A and B are stable operation points. C is a metastable operation point.

Bistable Circuits

- The cross-coupling of two inverters results in a bistable circuit (a circuit with two stable states)
- Have to be able to change the stored value by making A (or B) temporarily unstable by increasing the loop gain to a value larger than 1
  - done by applying a trigger pulse at Vi1 or Vi2
  - the width of the trigger pulse need be only a little larger than the total propagation delay around the loop circuit (twice the delay of an inverter)



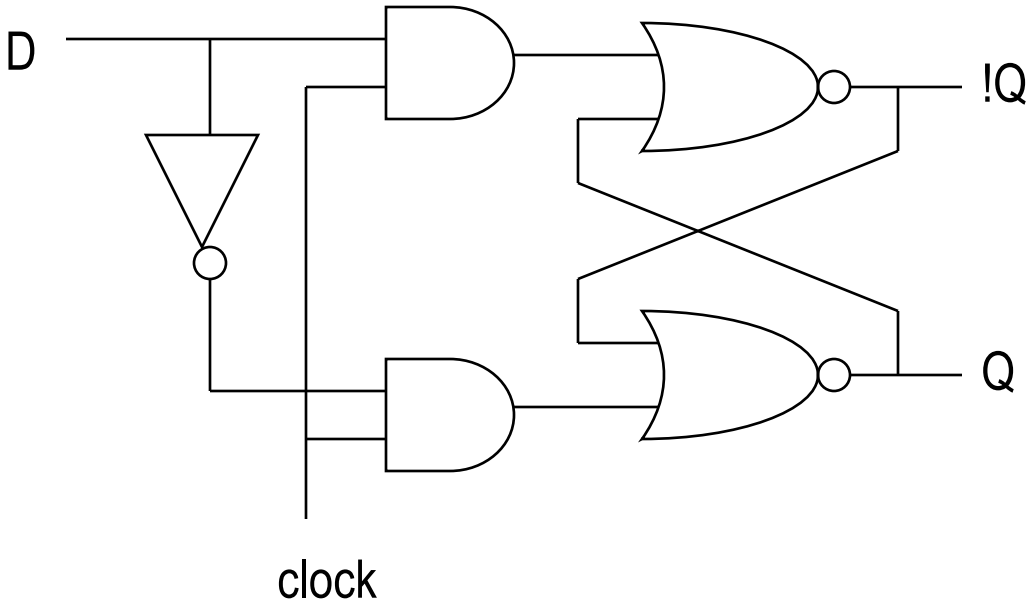
SR Latch

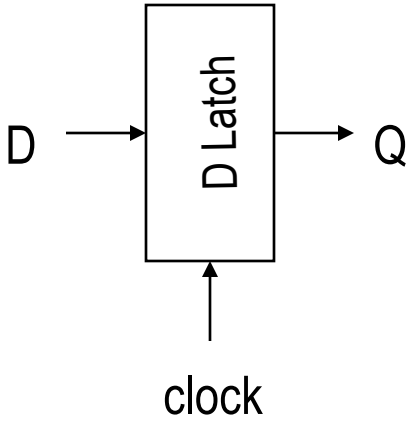


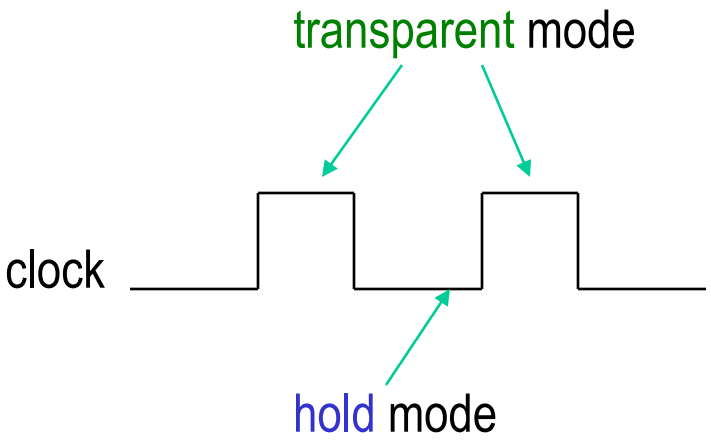
S	R	Q	!Q	
0	0	Q	!Q	memory
1	0	1	0	set
0	1	0	1	reset
1	1	0	0	disallowed

Break the feedback

Clocked D Latch





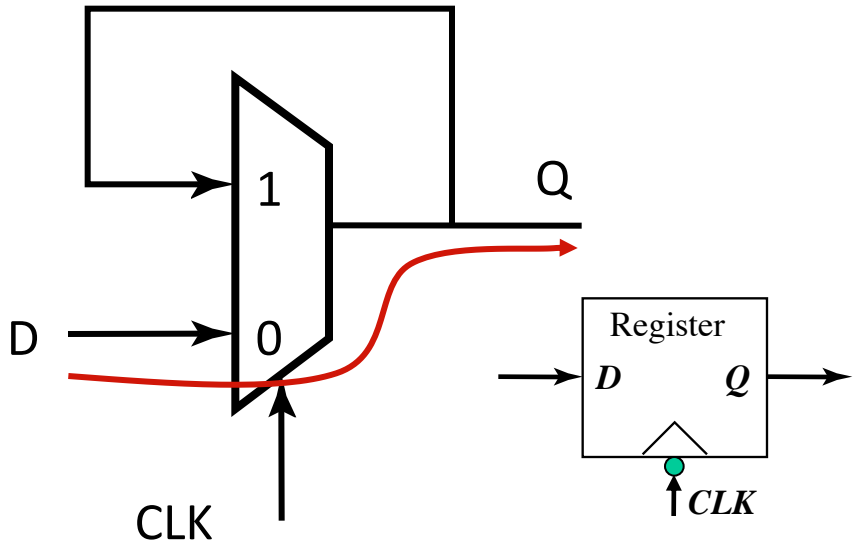


Static Latches and Registers

- The bistability principle
- ***Multiplexer-based latches***
- Master-slave edge-triggered register
- Low-voltage static latches
- Static SR Flip-flops-writing data by pure force

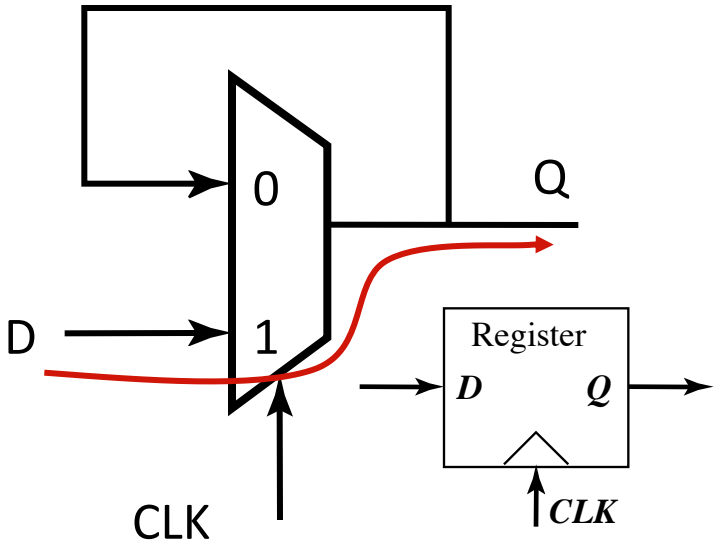
Mux-Based Latches

Negative latch
(transparent when CLK= 0)



$$Q = Clk \cdot Q + \overline{Clk} \cdot In$$

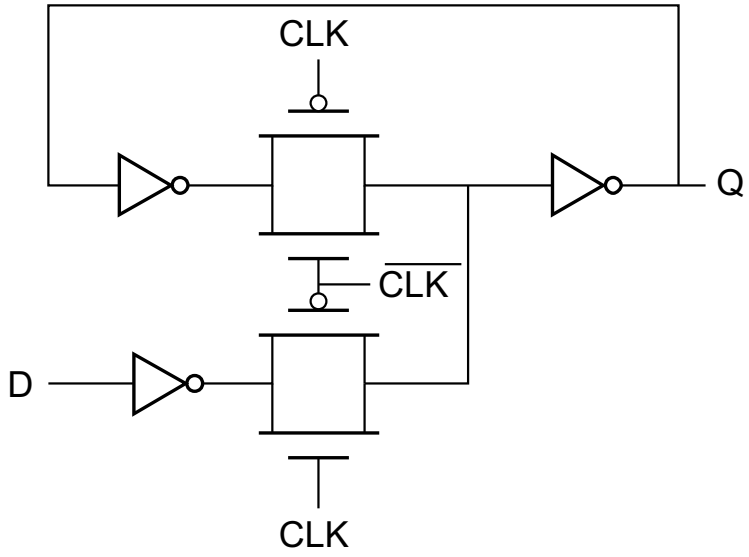
Positive latch
(transparent when CLK= 1)



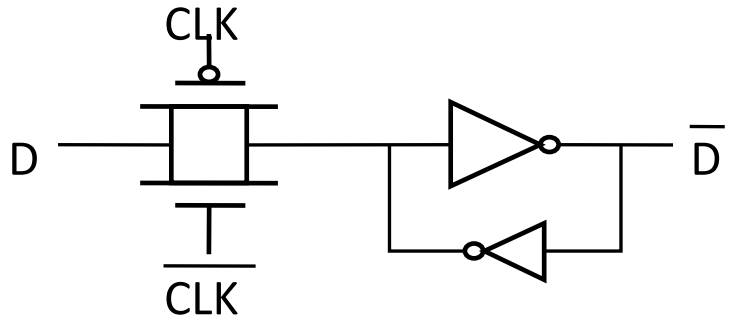
$$Q = \overline{Clk} \cdot Q + Clk \cdot In$$

Writing into a Static Latch

*Use the clock as a decoupling signal, that distinguishes between the **transparent** and **opaque** states*

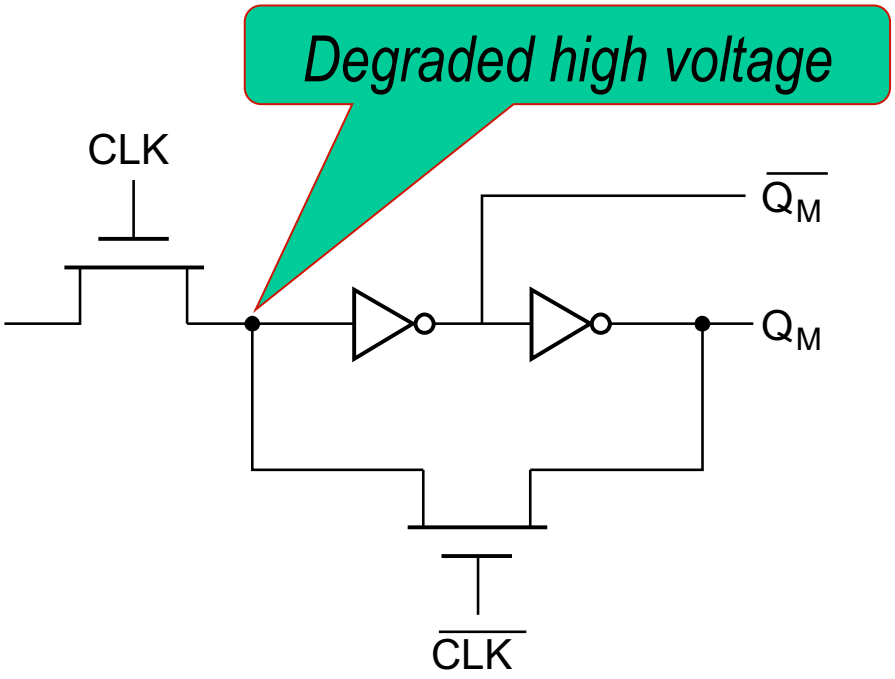


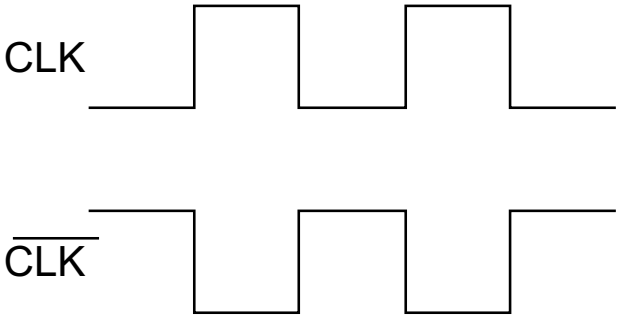
Converting into a MUX



Forcing the state(can implement as NMOS-only)

Mux-Based Latch



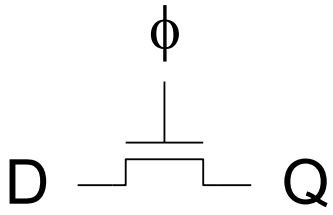


NMOS only

Non-overlapping clocks

Latch Design

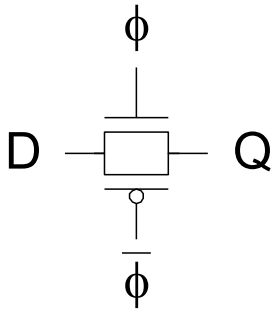
- Pass Transistor Latch
- Pros
  - + Tiny
  - + Low clock load
- Cons
  - V_t drop
  - nonrestoring
  - backdriving
  - output noise sensitivity
  - dynamic
  - diffusion input



Used in 1970's

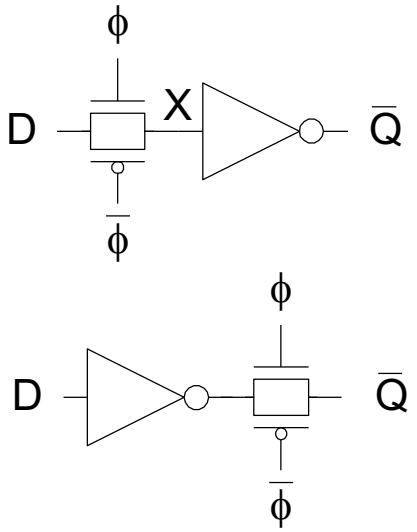
Latch Design

- Transmission gate
  - \+ No V_t drop
  - \- Requires inverted clock



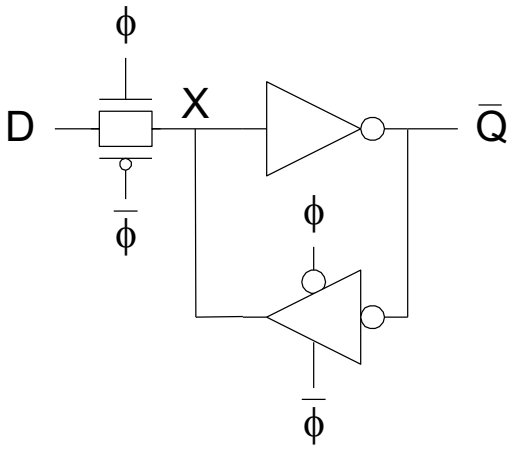
Latch Design

- Inverting buffer
  - + Restoring
  - + No backdriving
  - + Fixes either
    - Output noise sensitivity
    - Or diffusion input
  - Inverted output



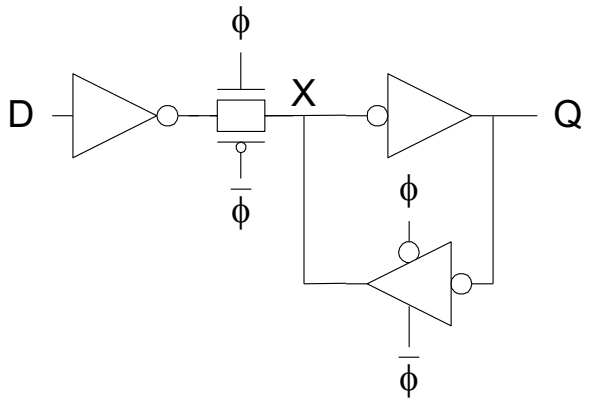
Latch Design

- Tristate feedback
  - + Static
  - • Backdriving risk
- Static latches are now essential



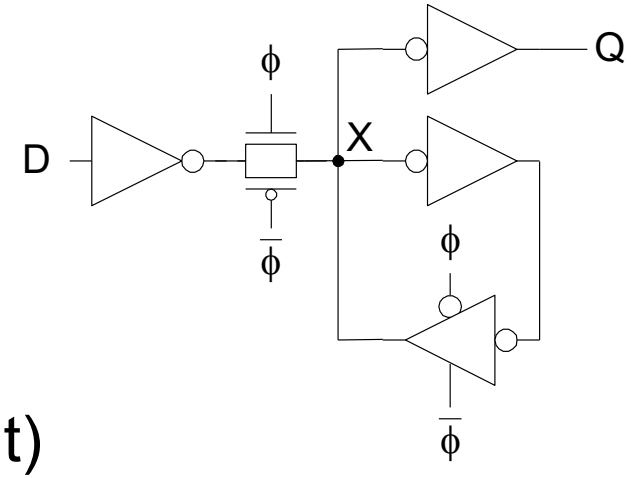
Latch Design

- Buffered input
  - + Fixes diffusion input
  - + Noninverting



Latch Design

- Buffered output
  - + No backdriving

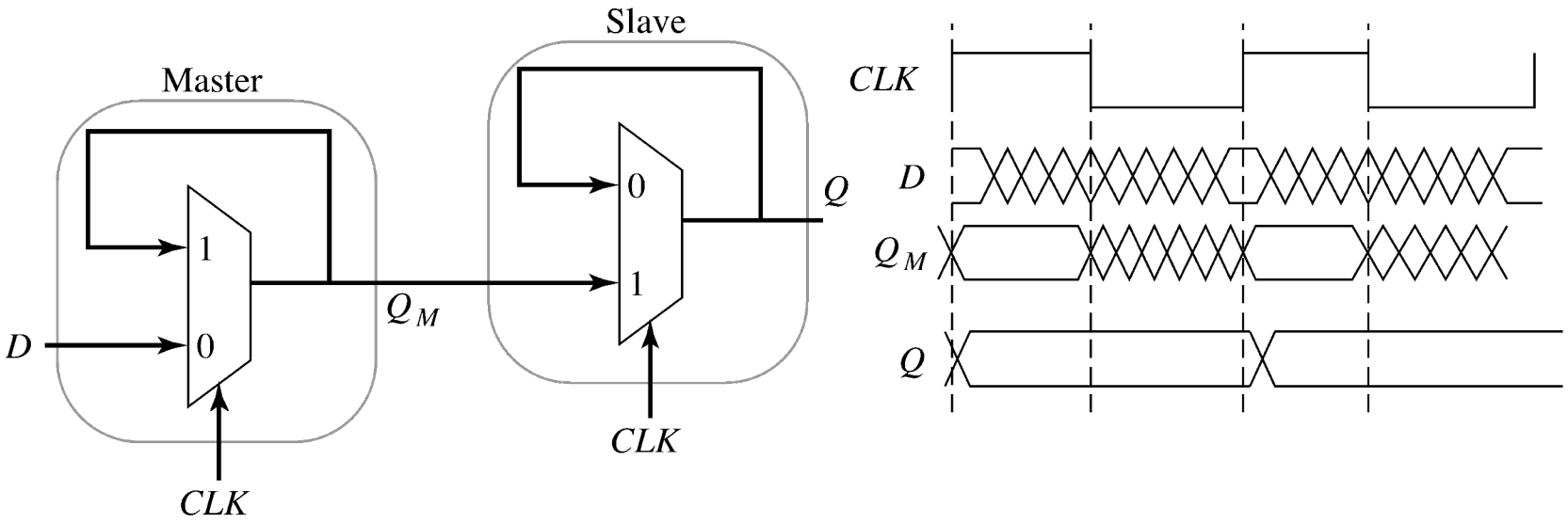


- Widely used in standard cells
  - + Very robust (most important)
  - - Rather large
  - - Rather slow (1.5 – 2 FO4 delays)
  - - High clock loading

Static Latches and Registers

- The bistability principle
- Multiplexer-based latches
- ***Master-slave edge-triggered register***
- Low-voltage static latches
- Static SR Flip-flops-writing data by pure force

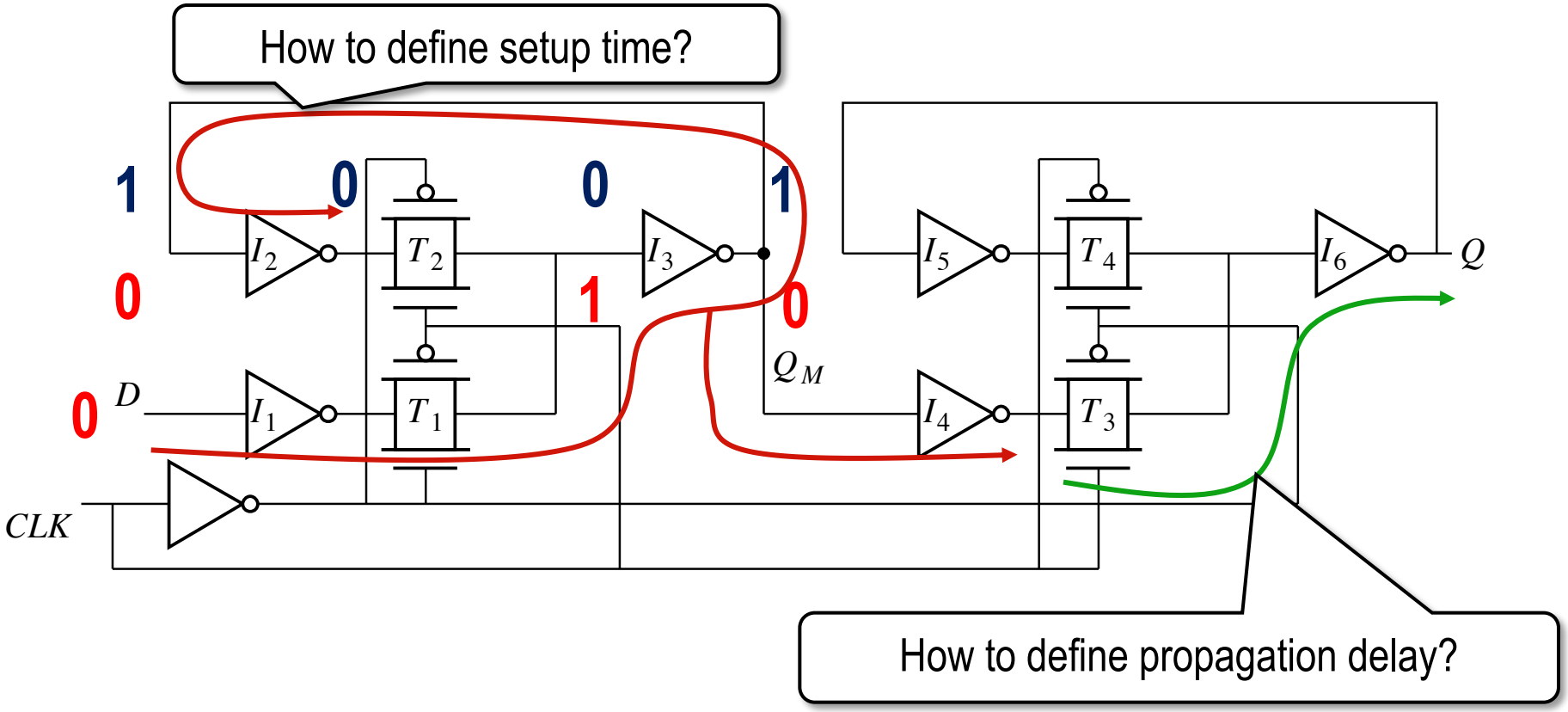
Master-Slave (Edge-Triggered) Register



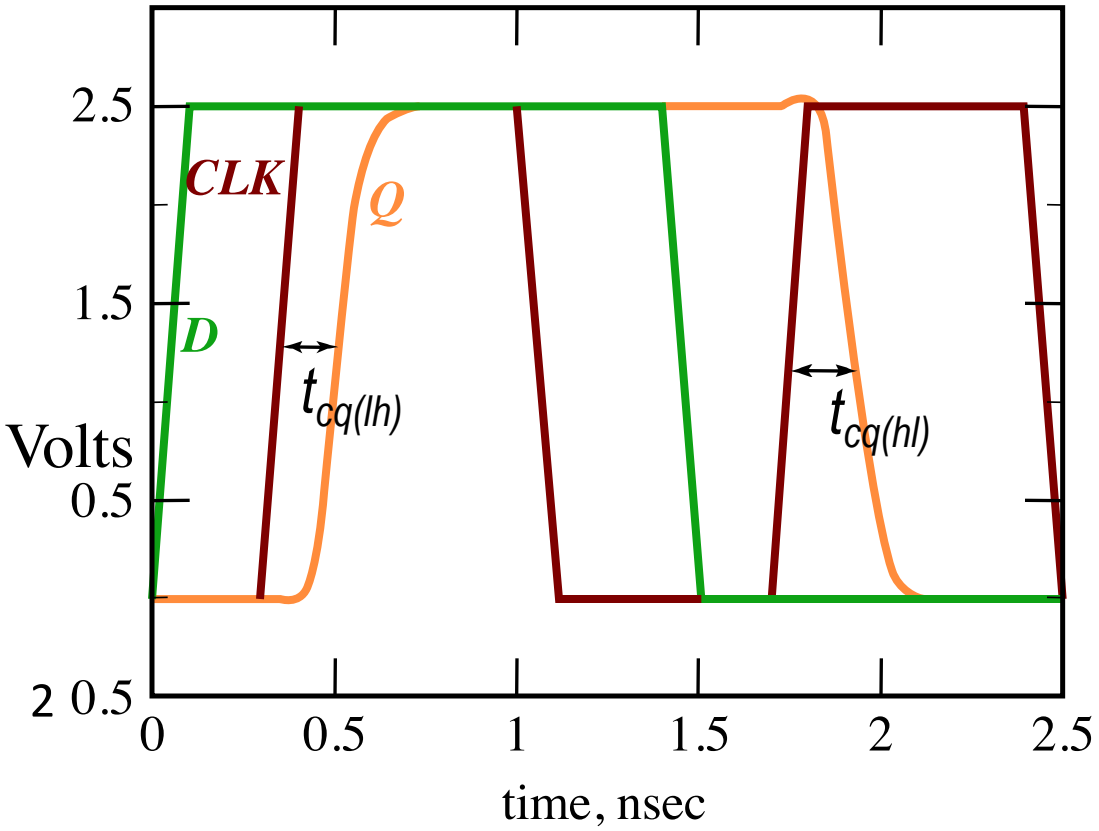
Two opposite latches trigger on edge Also called master-slave latch pair

Master-Slave Register

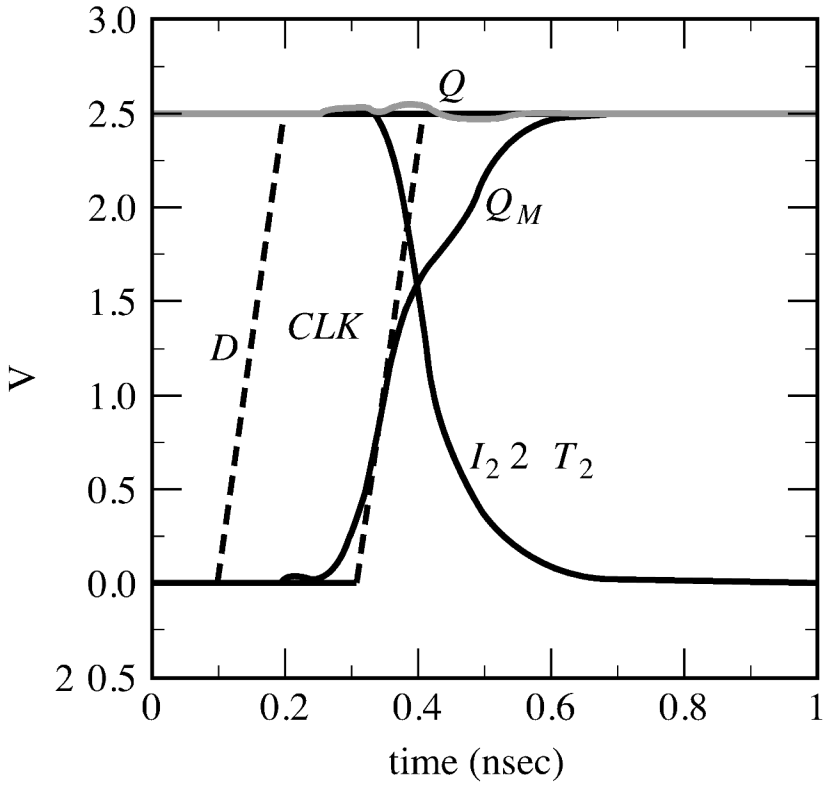
Multiplexer-based latch pair



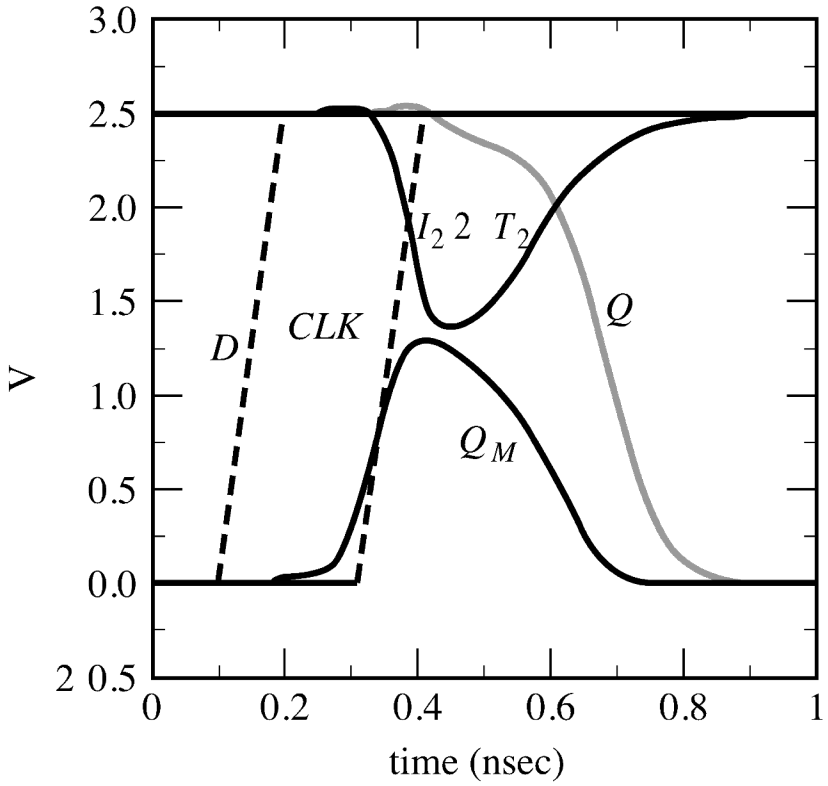
Clk-Q Delay



Setup Time

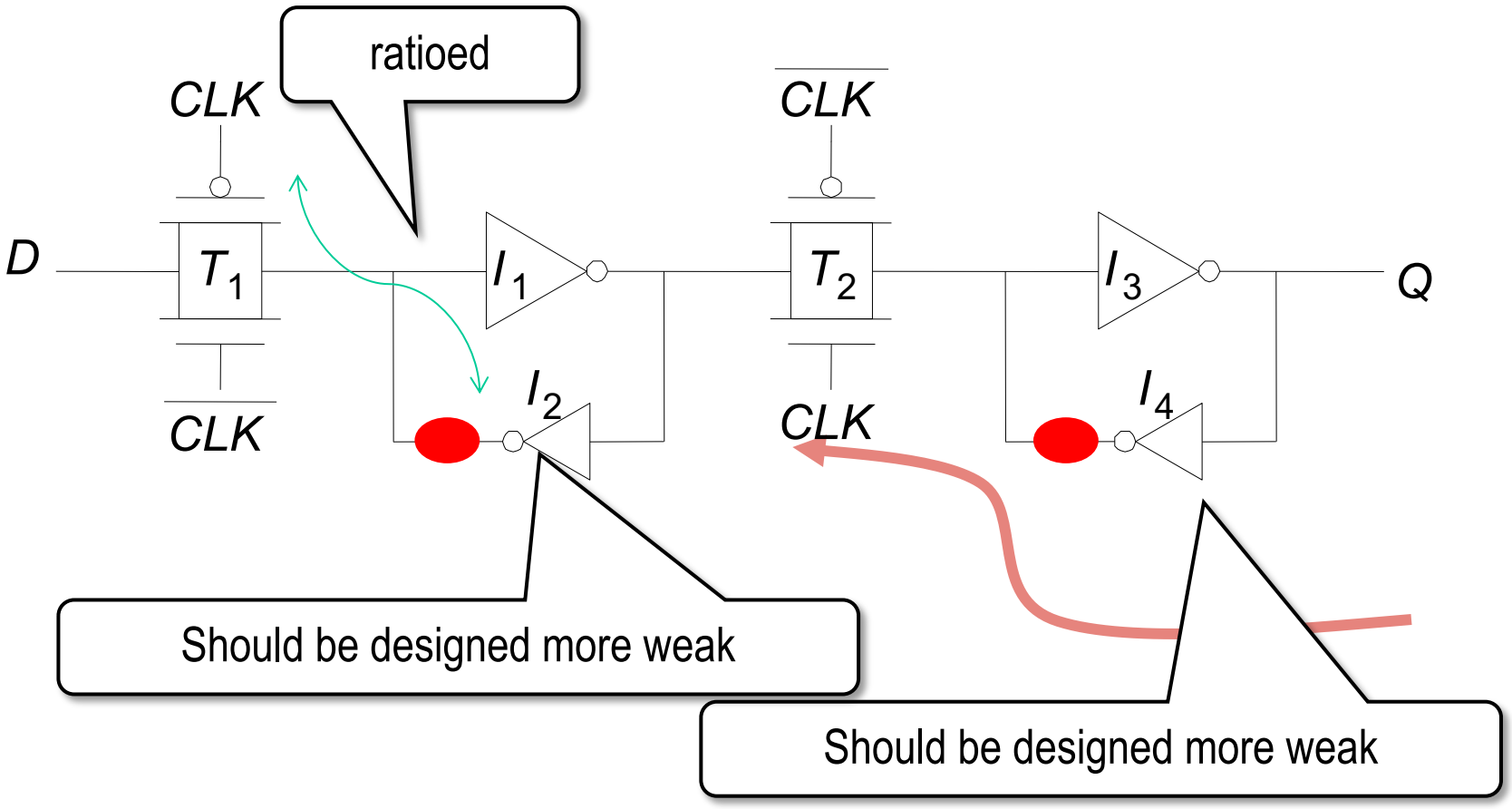


(a) T_{setup} 5 0.21 nsec

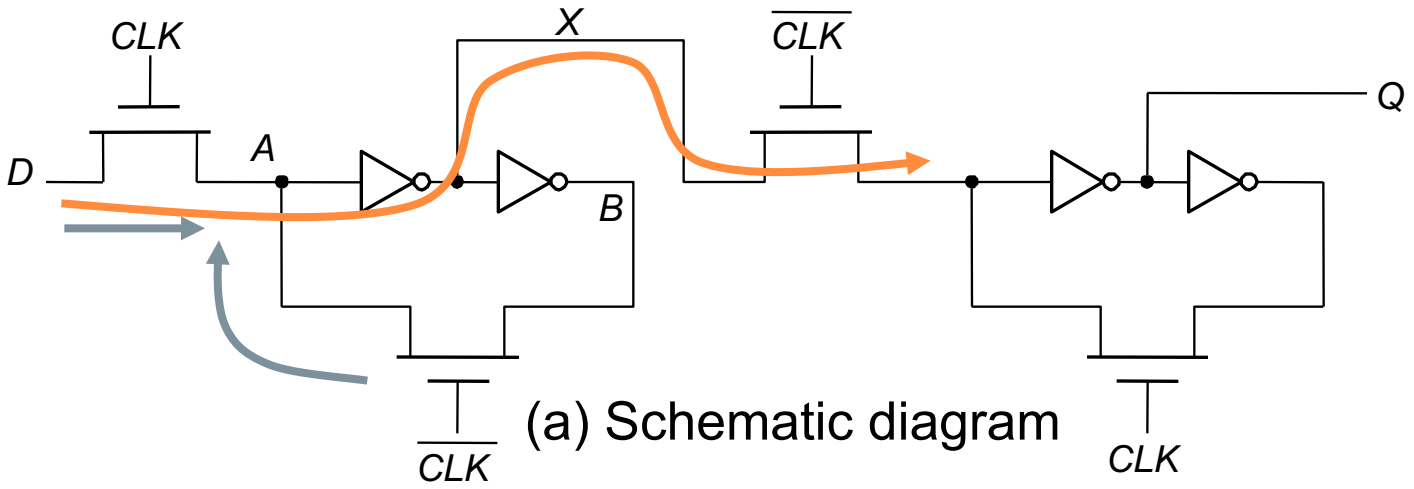


(b) T_{setup} 5 0.20 nsec

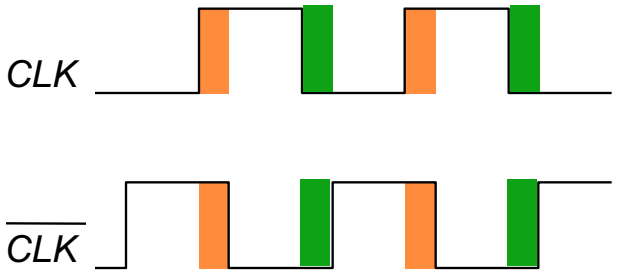
Reduced Clock Load Master-Slave Register



Avoiding Clock Overlap



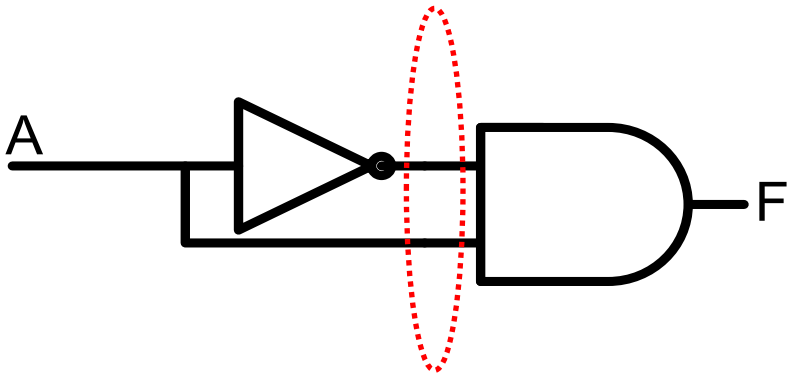
(a) Schematic diagram

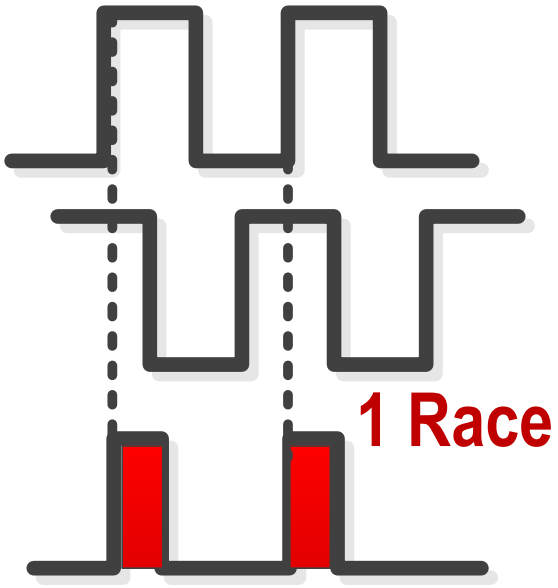


(b) Overlapping clock pairs

Race and hazard

Signal A and Second phase signal, meet with each one at gate G have ***Race***

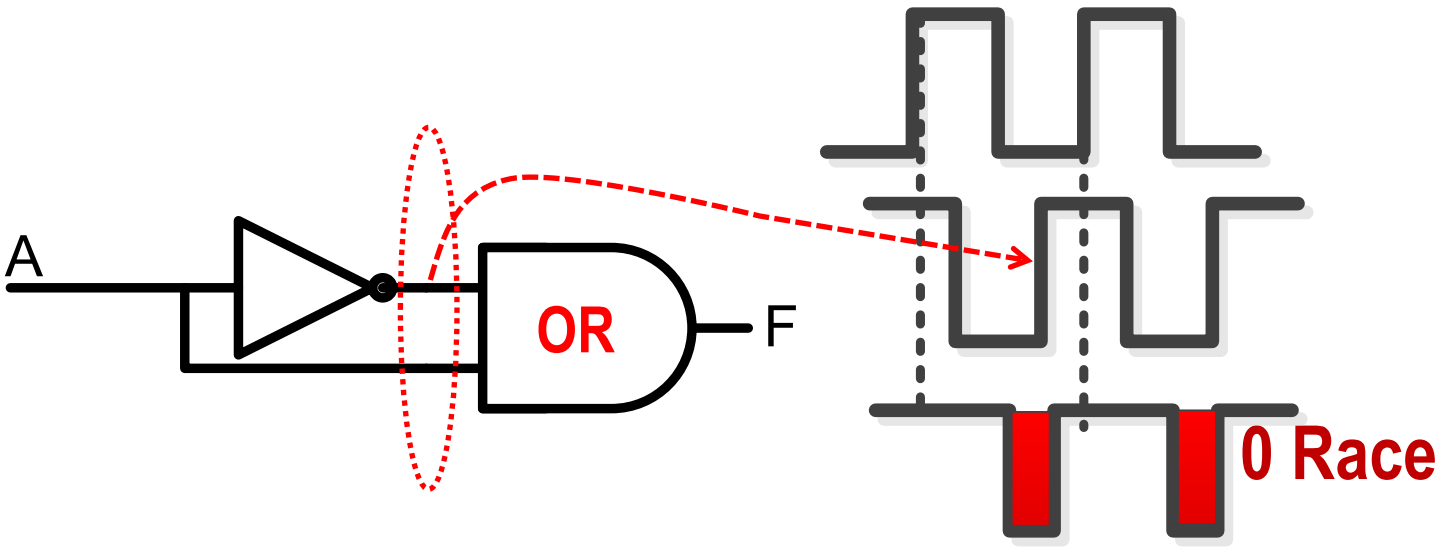




Result is Error which means ***Hazard***

Race and hazard

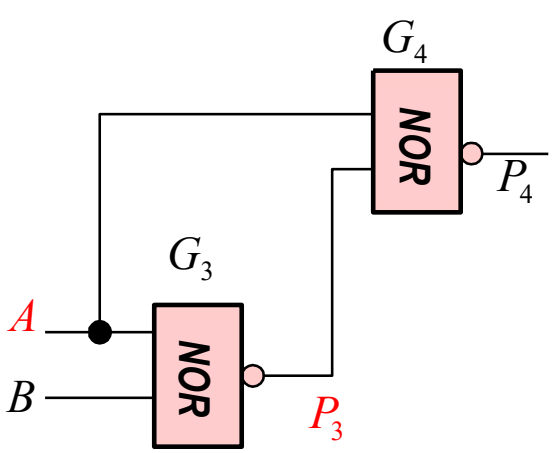
Signal A and Second phase signal, meet with each one at gate G have ***Race***

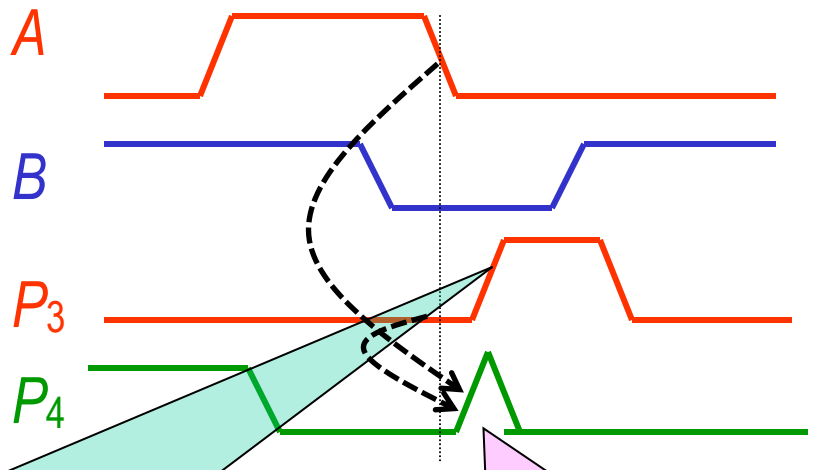


Result is Error which means ***Hazard***

Race and hazard

First phase signal A and Second phase signal P3, meet with each one at gate G4 and have race



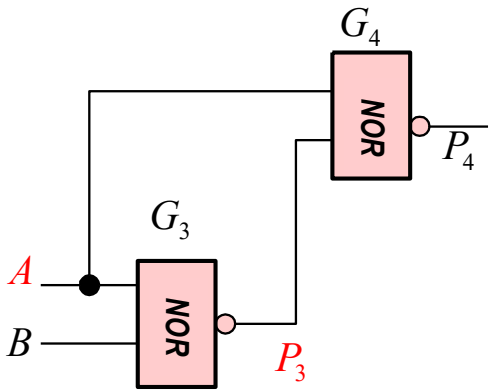


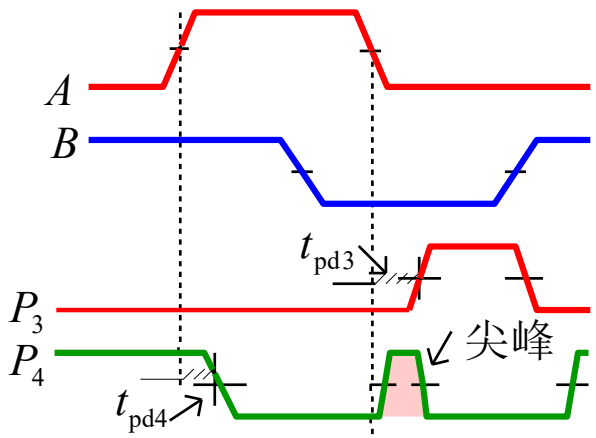
because P_3 after A， exist
a moment of “0” time,hazard happen at P_4

there is a error signal for NOR

Why hazard happen

- **Hazard is driven by race**
- Race condition
  - when at least two input signal change to different direction
  - two changing input come at different time
- ***A* and P_3 are race signal**





Hazard condition is B=0 let A and P3 meet, B=1 not

Hazard examples

$$P = A + \overline{A}$$

0-hazard
Stable sate is “1”, 0-1, so 0 is unstable hazard state

$$P = A\overline{A}$$

1-hazard
Stable sate is “0”, 1-0, so 1 is unstable hazard state

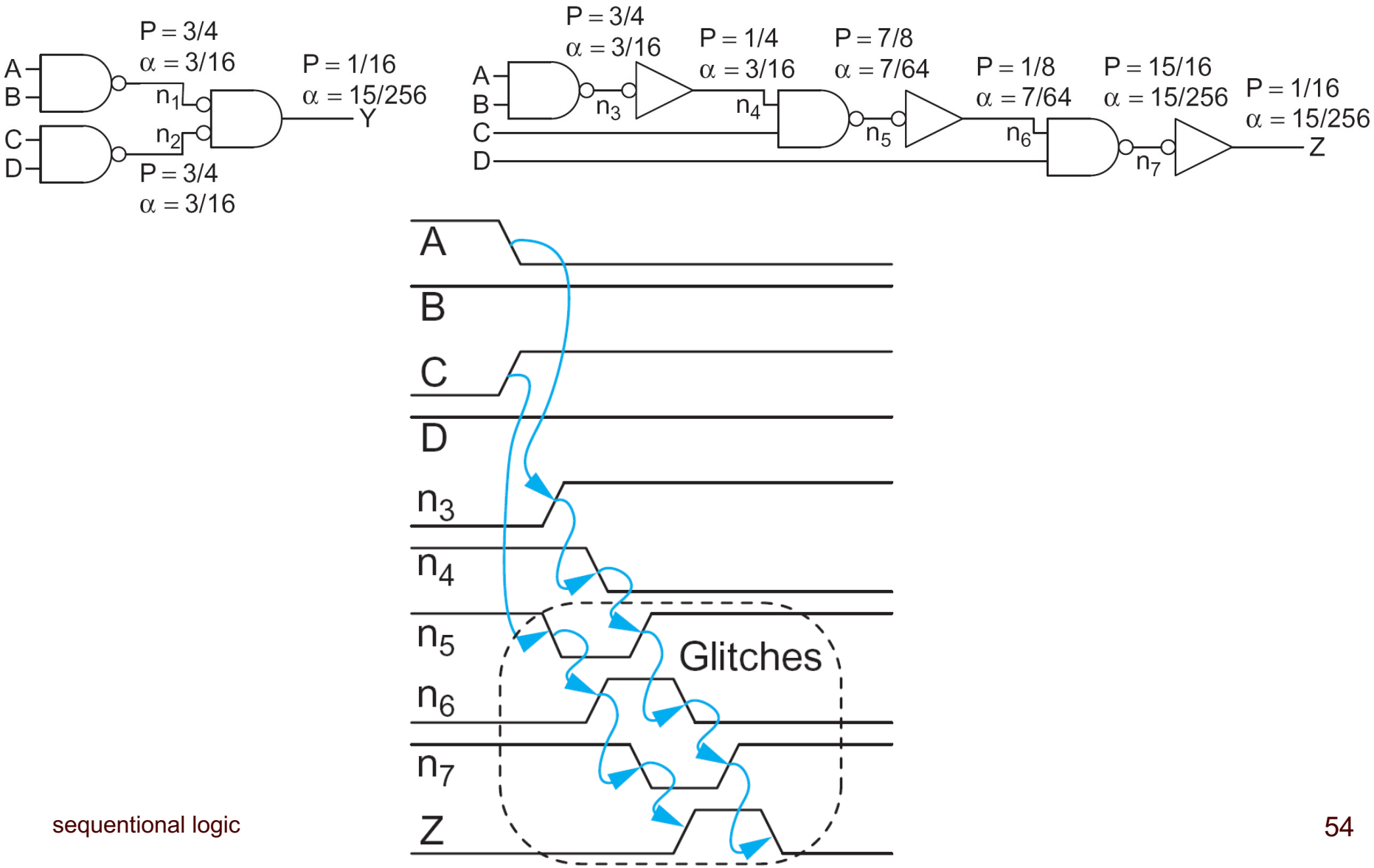
$$P_1 = AB + \overline{A}C$$

$$P_2 = (A + B)(\overline{A} + C)$$

$$P_3 = \overline{A}B + A\overline{C} + \overline{B}C$$

When B=C=1,P1=A+~A,”0” hazard happen
When B=C=0,P2=A~A,”1” hazard happen
When B=1/C=0,P3=A+~A ,”0” hazard happen
When C=1/A=0,P3=B+~B, 0” hazard happen
When A=1/B=0,P3=C+~C, 0” hazard happen

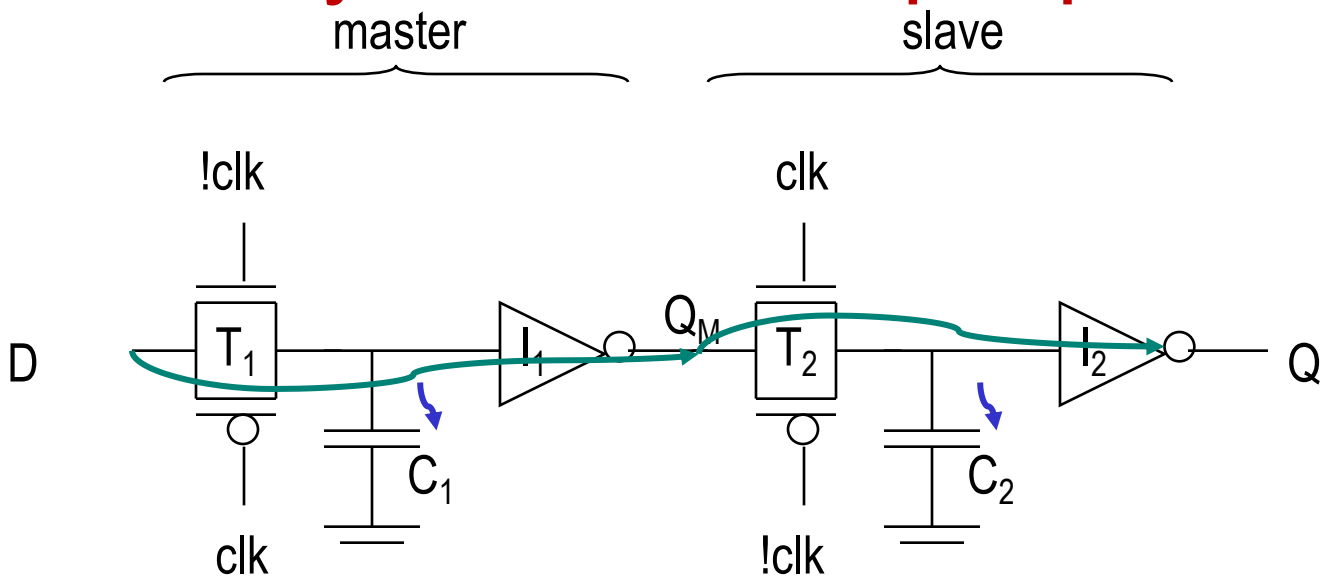
glitches


A
B
C
D
P = 3/4
α = 3/16
n1
n2
P = 3/4
α = 3/16
P = 1/16
α = 15/256
Y
P = 3/4
α = 3/16
n3
P = 1/4
α = 3/16
n4
P = 7/8
α = 7/64
n5
P = 1/8
α = 7/64
n6
P = 15/16
α = 15/256
n7
P = 1/16
α = 15/256
Z
A
B
C
D
n3
n4
n5
n6
n7
Z
Glitches

Design Sequential Logic Circuits

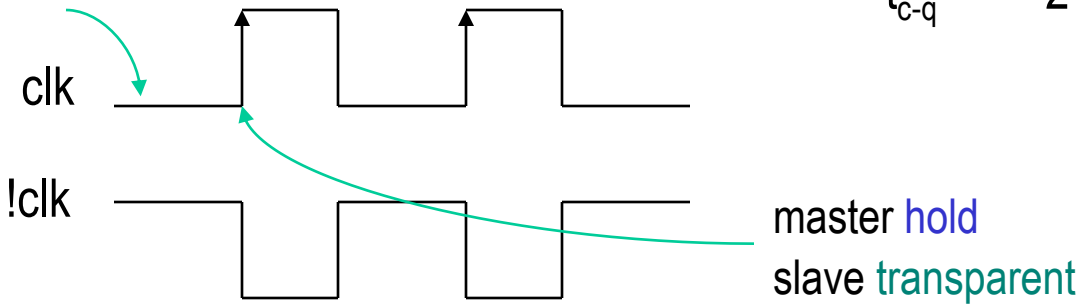
- Introduction
- Timing
- Static Latches and Registers
- ***Dynamic Latches and Registers***

Dynamic ET Flipflop

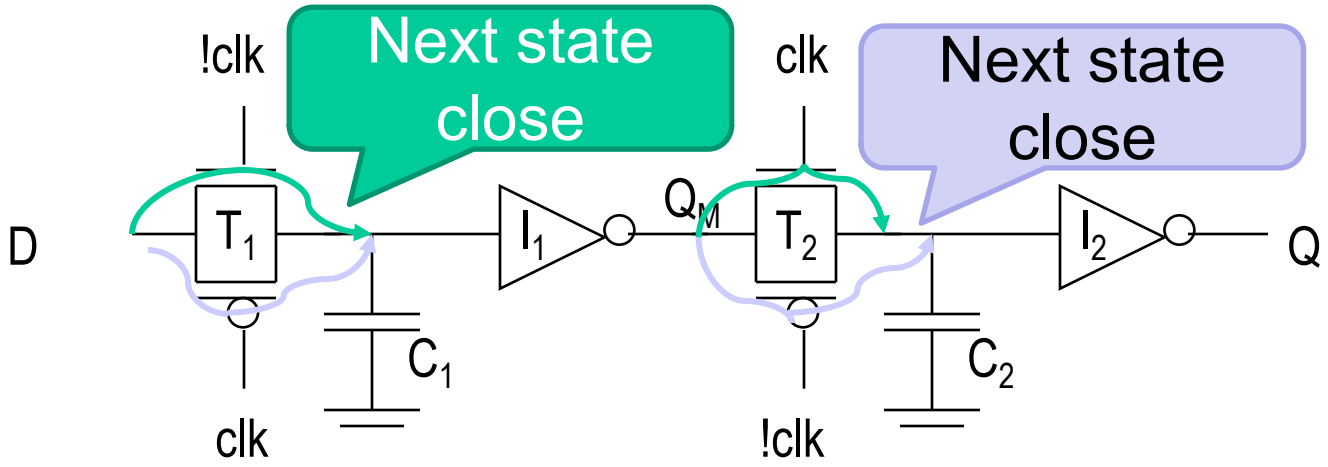


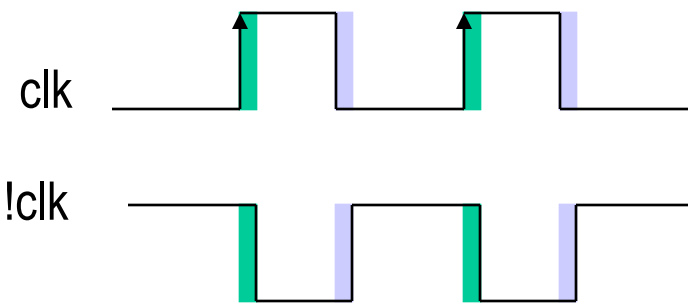
t_{su} = t_{pd_tx}
t_{hold} = zero
$t_{c\text{-}q}$ = 2 t_{pd_inv} + t_{pd_tx}

master transparent
slave hold



Dynamic ET FF Race Conditions





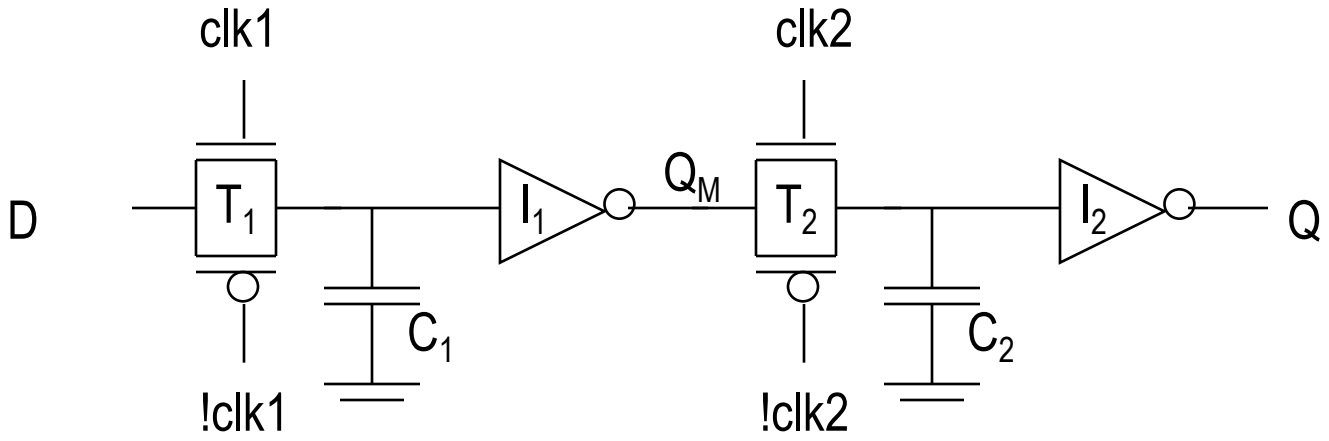
0-0 overlap race condition

$t_{overlap0\text{-}0} < t_{T1} + t_{I1} + t_{T2}$

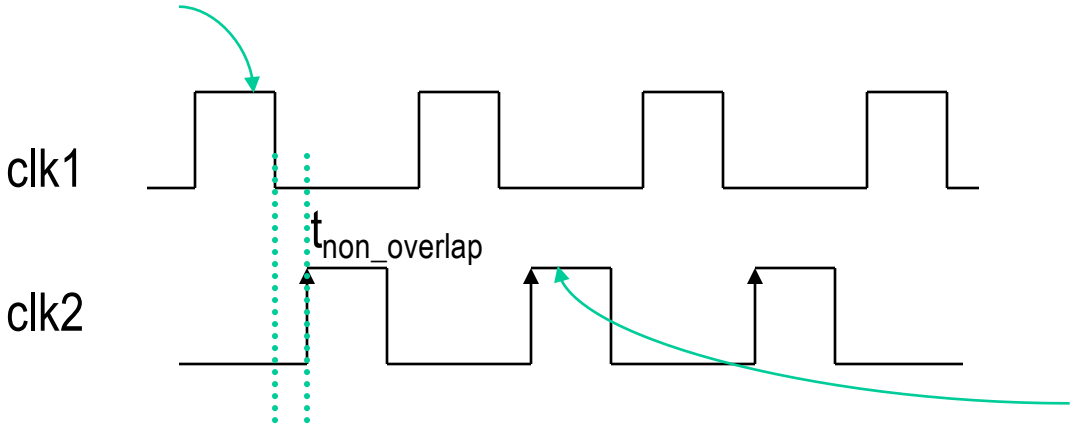
1-1 overlap race condition

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

Fix 1: Dynamic Two-Phase ET FF

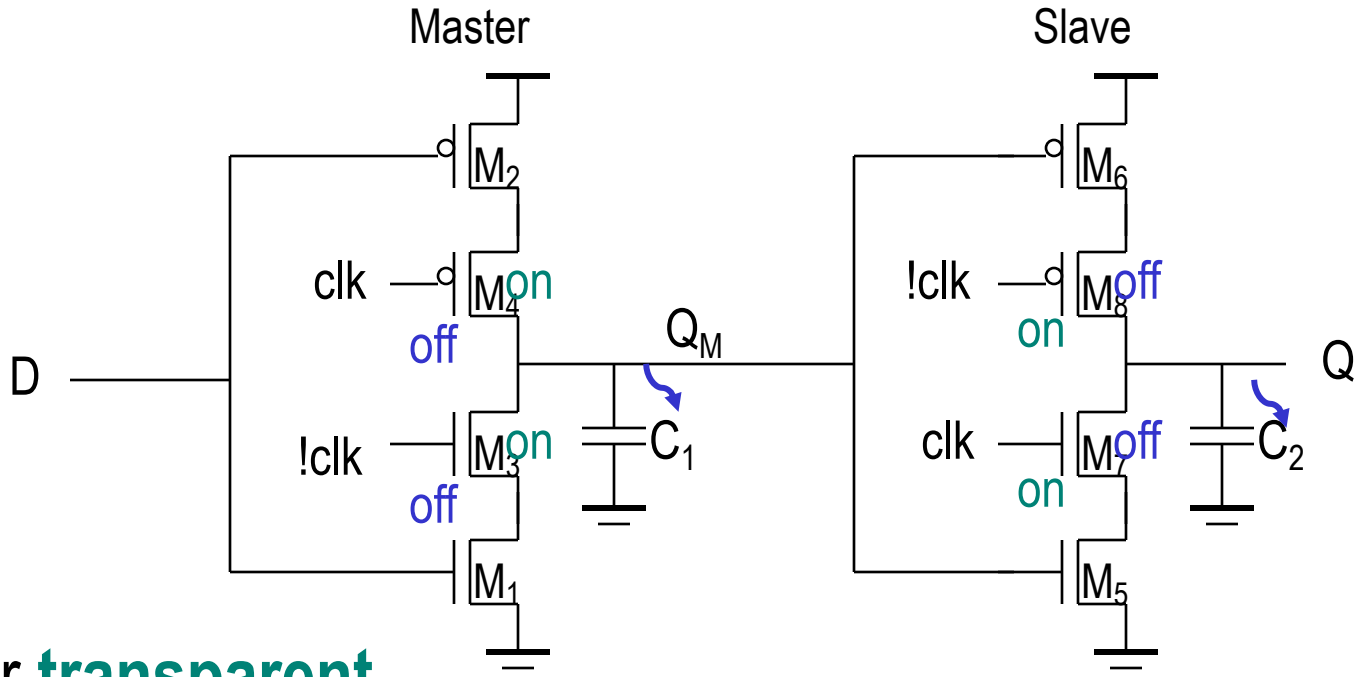


master transparent
slave hold

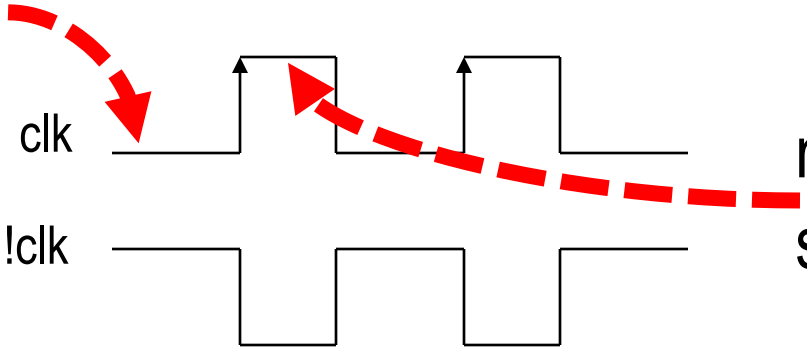


master hold
slave transparent

C²MOS (Clocked CMOS) ET Flipflop

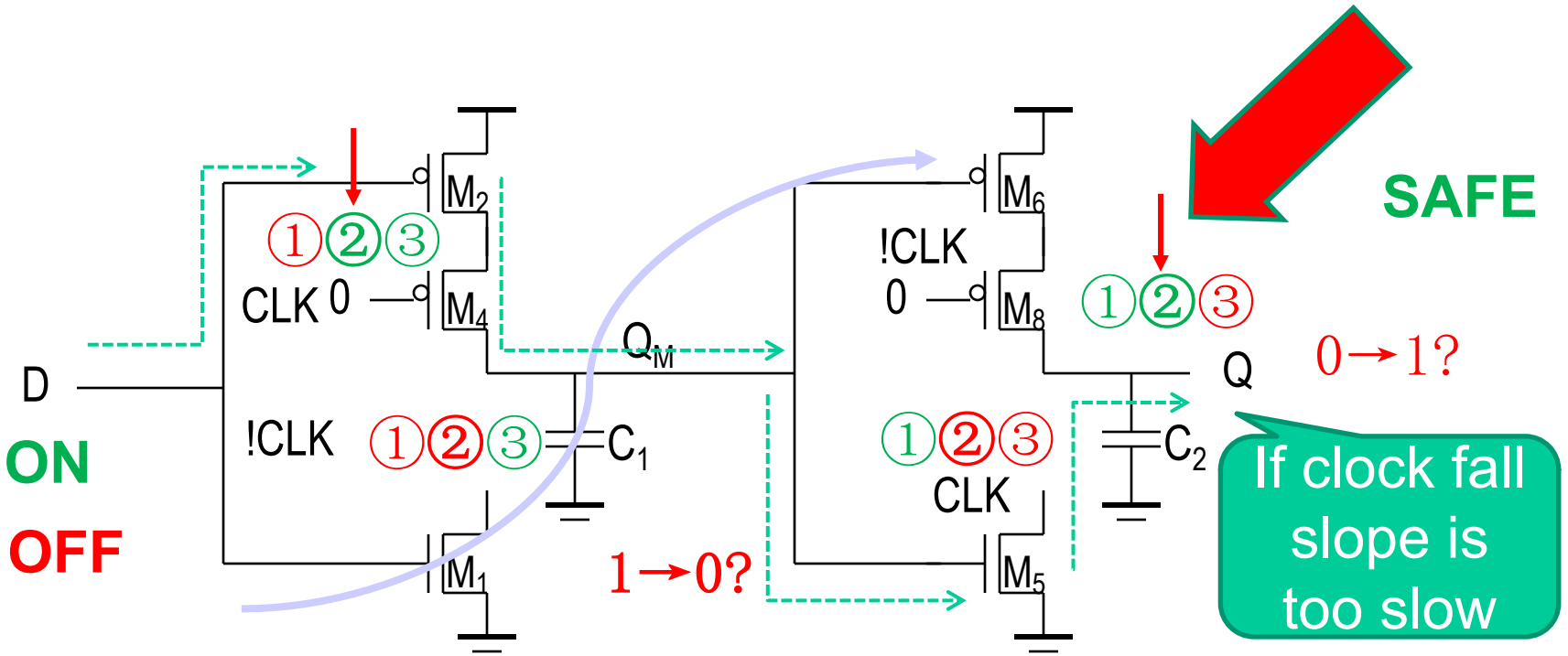


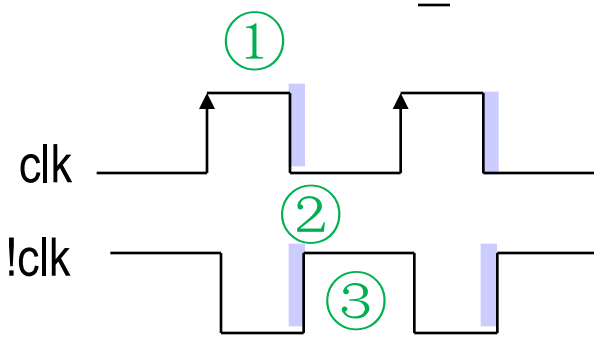
master **transparent**
slave **hold**



master **hold**
slave **transparent**

C²MOS FF 0-0 Overlap Case

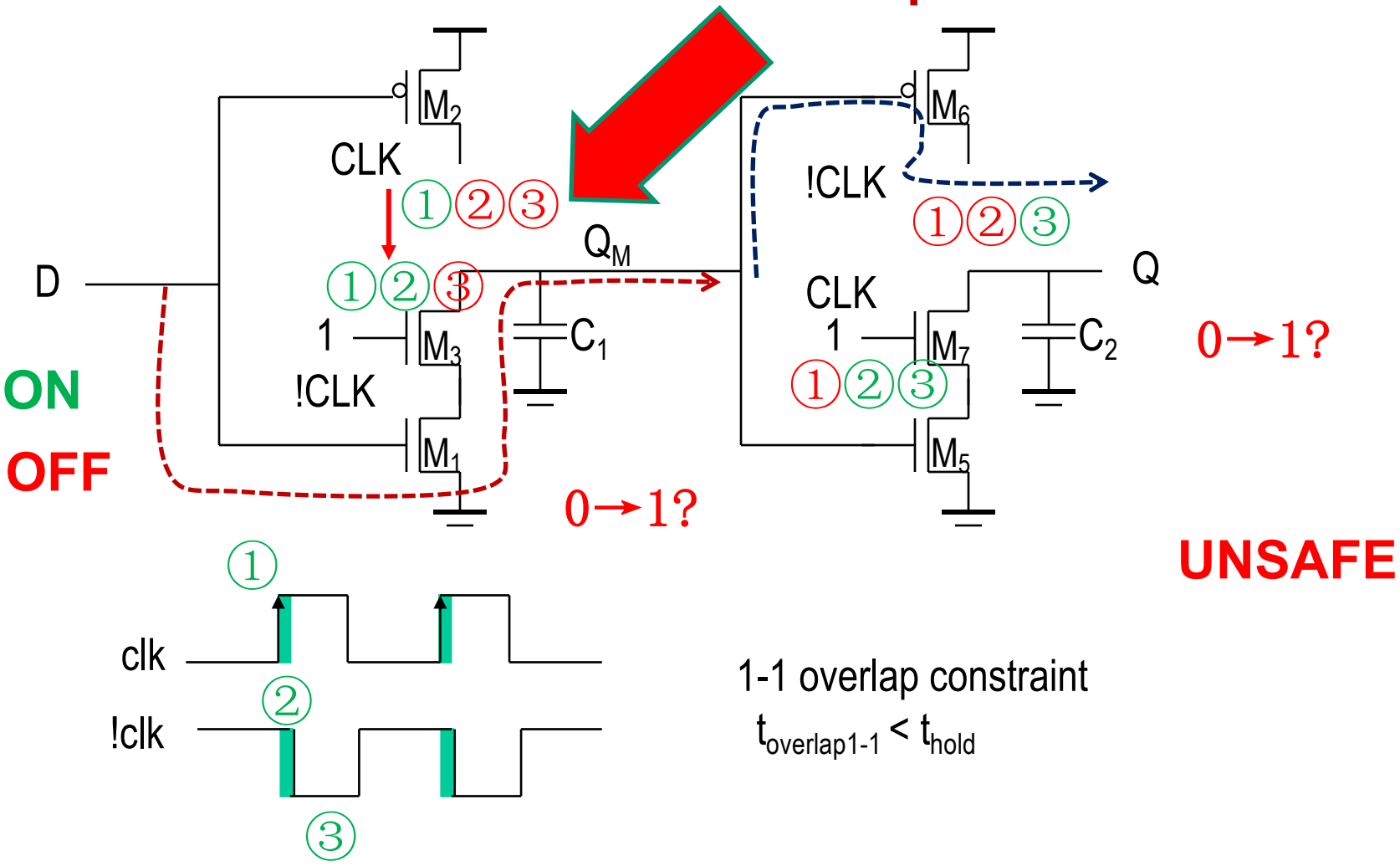




Clock-skew insensitive as long as the rise and fall times of the clock edges are sufficiently small

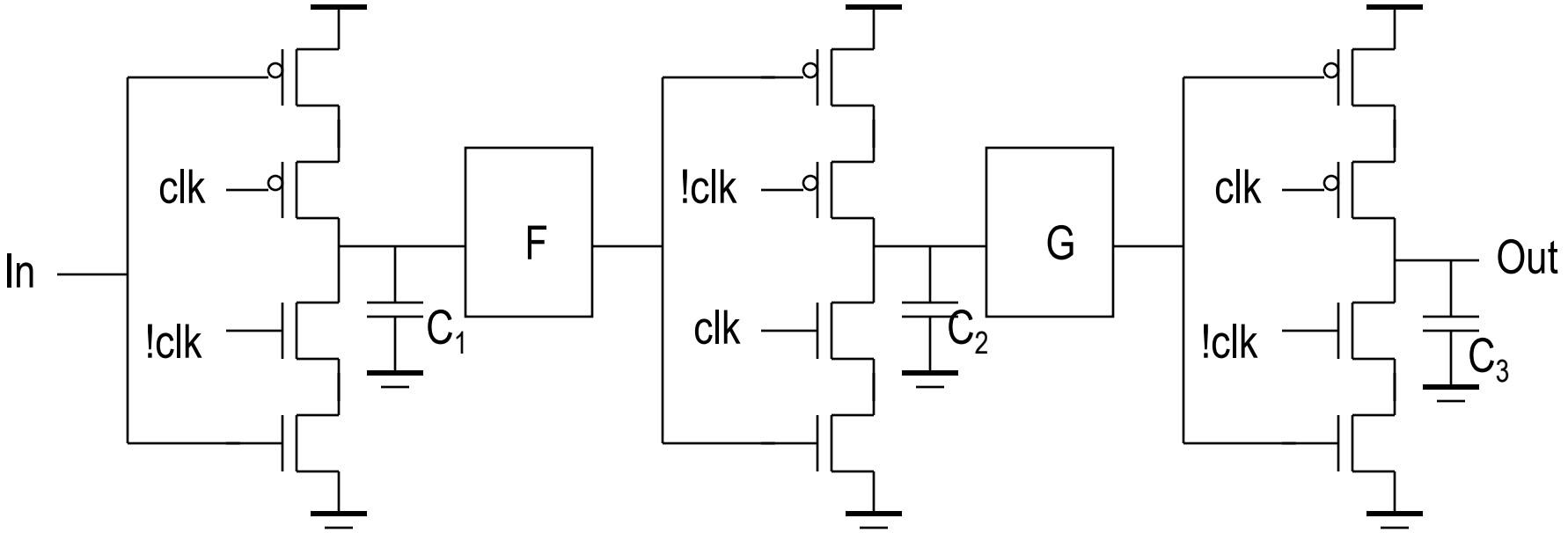
Hold state…

C²MOS FF 1-1 Overlap Case



Setup state…

Pipelining using C²MOS

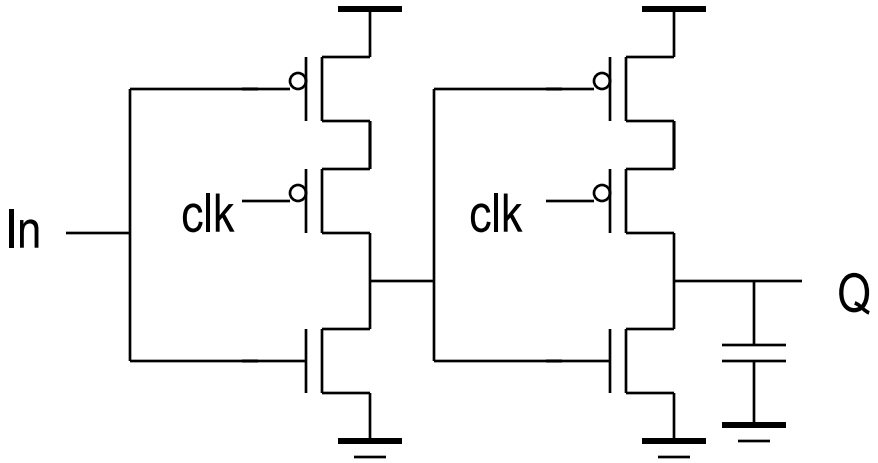


aka NORA (NO RAce) Logic

What are the constraints on F and G?

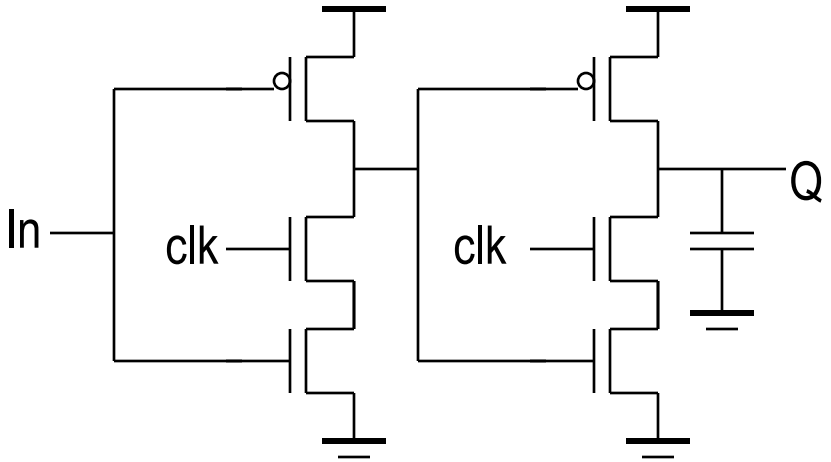
Fix 3: True Single Phase Clocked (TSPC) Latches

Negative Latch



hold when clk = 1
transparent when clk = 0

Positive Latch



transparent when clk = 1
hold when clk = 0

Traditional Domino Circuits

- Ping-pang approach for overlapping the precharge time

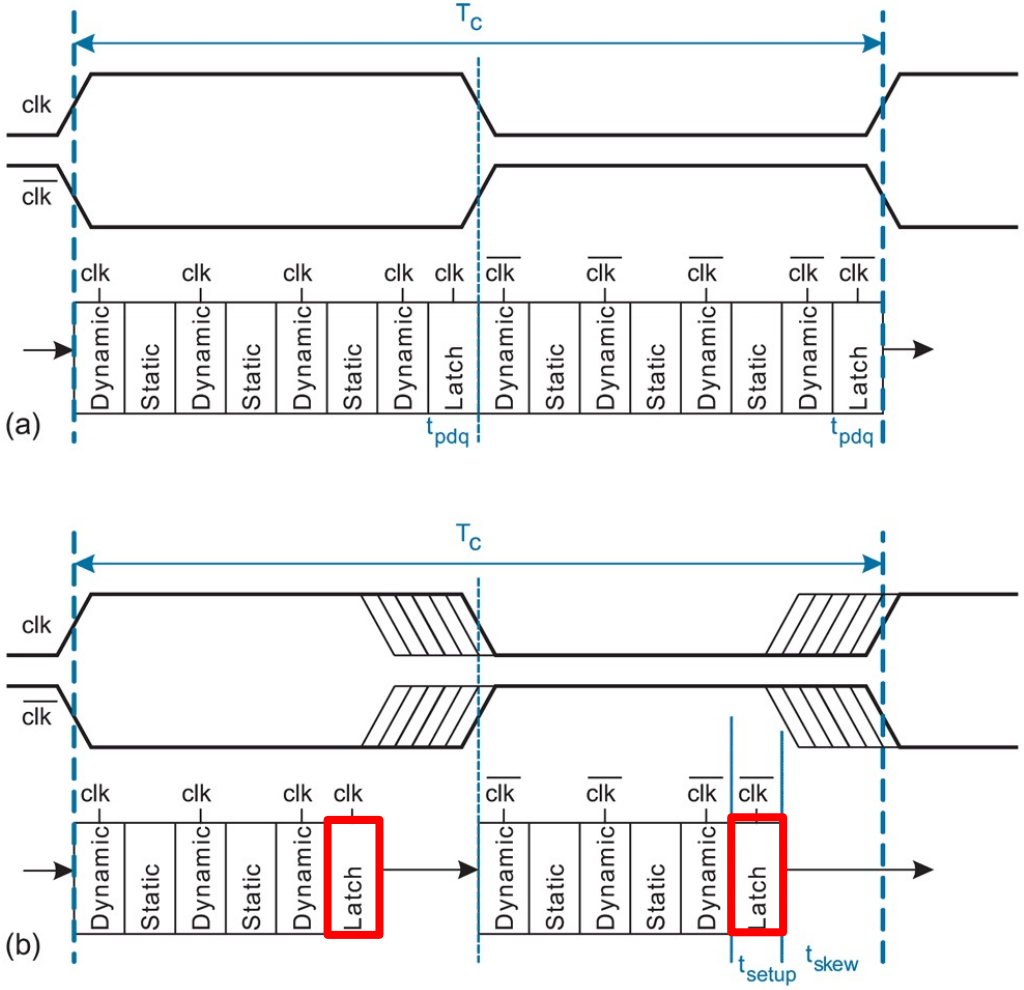


FIG 7.43 Traditional domino circuits

Skew-tolerant Domino Circuits

- Latch function
  - Prevent nonmonotonic signals from entering the next domino gate while it evaluates
  - Hold the results of the half-cycle while it precharges and the next half-cycle evaluates

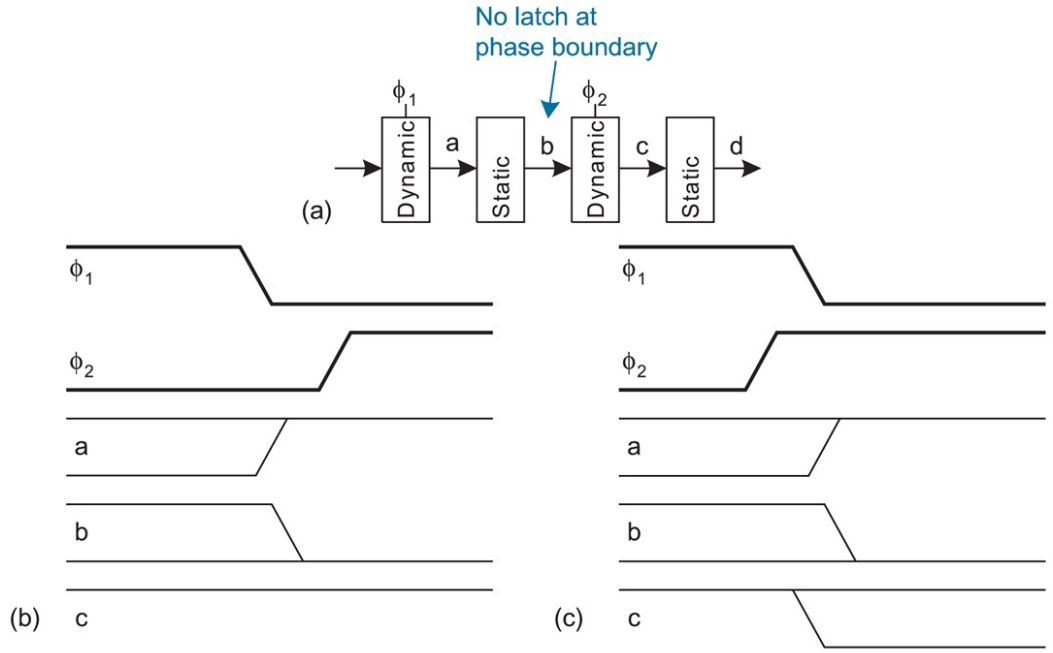


FIG 7.44 Eliminating latches in skew-tolerant domino circuits

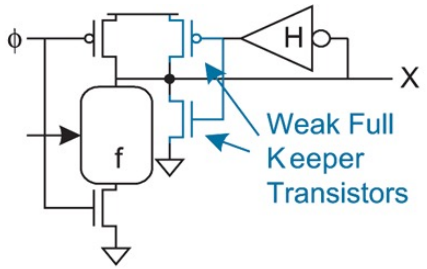


FIG 7.45 Full keeper