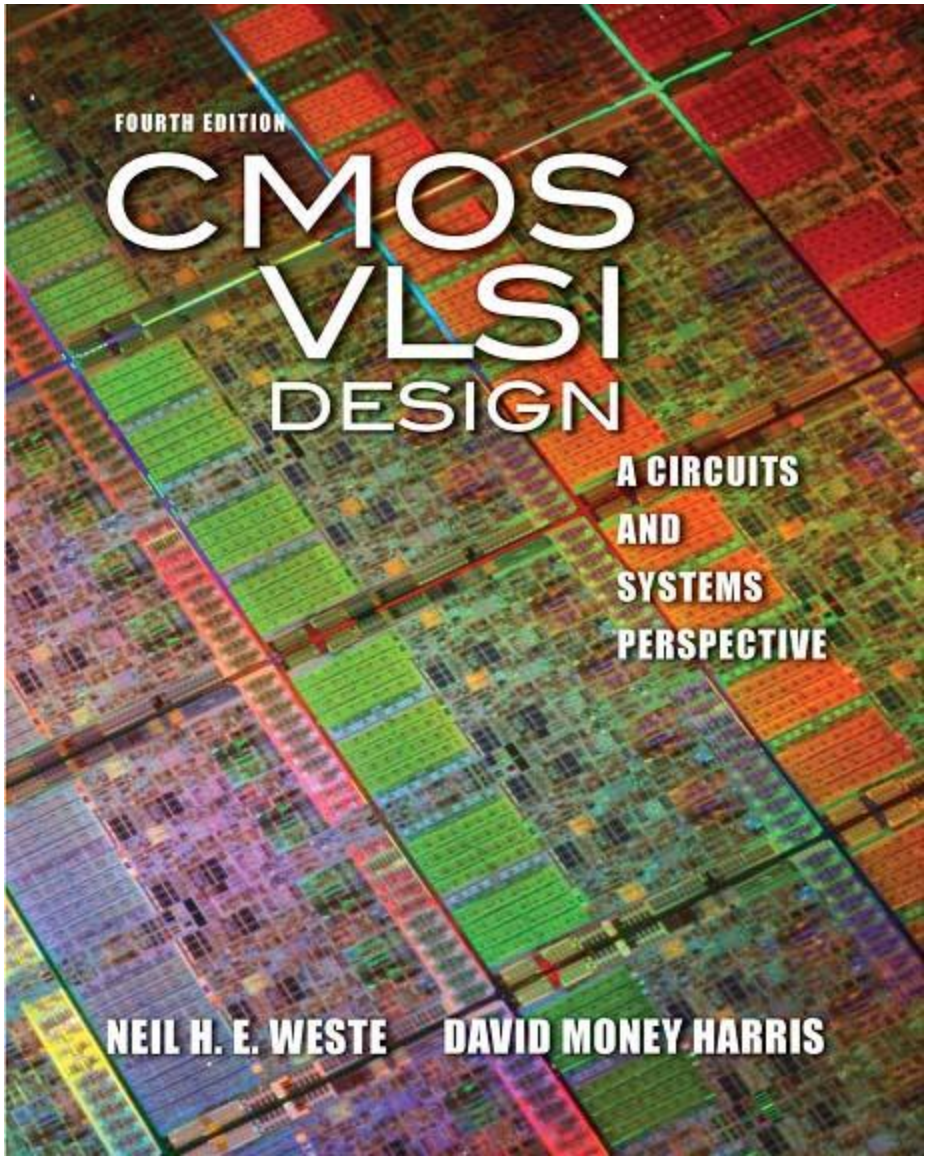


Sequential Circuit Design

Dr. Bassam Jamil

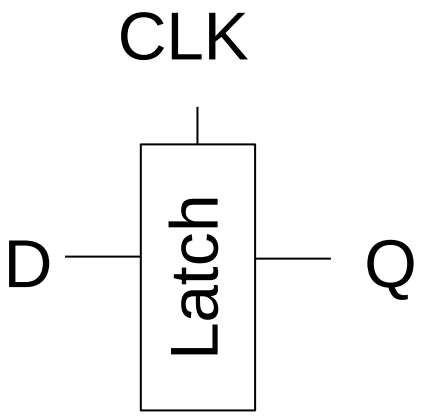
Adopted from slides of the textbook

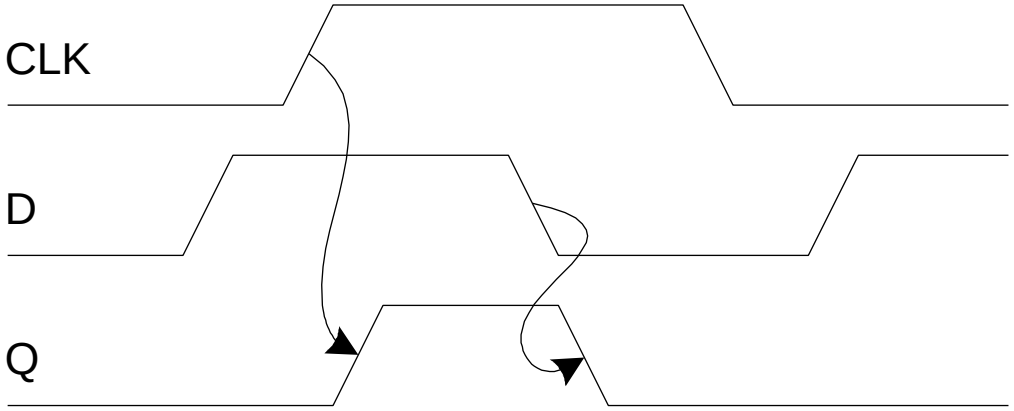
Outline

- Latch
  - Operation
  - Dynamic and Static Latches
- Flip-Flop
  - Operation
  - Dynamic and Static Flip Flops
- Adding (Enable/ Reset / Set) to Latch/Flip-Flop
- Timing

D Latch

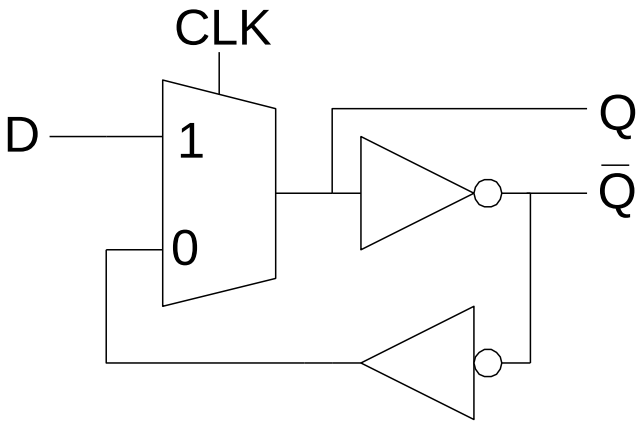
- When CLK = 1, latch is *transparent*
  - D flows through to Q like a buffer
- When CLK = 0, the latch is *opaque*
  - Q holds its old value independent of D
- a.k.a. *transparent latch* or *level-sensitive latch*

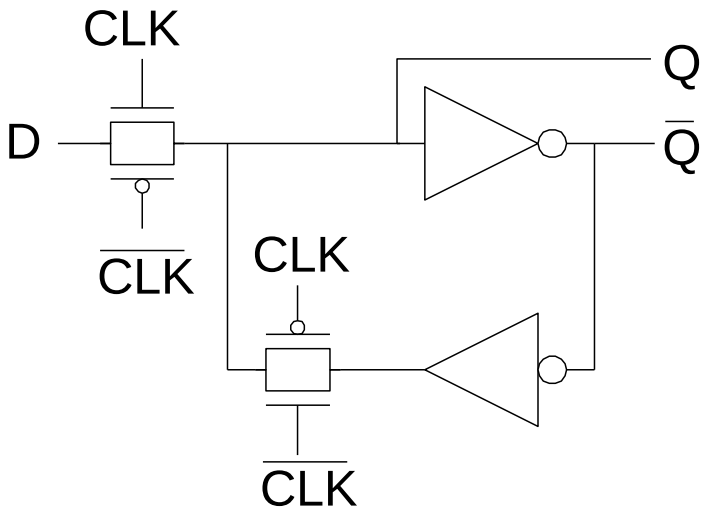




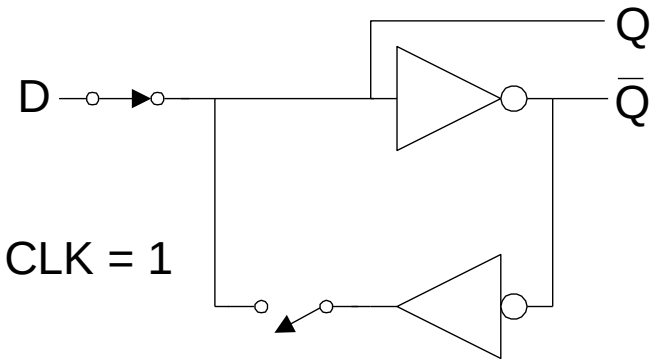
D Latch Design

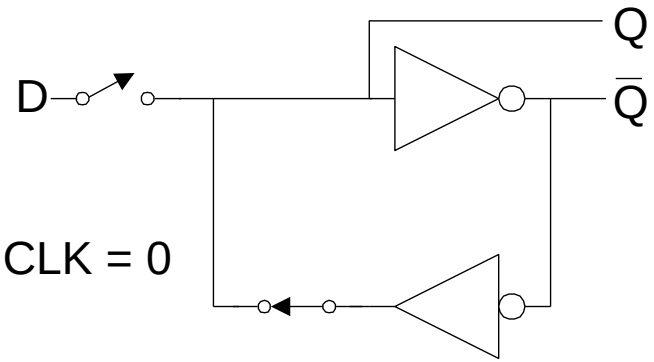
- Multiplexer chooses D or old Q





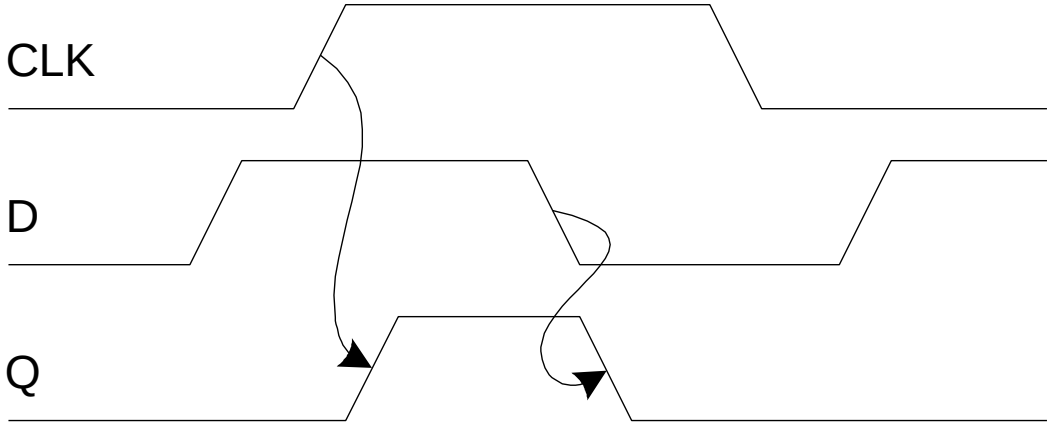
D Latch Operation





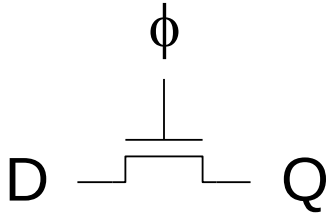
Write Data

Store Data



Dynamic Latch: Pass Transistor

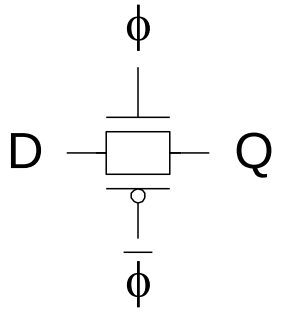
- Pass Transistor Latch
- Pros
  - + Tiny
  - + Low clock load
- Cons
  - – V_t drop
  - – nonrestoring
  - – backdriving
  - – output noise sensitivity
  - – diffusion input
  - – **Dynamic → Leakage**



Used in 1970's

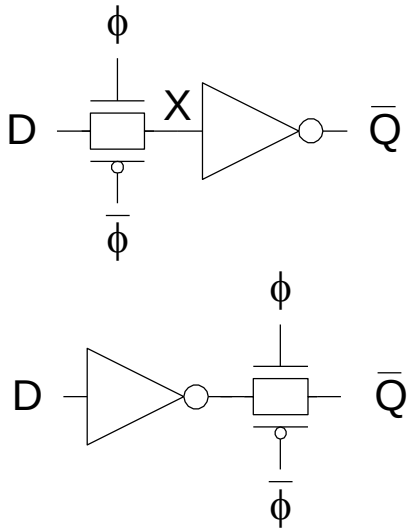
Dynamic Latch: Transmission Gate

- Transmission gate
  - \+ No V_t drop
  - \- Requires inverted clock
  - \- **Dynamic → Leakage**



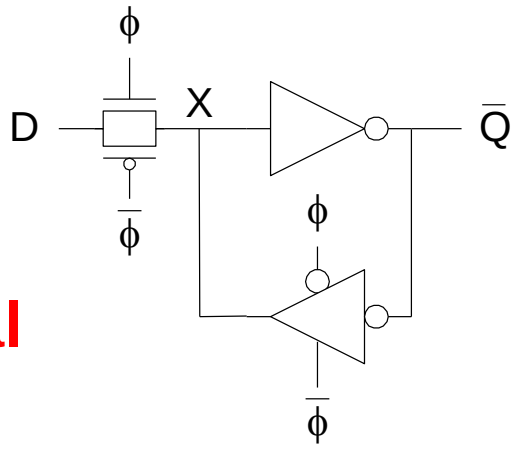
Dynamic Latch: Transmission Gate

- Inverting buffer
  - + Restoring
  - + No backdriving
  - + Fixes either
    - Output noise sensitivity
    - Or diffusion input
  - – Inverted output
  - – **Dynamic → Leakage**



Static Latch Design

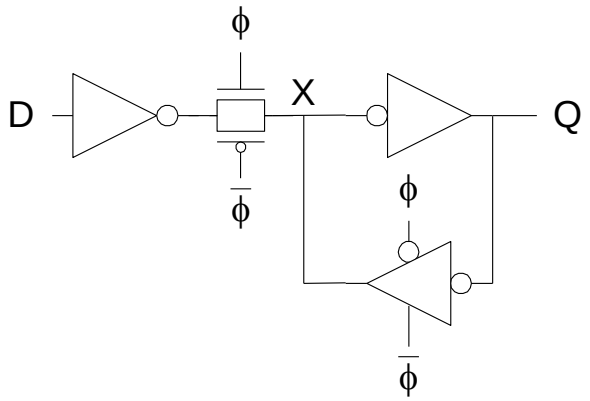
- Tristate feedback
  - \+ Static
  - – Backdriving risk



- **Static latches are now essential because of leakage**

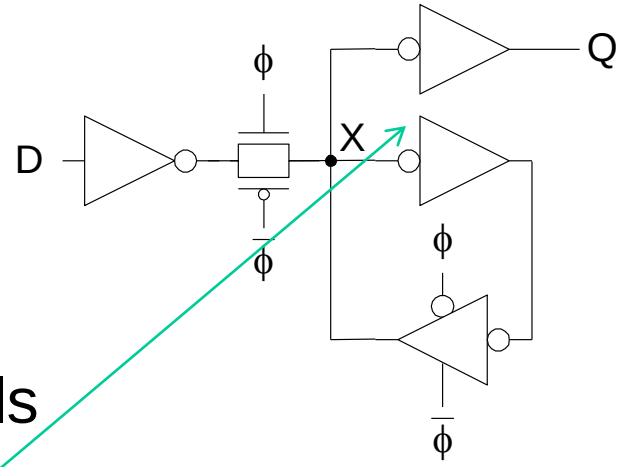
Static Latch Design

- Buffered input
  - + Fixes diffusion input
  - + Noninverting



Static Latch Design

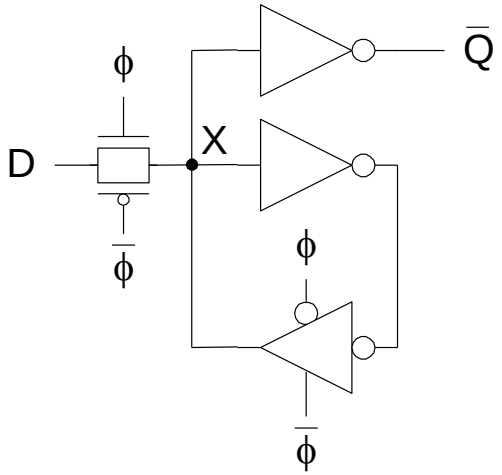
- Buffered output
  - + No backdriving

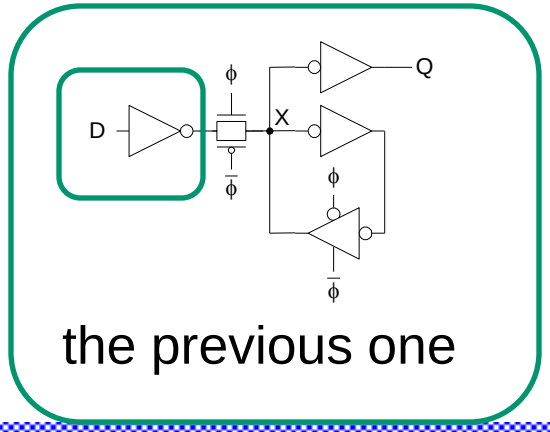


- Widely used in standard cells
  - + Very robust (most important)
  - - Rather large (extra inverter)
  - - Rather slow (1.5 – 2 FO4 delays)
  - - High clock loading

Latch Design

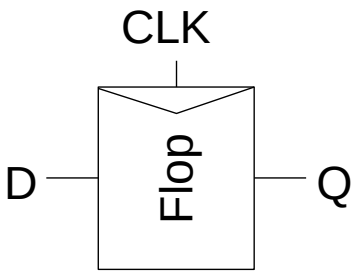
- Datapath latch
  - \+ smaller
  - \+ faster
  - \- unbuffered input

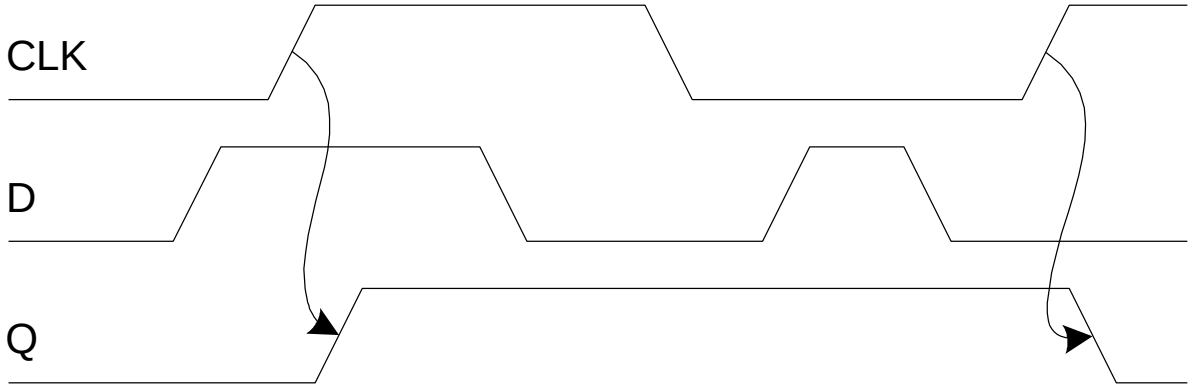




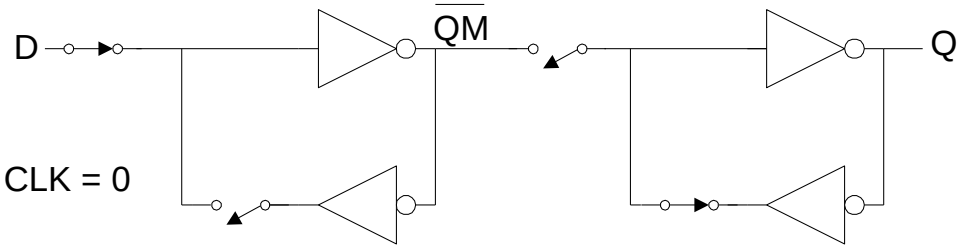
D Flip-flop

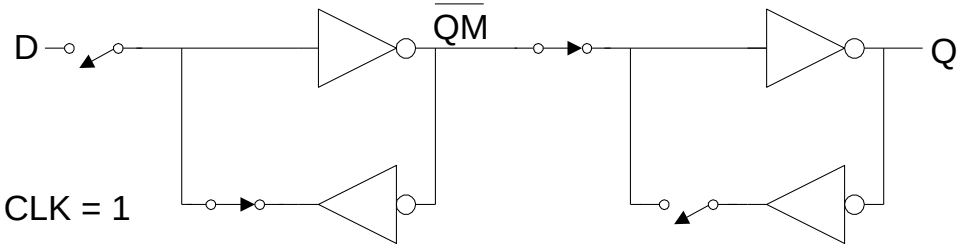
- When CLK rises, D is copied to Q
- At all other times, Q holds its value
- a.k.a. *positive edge-triggered flip-flop*, *master-slave flip-flop*

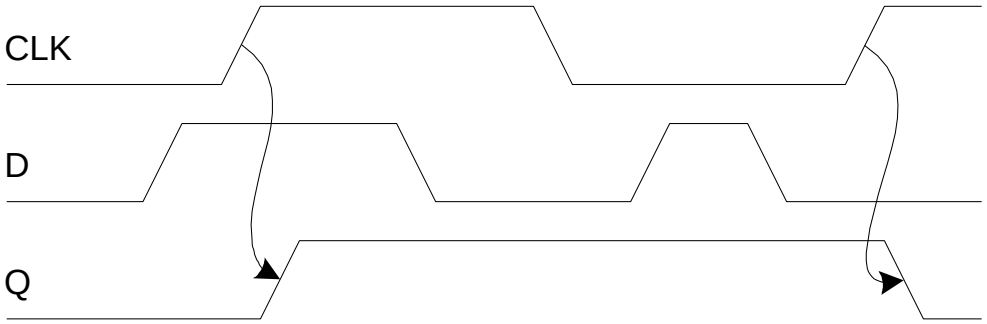




D Flip-flop Operation

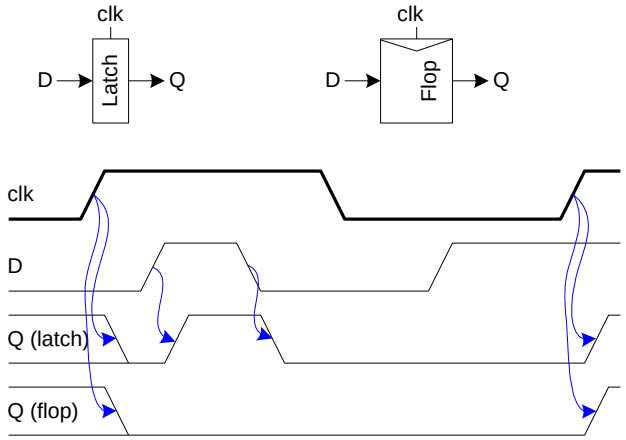






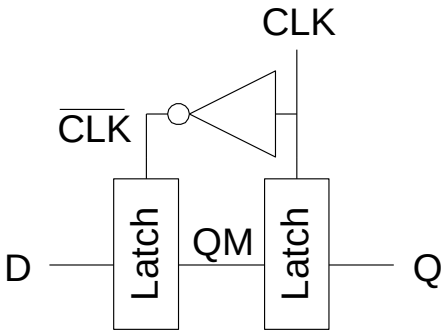
Sequencing Elements

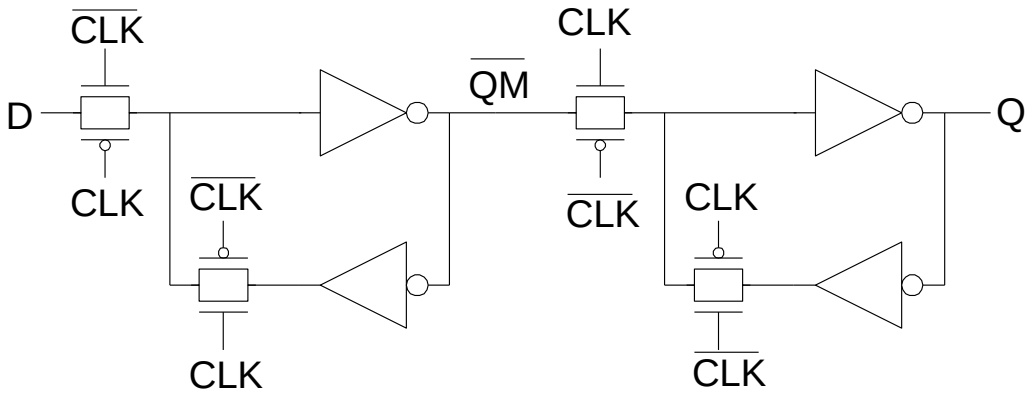
- **Latch**: Level sensitive
  - a.k.a. transparent latch, D latch
- **Flip-flop**: edge triggered
  - A.k.a. master-slave flip-flop, D flip-flop, D register
- Timing Diagrams
  - Transparent
  - Opaque
  - Edge-trigger



D Flip-flop Design

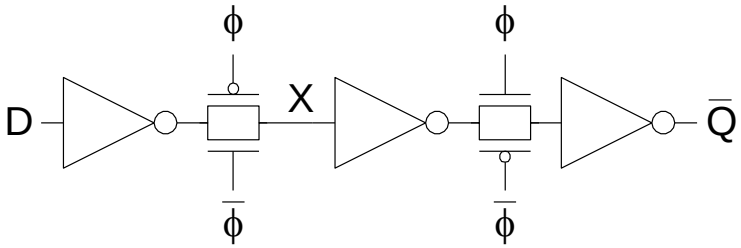
- Built from master and slave D latches



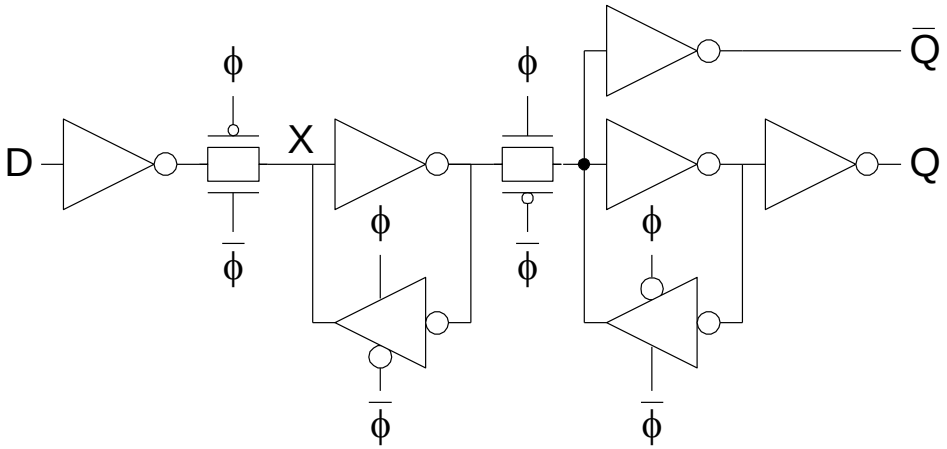


Flip-Flop Design

- Flip-flop is built as pair of back-to-back latches



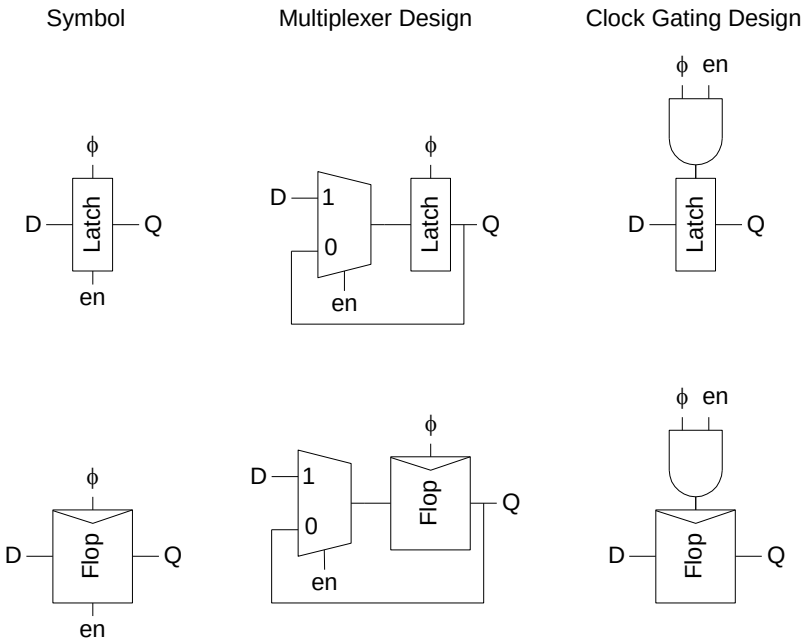
Dynamic FF



Static FF

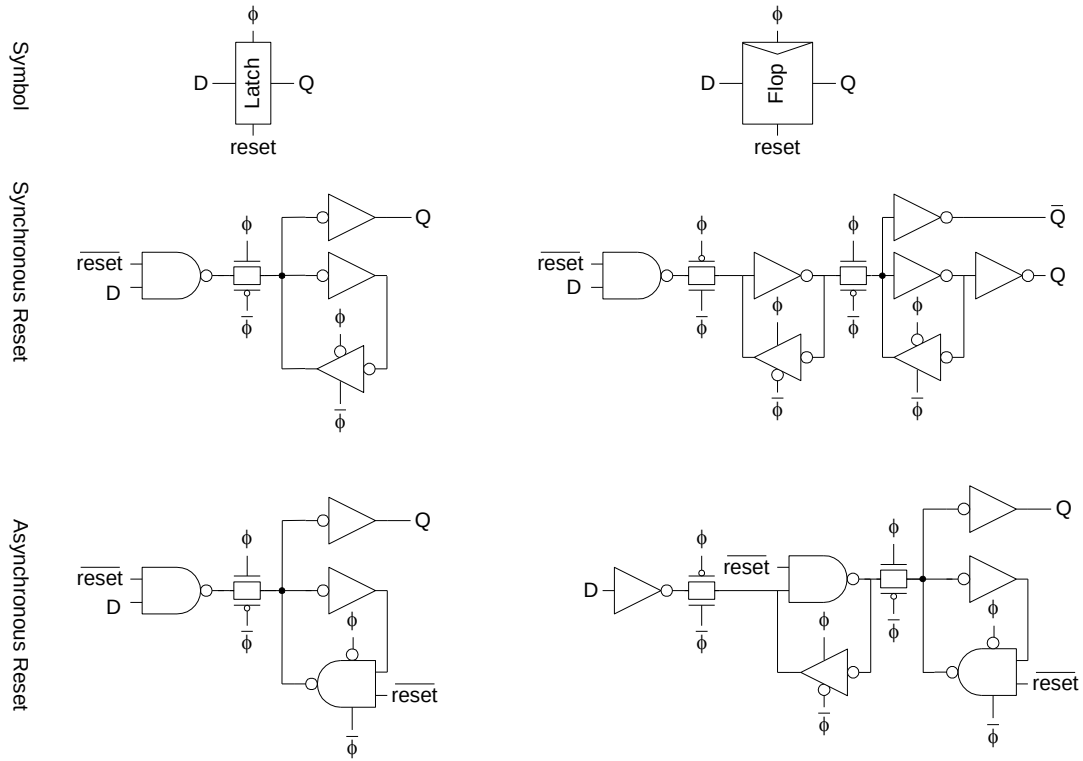
Design of FF/Latch with Enable

- Enable: ignore clock when en = 0
  - Mux: increase latch D-Q delay
  - Clock Gating: increase en setup time, skew



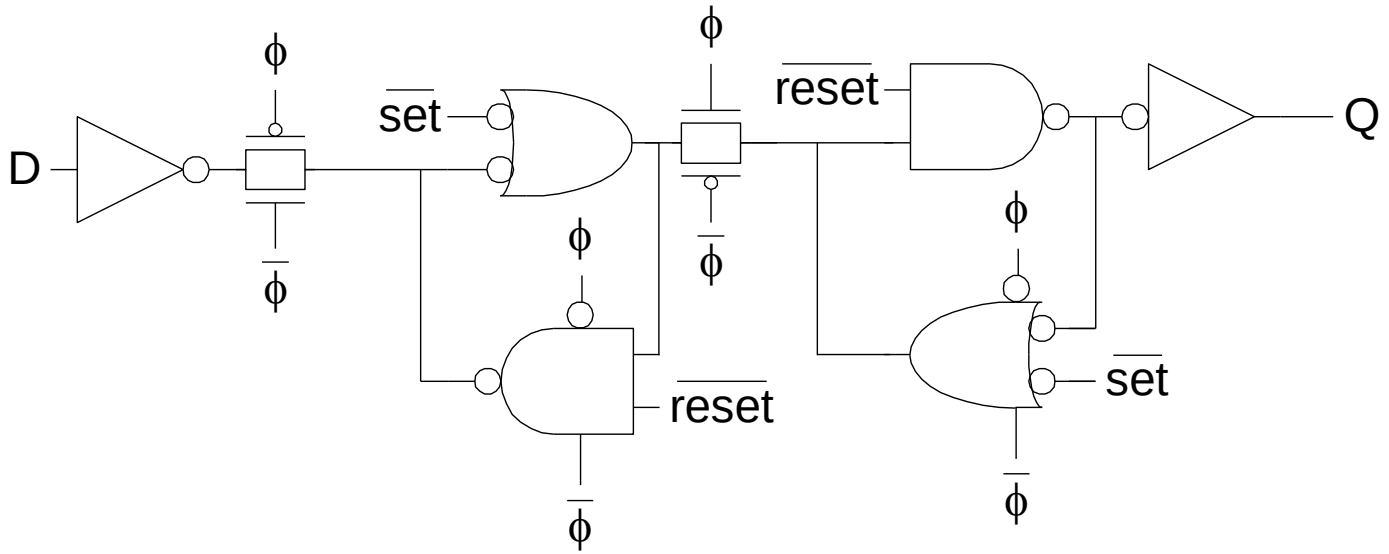
Design of FF/Latch with Reset

- Force output low when reset asserted
- Synchronous vs. asynchronous



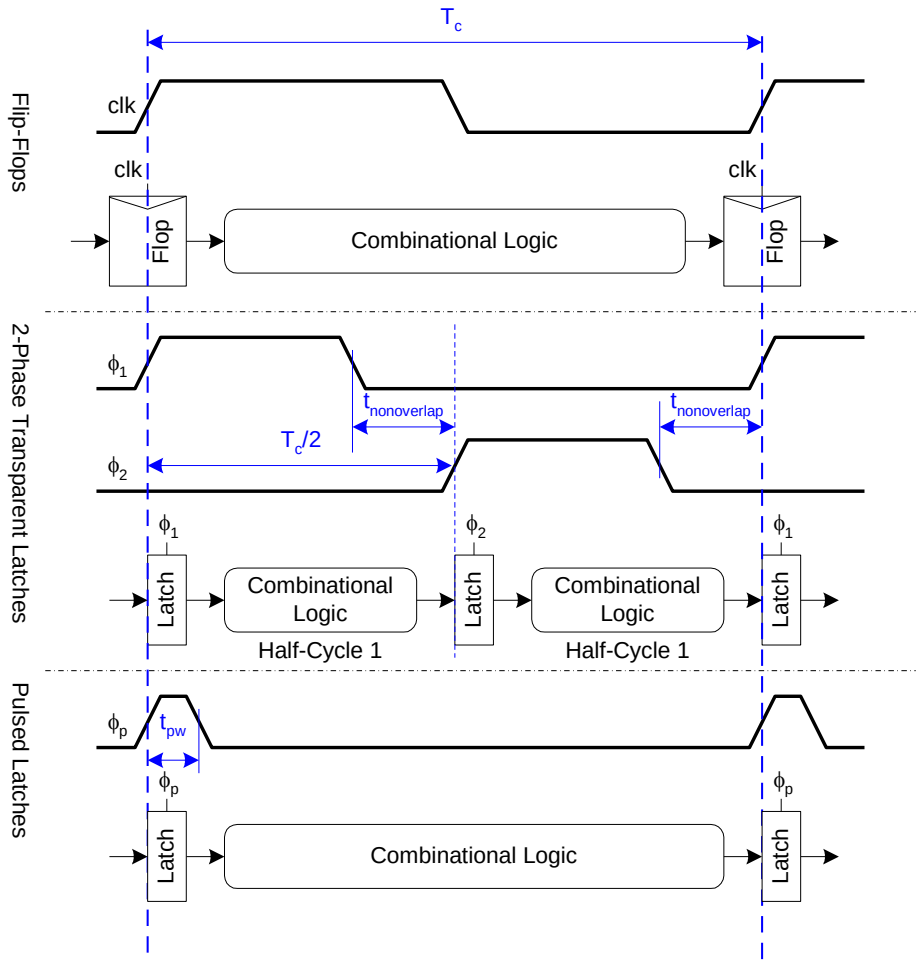
Design of FF with Set / Reset

- Set forces output high when enabled
- Flip-flop with asynchronous set and reset



Sequencing Methods

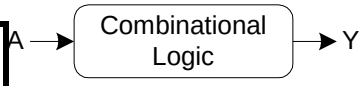
- Flip-flops
- 2-Phase Latches
- Pulsed Latches

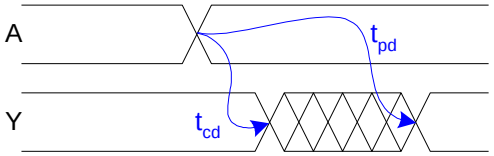


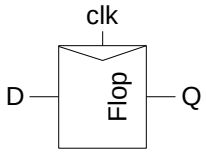
Timing Diagrams

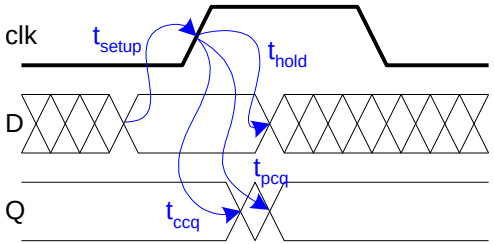
Contamination and Propagation Delays

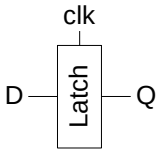
t_{pd}	Logic Prop. Delay
t_{cd}	Logic Cont. Delay
t_{pcq}	Latch/Flop Clk->Q Prop. Delay
t_{ccq}	Latch/Flop Clk->Q Cont. Delay
t_{pdq}	Latch D->Q Prop. Delay
t_{cdq}	Latch D->Q Cont. Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

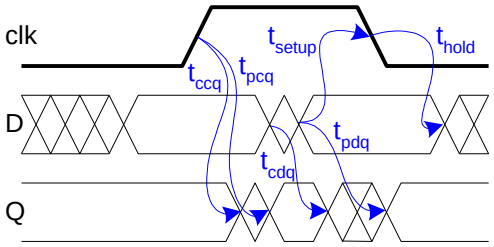






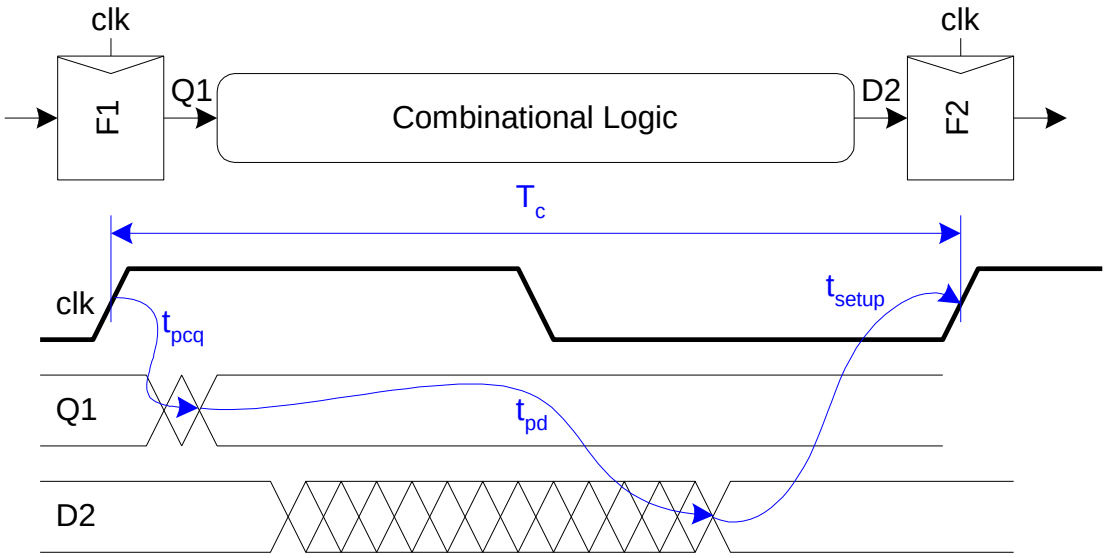






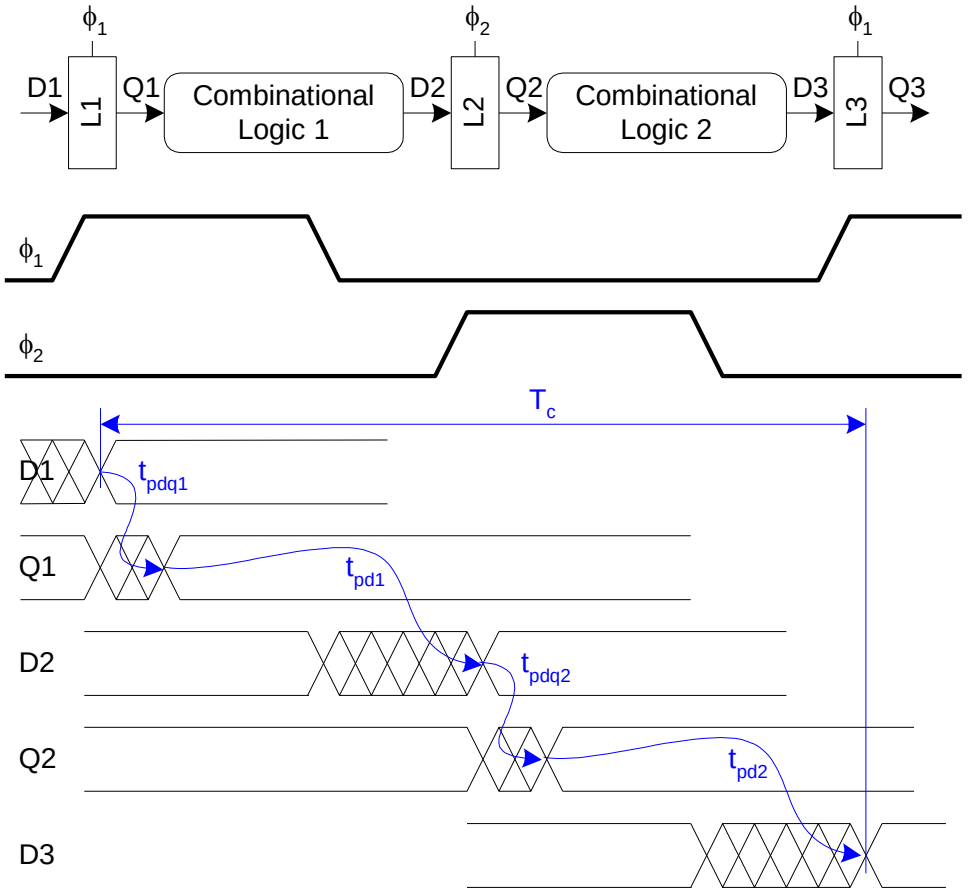
Max-Delay: Flip-Flops

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



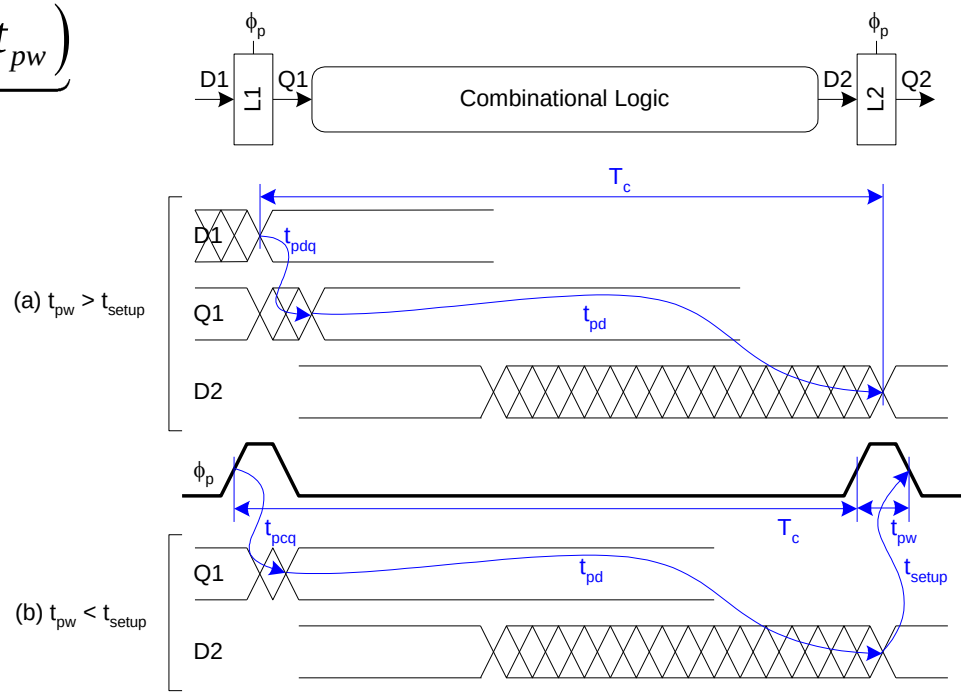
Max Delay: 2-Phase Latches (read)

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



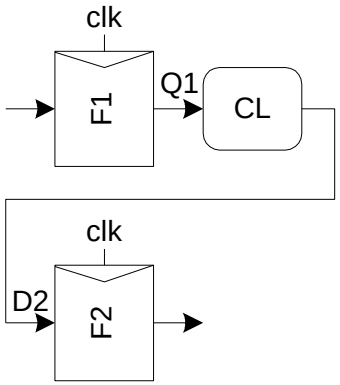
Max Delay: Pulsed Latches (read)

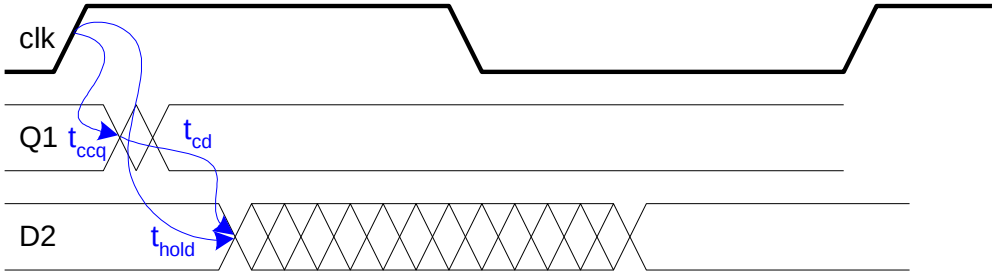
$$t_{pd} \leq T_c - \underbrace{\max\left(t_{pdq}, t_{pcq} + t_{\text{setup}} - t_{pw}\right)}_{\text{sequencing overhead}}$$



Min-Delay: Flip-Flops

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





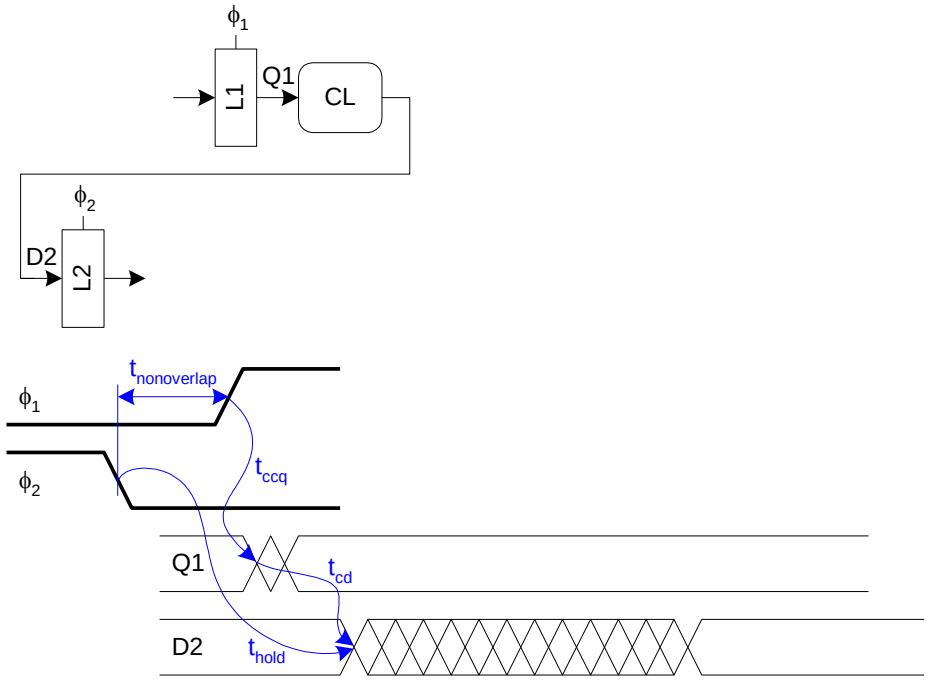
Min-Delay: 2-Phase Latches (read)

$t_{cd1}, t_{cd2} \geq$

Hold time reduced by nonoverlap

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

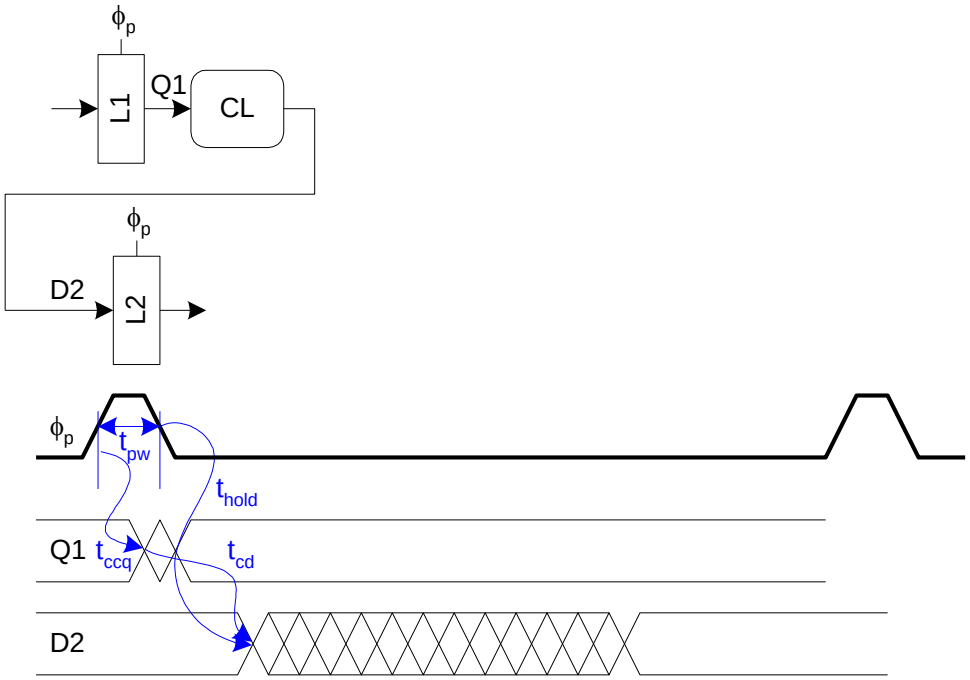
But a flop is made of two latches!



Min-Delay: Pulsed Latches (read)

$t_{cd} \geq$

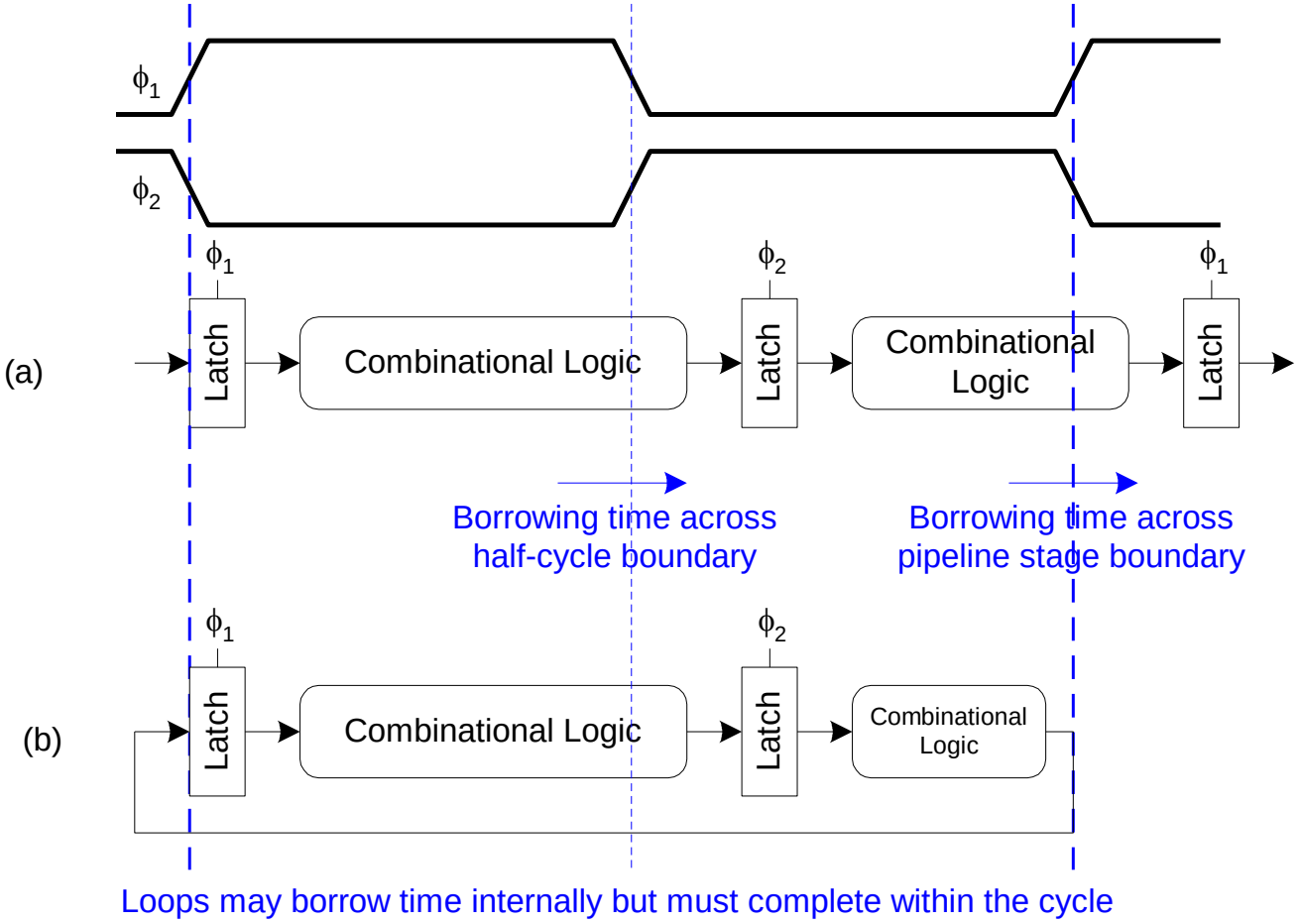
Hold time increased by pulse width



Time Borrowing (read)

- In a flop-based system:
  - Data launches on one rising edge
  - Must setup before next rising edge
  - If it arrives late, system fails
  - If it arrives early, time is wasted
  - Flops have hard edges
- In a latch-based system
  - Data can pass through latch while transparent
  - Long cycle of logic can borrow time into next
  - As long as each loop completes in one cycle

Time Borrowing Example (read)



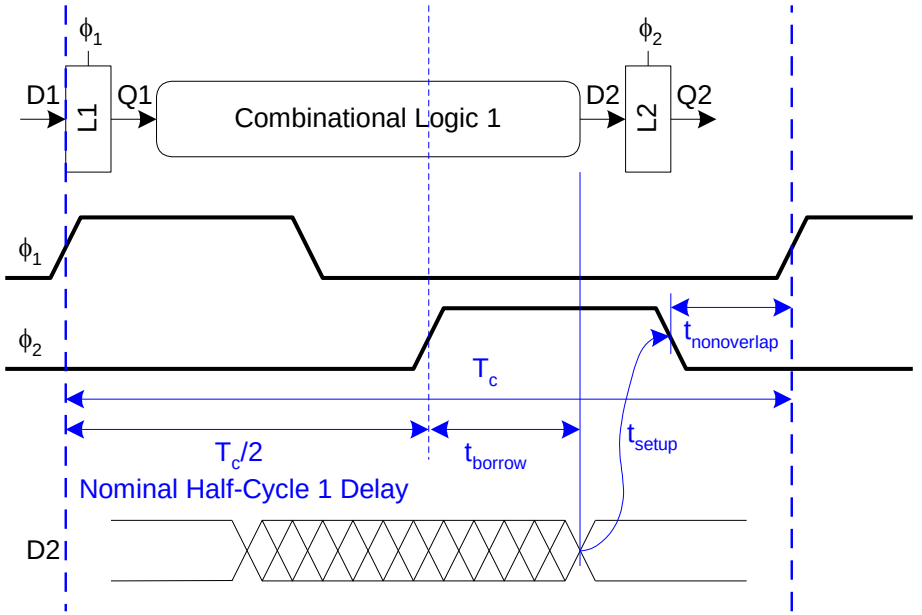
How Much Borrowing? (read)

2-Phase Latches

$$t_{\text{borrow}} \leq \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}}\right)$$

Pulsed Latches

$$t_{\text{borrow}} \leq t_{pw} - t_{\text{setup}}$$



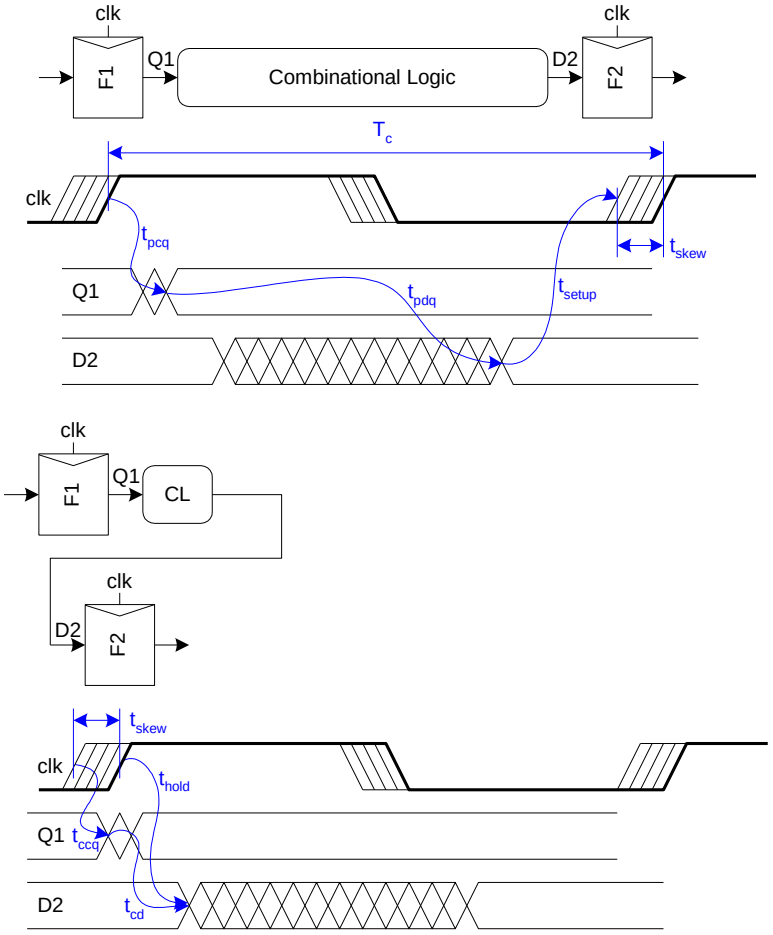
Clock Skew

- We have assumed zero clock skew
- Clocks really have uncertainty in arrival time
  - Decreases maximum propagation delay
  - Increases minimum contamination delay
  - Decreases time borrowing

Skew: Flip-Flops

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

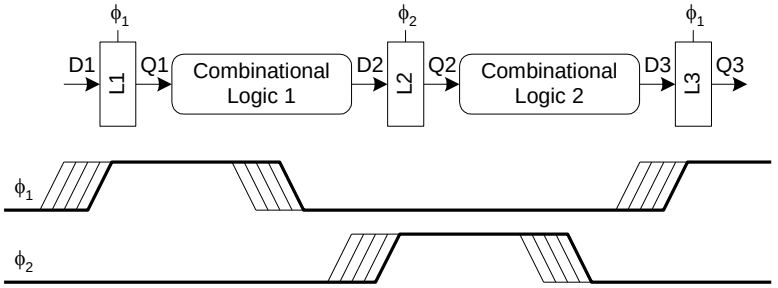
$$t_{cd} \geq t_{\text{hold}} - t_{ccq} + t_{\text{skew}}$$



Skew: Latches (read)

2-Phase Latches

$$t_{pd} \leq T_c - \underbrace{\left(2t_{pdq}\right)}_{\text{sequencing overhead}}$$



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

$$t_{\text{borrow}} \leq \frac{T_c}{2} - \left(t_{\text{setup}} + t_{\text{nonoverlap}} + t_{\text{skew}}\right)$$

Pulsed Latches

$$t_{pd} \leq T_c - \underbrace{\max\left(t_{pdq}, t_{pcq} + t_{\text{setup}} - t_{pw} + t_{\text{skew}}\right)}_{\text{sequencing overhead}}$$

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

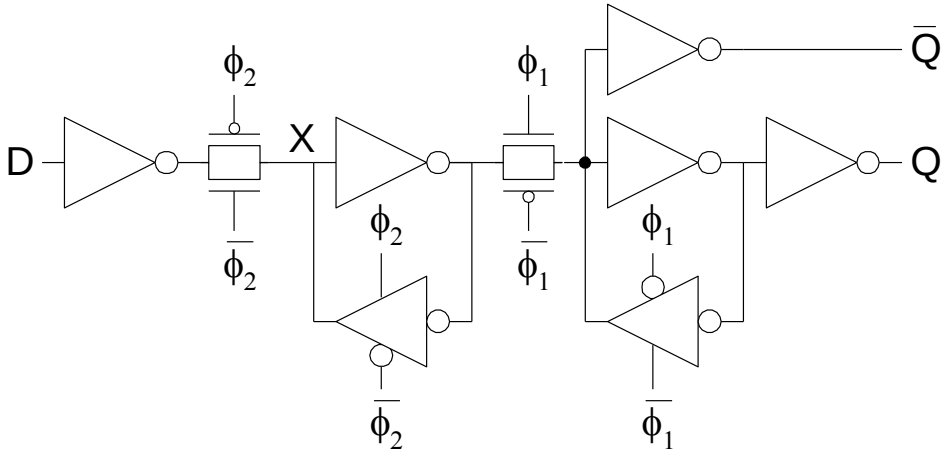
$$t_{\text{borrow}} \leq t_{pw} - \left(t_{\text{setup}} + t_{\text{skew}}\right)$$

Two-Phase Clocking

- If setup times are violated, reduce clock speed
- If hold times are violated, chip fails at any speed
- In this class, working chips are most important
  - No tools to analyze clock skew
- An easy way to guarantee hold times is to use 2-phase latches with big nonoverlap times
- Call these clocks ϕ_1, ϕ_2 (ph1, ph2)

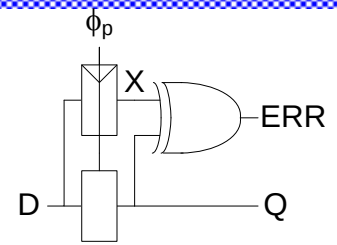
Safe Flip-Flop

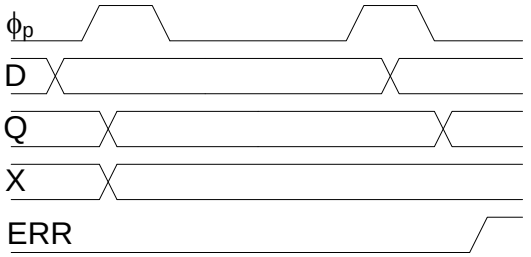
- Past years used flip-flop with nonoverlapping clocks
  - Slow – nonoverlap adds to setup time
  - But no hold times
- In industry, use a better timing analyzer
  - Add buffers to slow signals if hold time is at risk



Adaptive Sequencing (read)

- Designers include timing margin
  - Voltage
  - Temperature
  - Process variation
  - Data dependency
  - Tool inaccuracies





- Alternative: run faster and check for near failures
  - Idea introduced as “Razor”
    - Increase frequency until at the verge of error
    - Can reduce cycle time by ~30%

Summary

- Flip-Flops:
  - Very easy to use, supported by all tools
- 2-Phase Transparent Latches:
  - Lots of skew tolerance and time borrowing
- Pulsed Latches:
  - Fast, some skew tol & borrow, hold time risk

	Sequencing overhead ($T_c - t_{pd}$)	Minimum logic delay t_{cd}	Time borrowing t_{borrow}
Flip-Flops	$t_{pcq} + t_{setup} + t_{skew}$	$t_{hold} - t_{ccq} + t_{skew}$	0
Two-Phase Transparent Latches	$2t_{pdq}$	$t_{hold} - t_{ccq} - t_{nonoverlap} + t_{skew}$ in each half-cycle	$\frac{T_c}{2} - \left(t_{setup} + t_{nonoverlap} + t_{skew}\right)$
Pulsed Latches	$\max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw} + t_{skew}\right)$	$t_{hold} - t_{ccq} + t_{pw} + t_{skew}$	$t_{pw} - \left(t_{setup} + t_{skew}\right)$