Digital Integrated Circuit Design

CMPE 530/630

Sequential Circuit Design

Dr. Cory Merkel

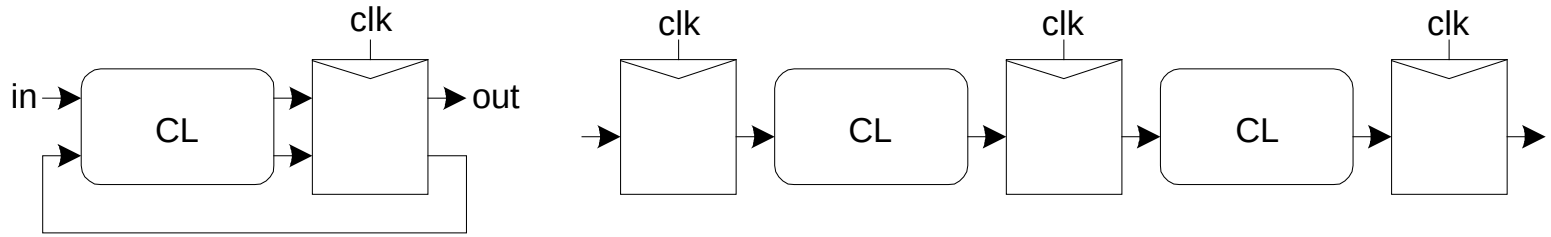
Outline

- Sequencing
- Sequencing Element Design
- Max and Min-Delay
- Clock Skew
- Time Borrowing
- Two-Phase Clocking

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 Cont.

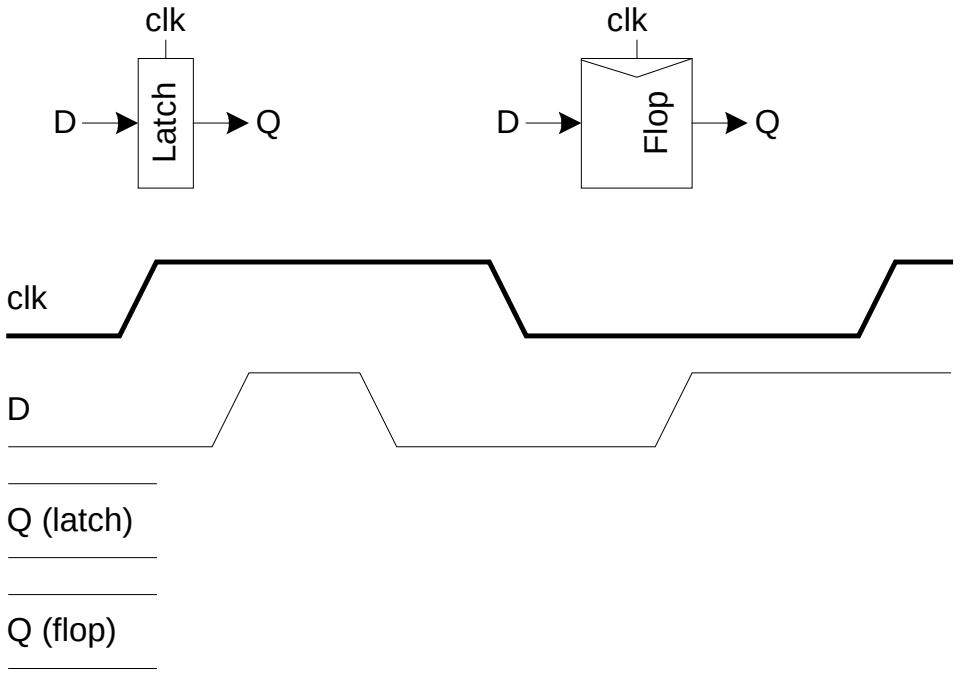
- If tokens moved through pipeline at constant speed, no sequencing elements would be necessary
- Ex: fiber-optic cable
  - Light pulses (tokens) are sent down cable
  - Next pulse sent before first reaches end of cable
  - No need for hardware to separate pulses
  - But *dispersion* sets min time between pulses
- This is called *wave pipelining* in circuits
- In most circuits, dispersion is high
  - Delay fast tokens so they don't catch slow ones.

Sequencing Overhead

- Use flip-flops to delay fast tokens so they move through exactly one stage each cycle.
- Inevitably adds some delay to the slow tokens
- Makes circuit slower than just the logic delay
  - Called sequencing overhead
- Some people call this clocking overhead
  - But it applies to asynchronous circuits too
  - Inevitable side effect of maintaining sequence

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

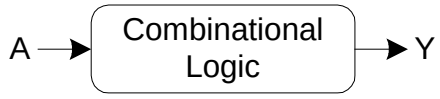


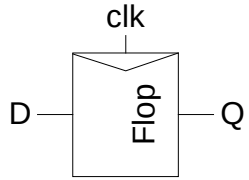
Timing Diagrams

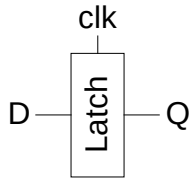
- Key parameters in sequential elements:

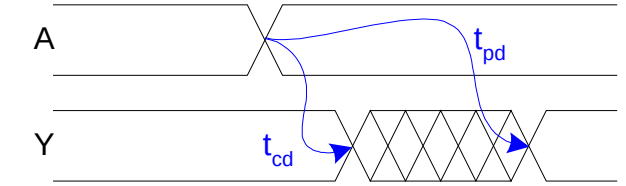
Contamination and
Propagation Delays

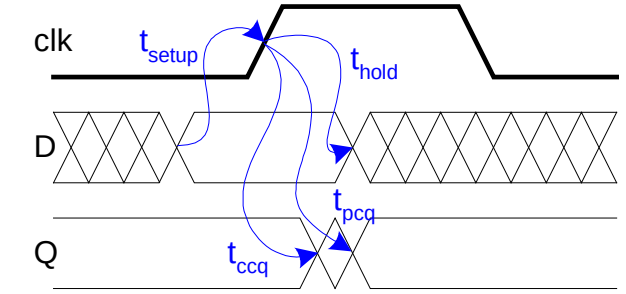
t_{pd}	Logic Prop. Delay
t_{cd}	Logic Cont. Delay
t_{pcq}	Latch/Flop Clk->Q Prop. Delay
t_{ccq}	Latch/Flop Clk->Q Cont. Delay
t_{pdq}	Latch D->Q Prop. Delay
t_{cdq}	Latch D->Q Cont. Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

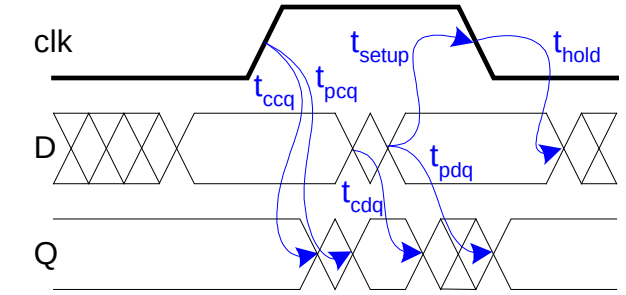






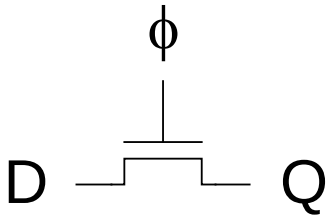






Latch Design

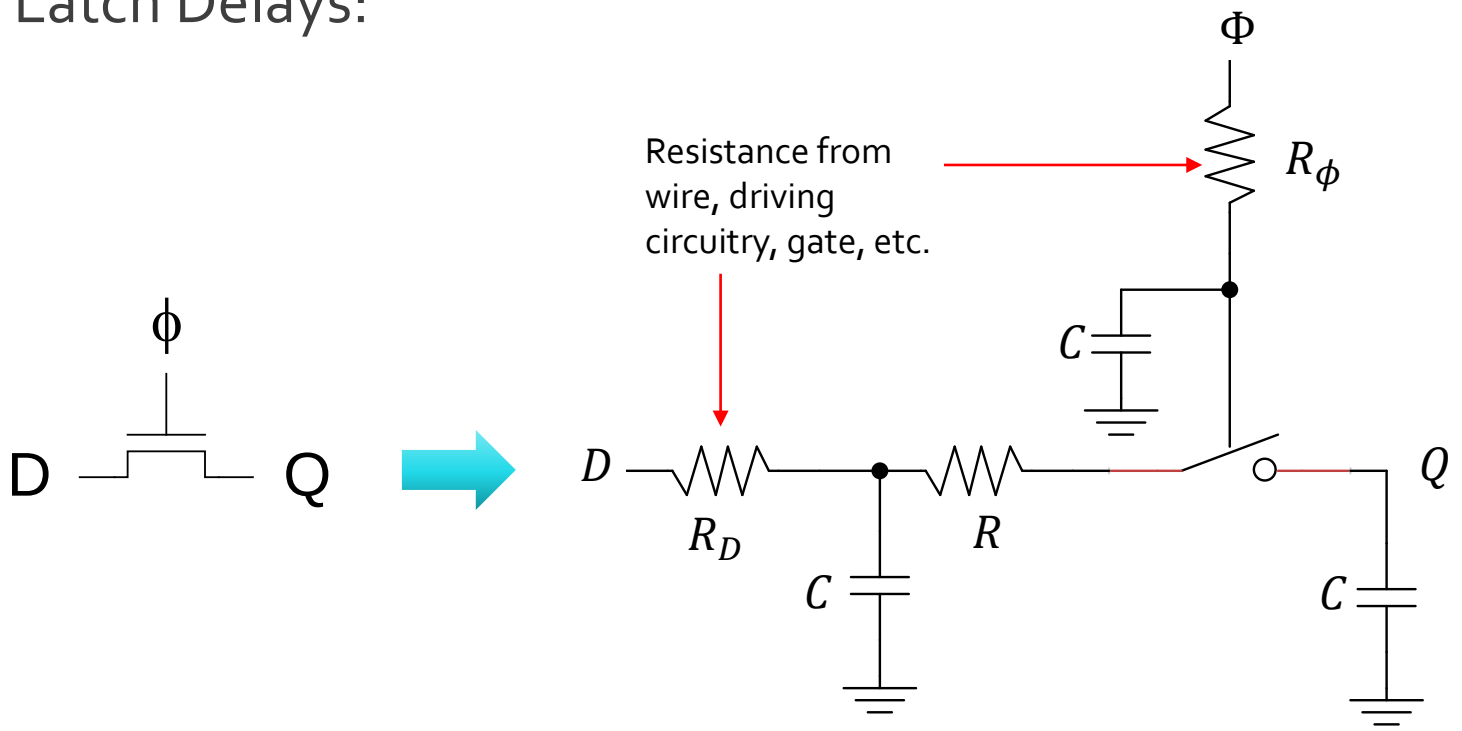
- Pass Transistor Latch
- Pros
  - \+ Tiny
  - \+ Low clock load
- Cons
  - V_t drop
  - Nonrestoring
  - Backdriving
    - Output drives input
  - Output noise sensitivity
  - Dynamic – Output floats when latch is opaque
  - Diffusion input
    - Noise at D could cause transistor to turn on unintentionally
  - Subthreshold leakage – How long does state last?



Used in 1970's

Latch Design

» Pass Transistor Latch Delays:



$$t_{pcq} \approx t_{ccq} \approx R_\phi C + R_D C + (R_D + R)C$$

$$t_{pdq} \approx R_D C + (R_D + R)C$$

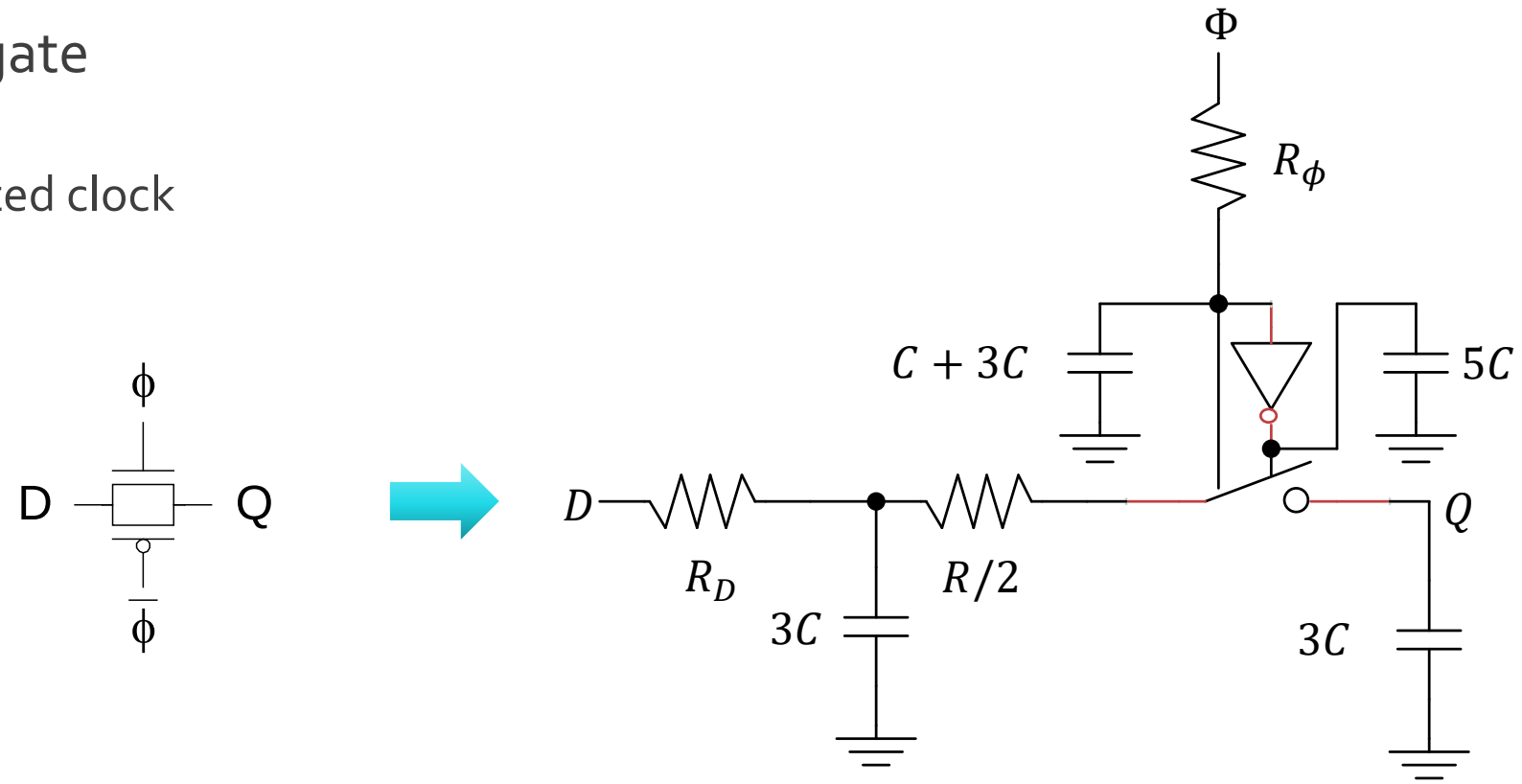
$t_{setup} \approx K t_{pdq}$, e.g. $K = 3$ gets us to 95% of final value

$$t_{hold} \approx R_\phi C - t_{pdq}$$

*Recall that there is a factor of ln2 in these equations.

Latch Design

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



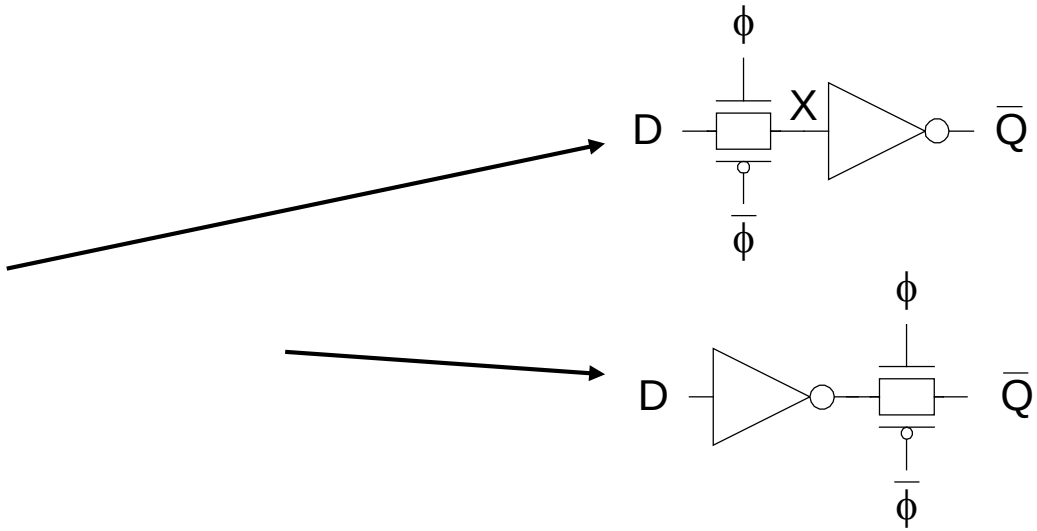
$t_{pcq} \approx t_{ccq}$
$\approx R_\phi 4C + 5RC + R_D 3C + (R_D + R/2)3C$
$t_{pdq} \approx R_D 3C + (R_D + R/2)3C$

$t_{setup} \approx K t_{pdq}$, e.g. $K = 3$ gets us to 95% of final value

$t_{hold} \approx R_\phi 4C + 5RC - t_{pdq}$

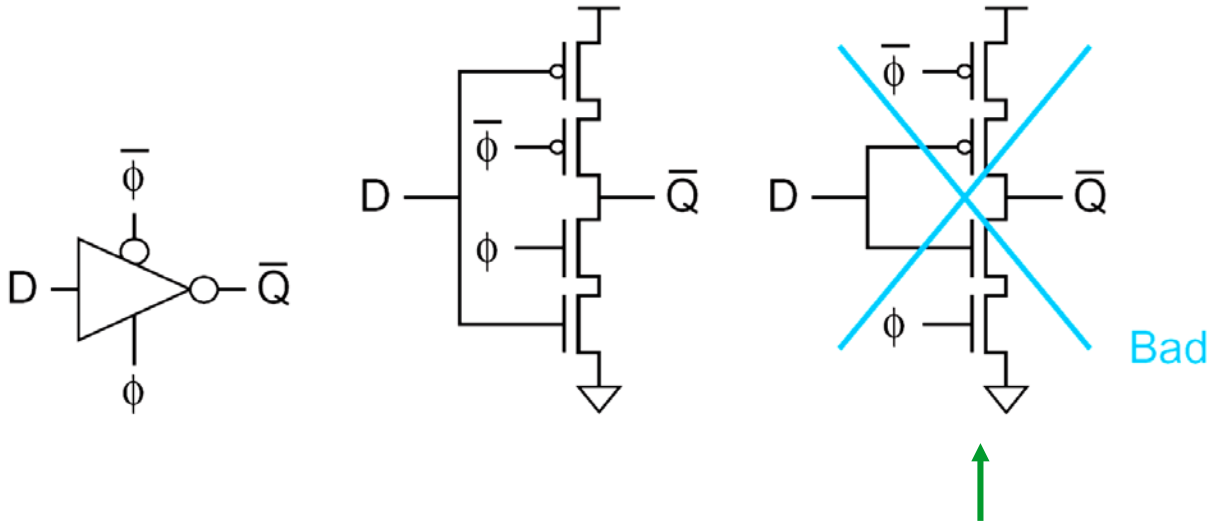
Latch Design

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



Latch Design

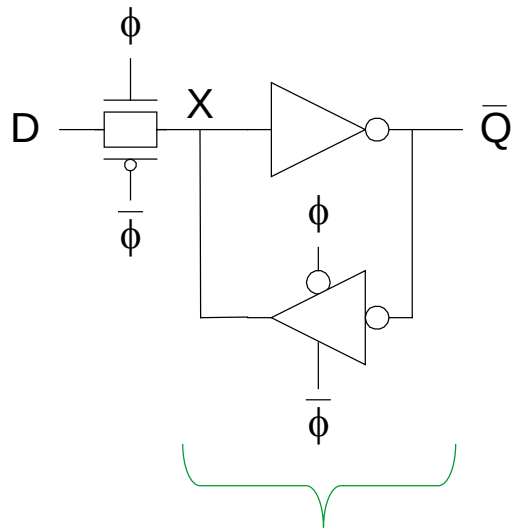
- Clocked CMOS (C^2MOS):
  - \+ No diffusion input
  - \+ Small
  - \- Dynamic



Charge sharing could cause output to change.

Latch Design

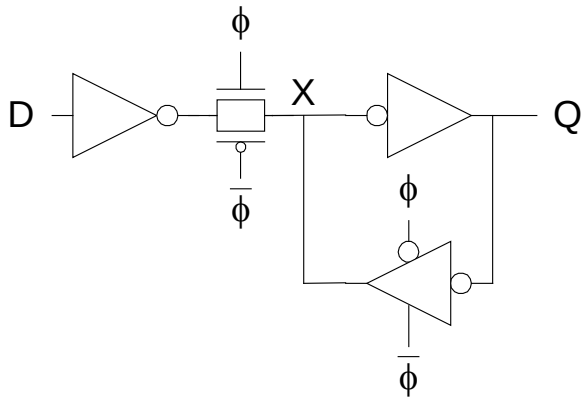
- Tristate feedback
  - \+ Static
  - • Backdriving risk
- Static latches are now essential because of leakage



Does this need to be tri-state? Why?

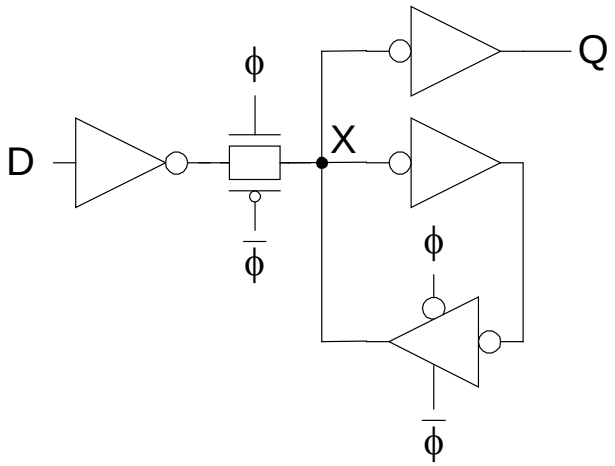
Latch Design

- Buffered input
  - + Fixes diffusion input
  - + Noninverting
  - • Noise at output could change state



Latch Design

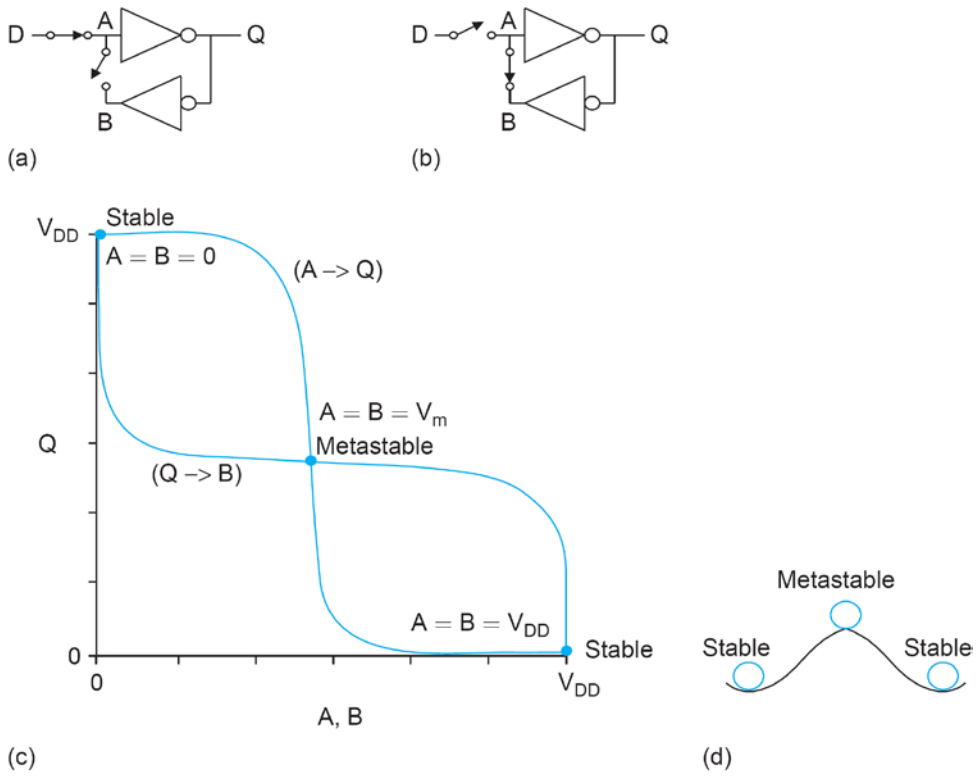
» Buffered output
  + No backdriving



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

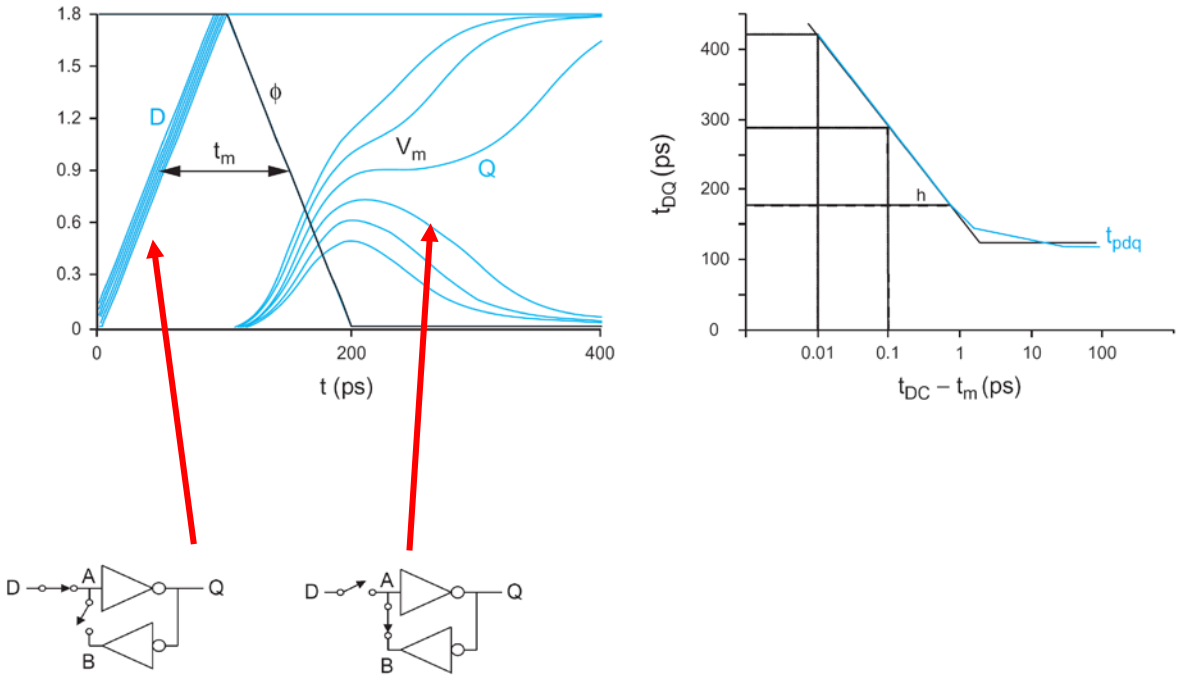
Latch Design

» Metastability
- There are unstable fixed (metastable) points in the latch's voltage space
- Any small disturbance will cause the latch to move from the metastable point to a stable value



Latch Design

» Consider a critical delay t_m between D and the latch input
- If D changes $\approx t_m$ before ϕ, then the latch will be near a metastable state
- The closer D changes to the critical point, the longer the output will take to settle

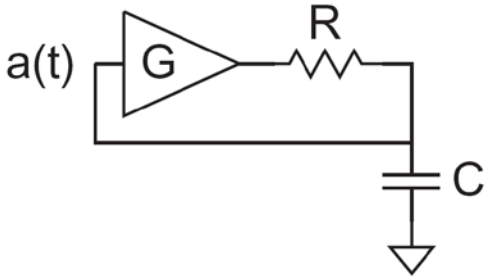


Latch Design

- The latch's behavior near the metastable point V_m can be modeled using a small signal model:
- Initial voltage on the latch input when it becomes opaque:

$$A(0) = V_m + a(0)$$

where a is a small signal offset



- Then,

$$\frac{Ga(t) - a(t)}{R} = C\frac{\mathrm{d}a(t)}{\mathrm{d}t}$$

or,

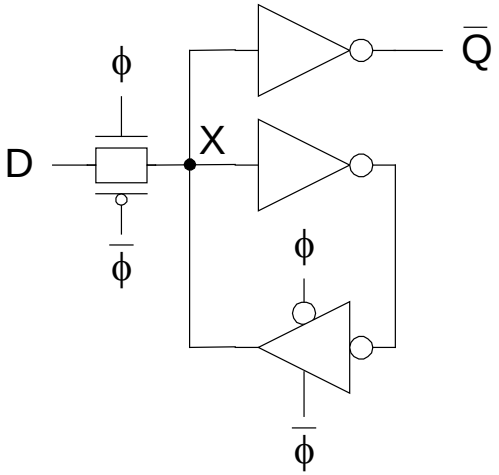
$$a(t) = a(0)\mathrm{e}^{t/\tau_s}, \qquad \tau_s \equiv \frac{RC}{G-1}$$

- Now, define ΔV to be the voltage deviation required at the input node for the latch's state to be considered valid:

$$t_{DQ} = \tau_s[\ln \Delta V - \ln a(0)]$$

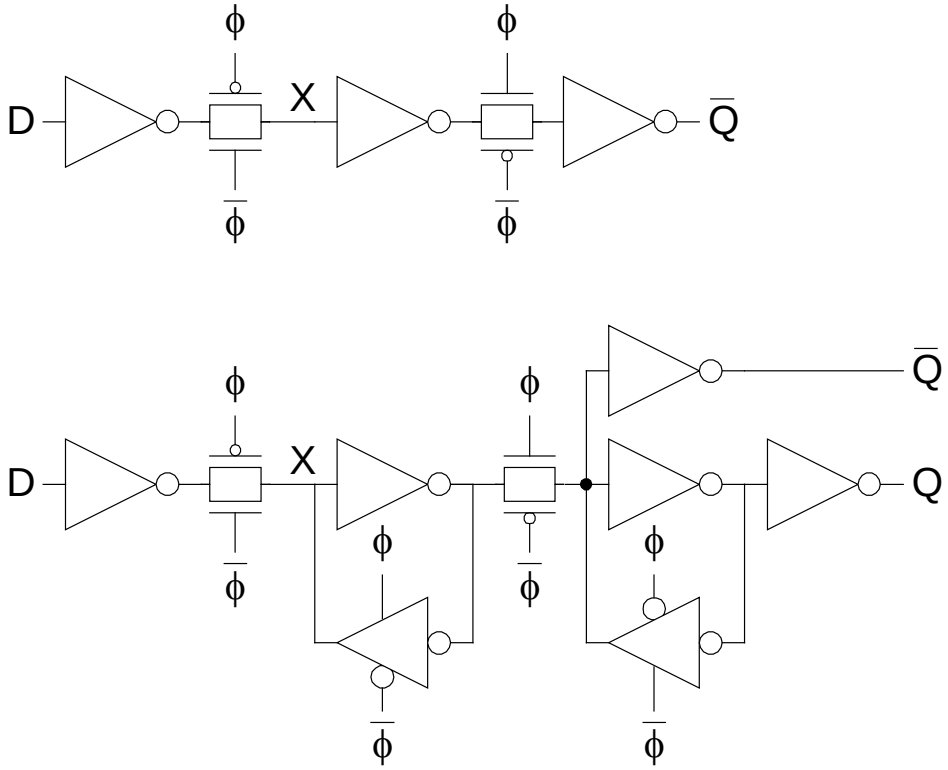
Latch Design

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



Flip-Flop Design

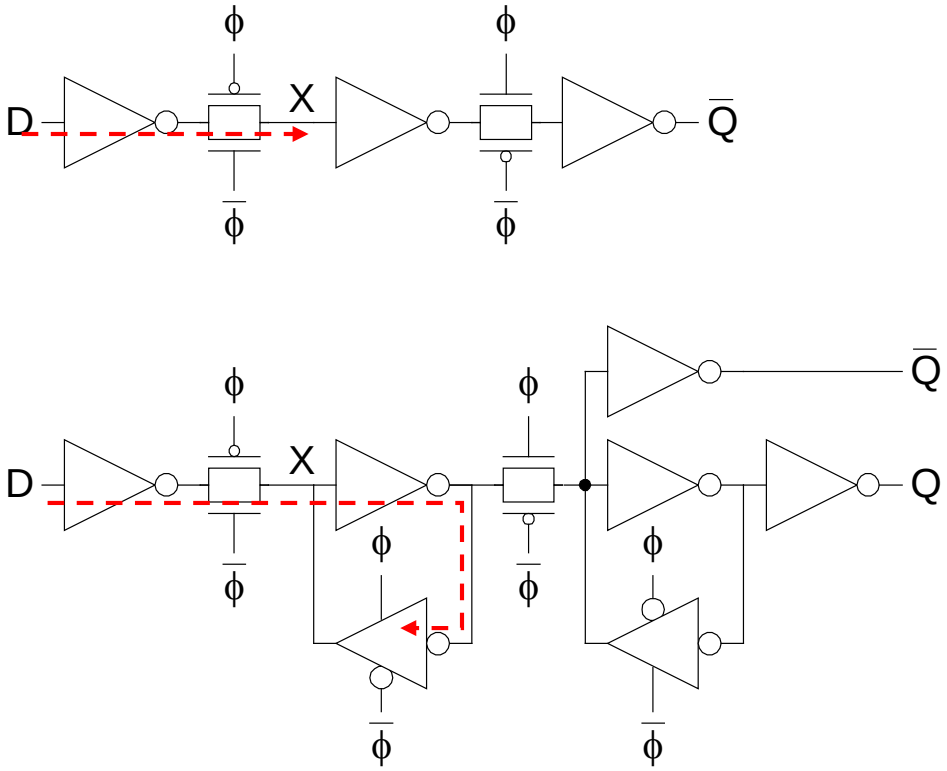
» Flip-flop is built as pair of back-to-back latches



Flip-Flop Design

» Setup Time

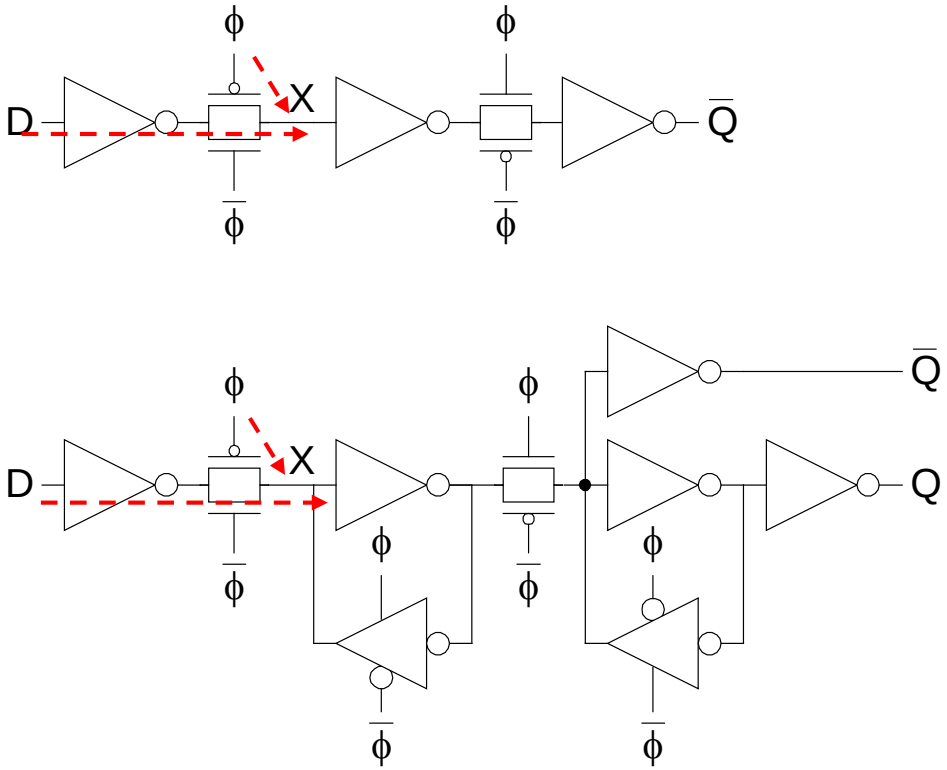
- Setup time is the time it takes for D to be reflected in the master latch:



Flip-Flop Design

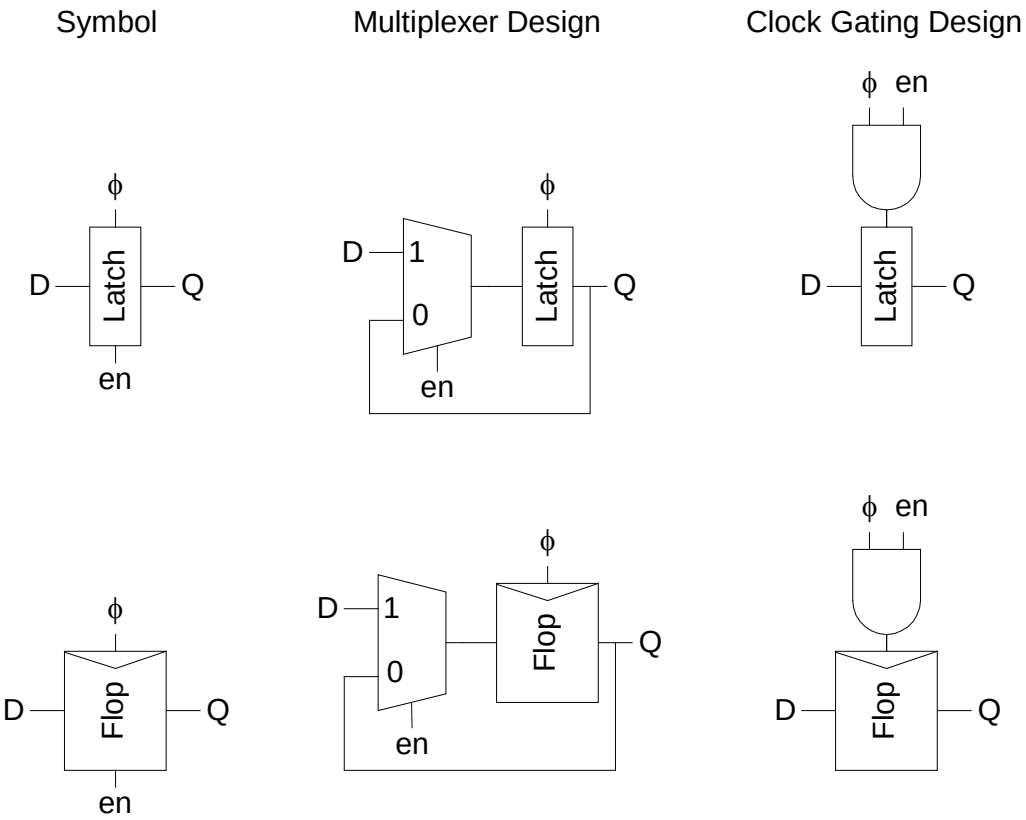
» Hold Time:

- Hold time is the difference between the time it takes D to be reflected at latch input (after the TG) and the time it takes for the TG to turn off after clock edge



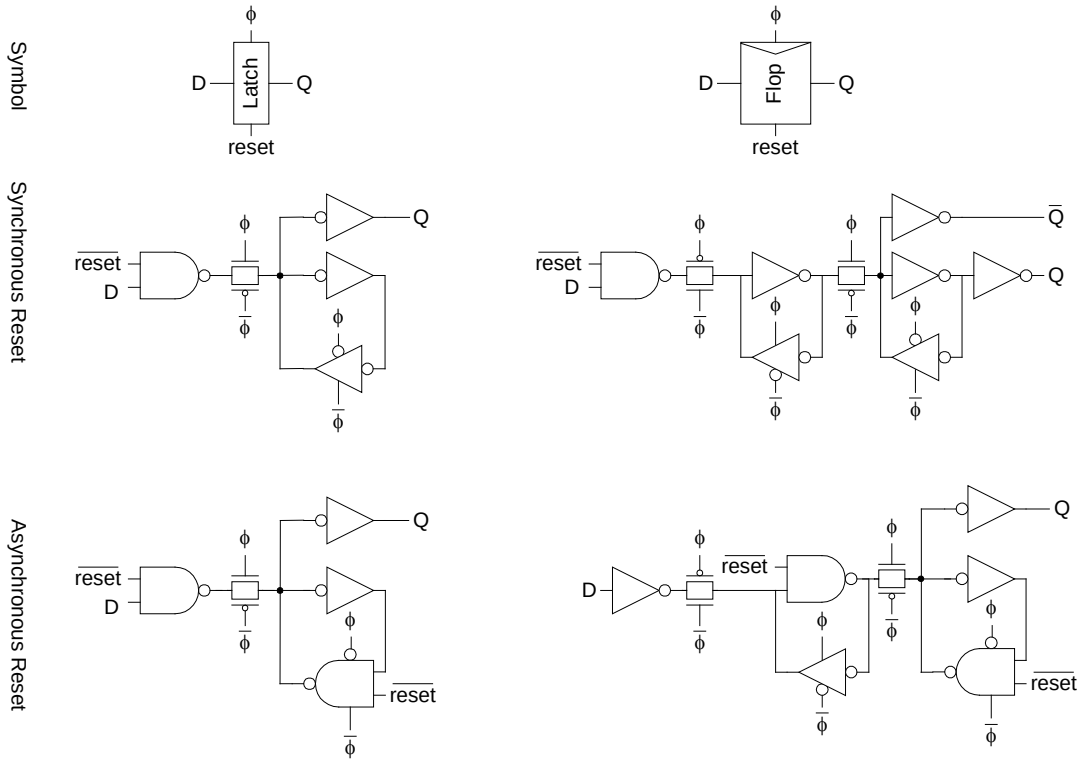
Enable

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



Reset

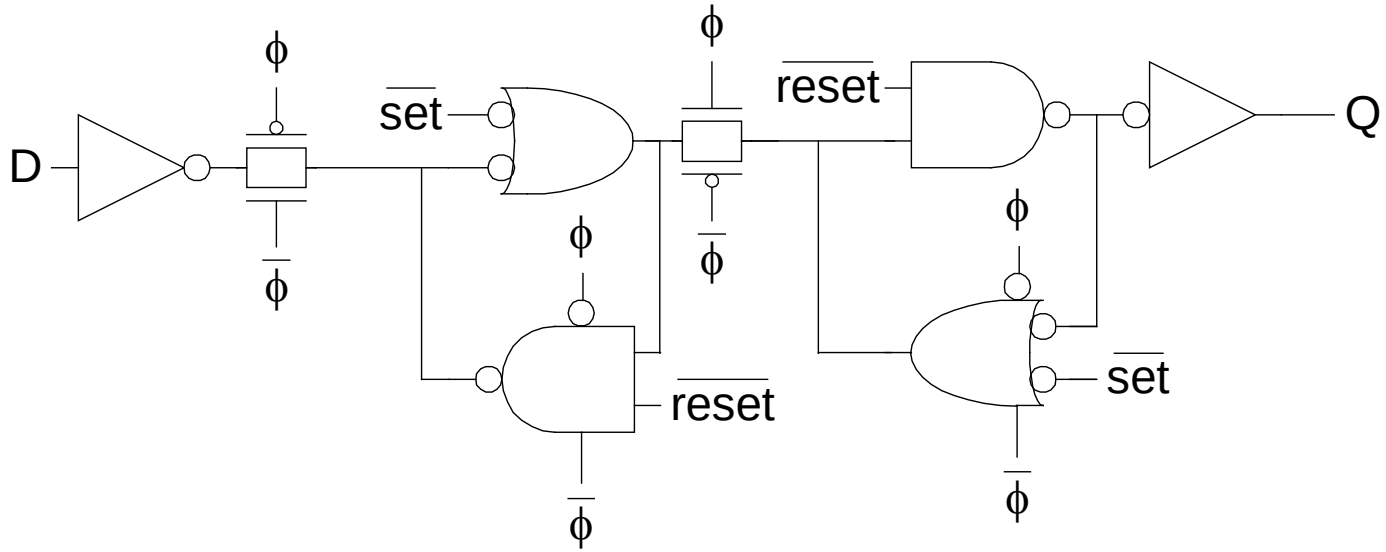
- Force output low when reset asserted
- Synchronous vs. asynchronous



Set / Reset

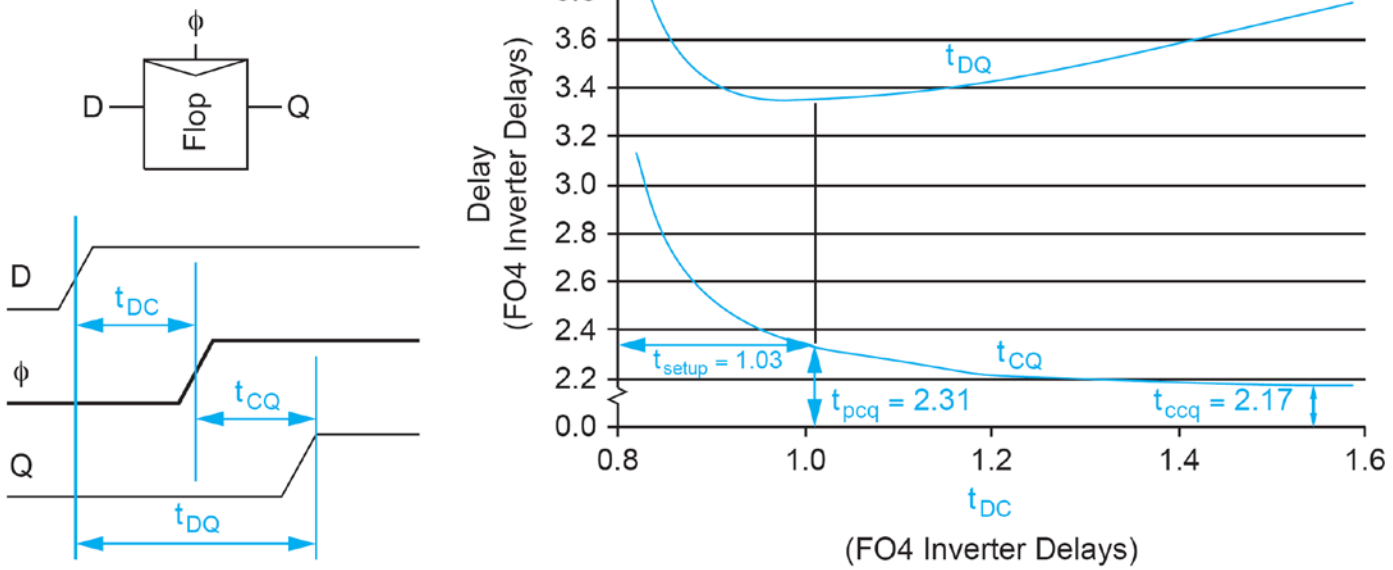
» Set forces output high when enabled

» Flip-flop with asynchronous set and reset



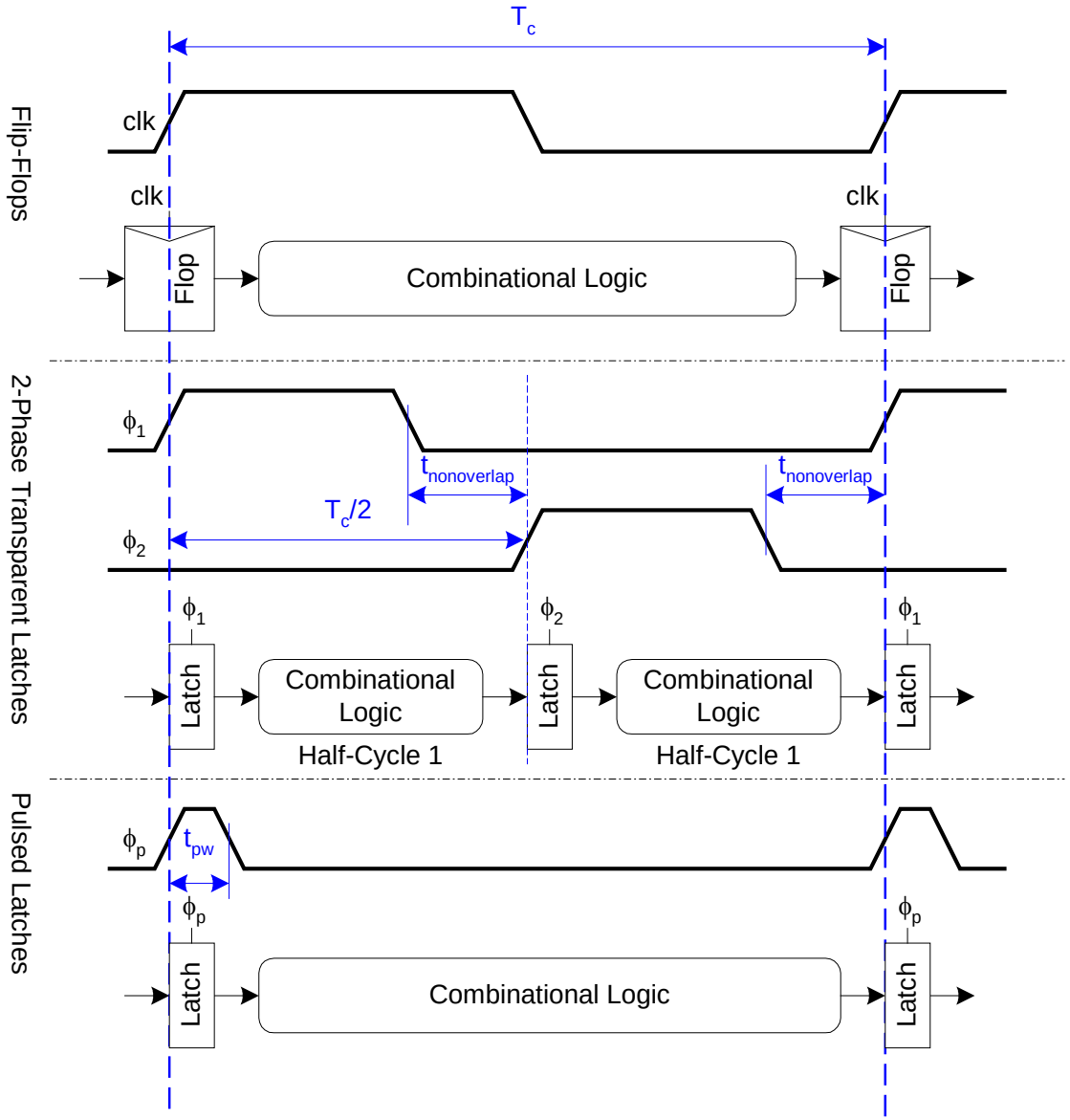
Characterizing Sequencing Elements

- Sequencing elements such as flip flops can be characterized via simulation
- Example:
  - Deriving the setup time as the t_{DC} that yields the minimum t_{DQ}



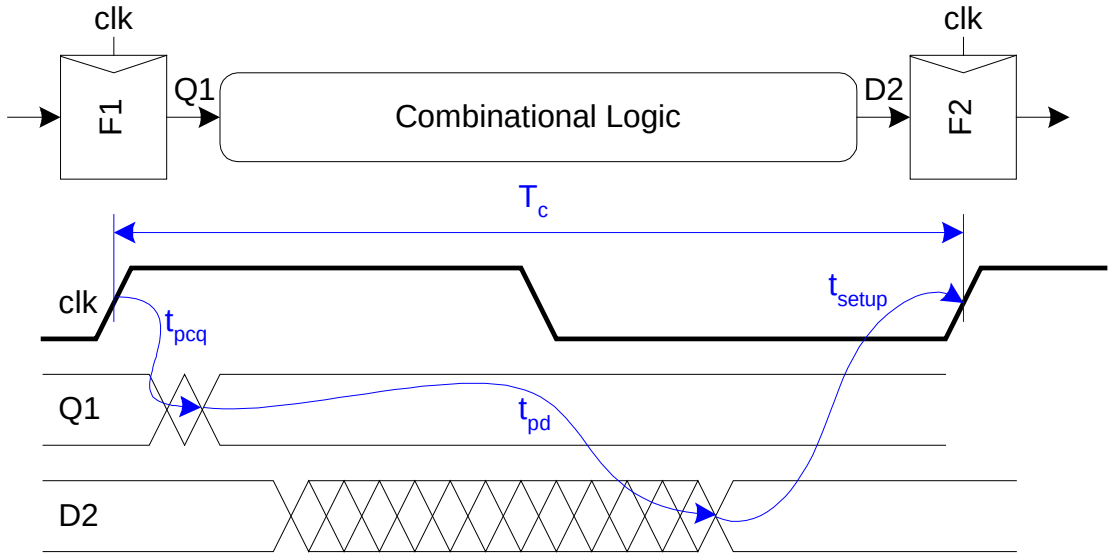
Sequencing Methods

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



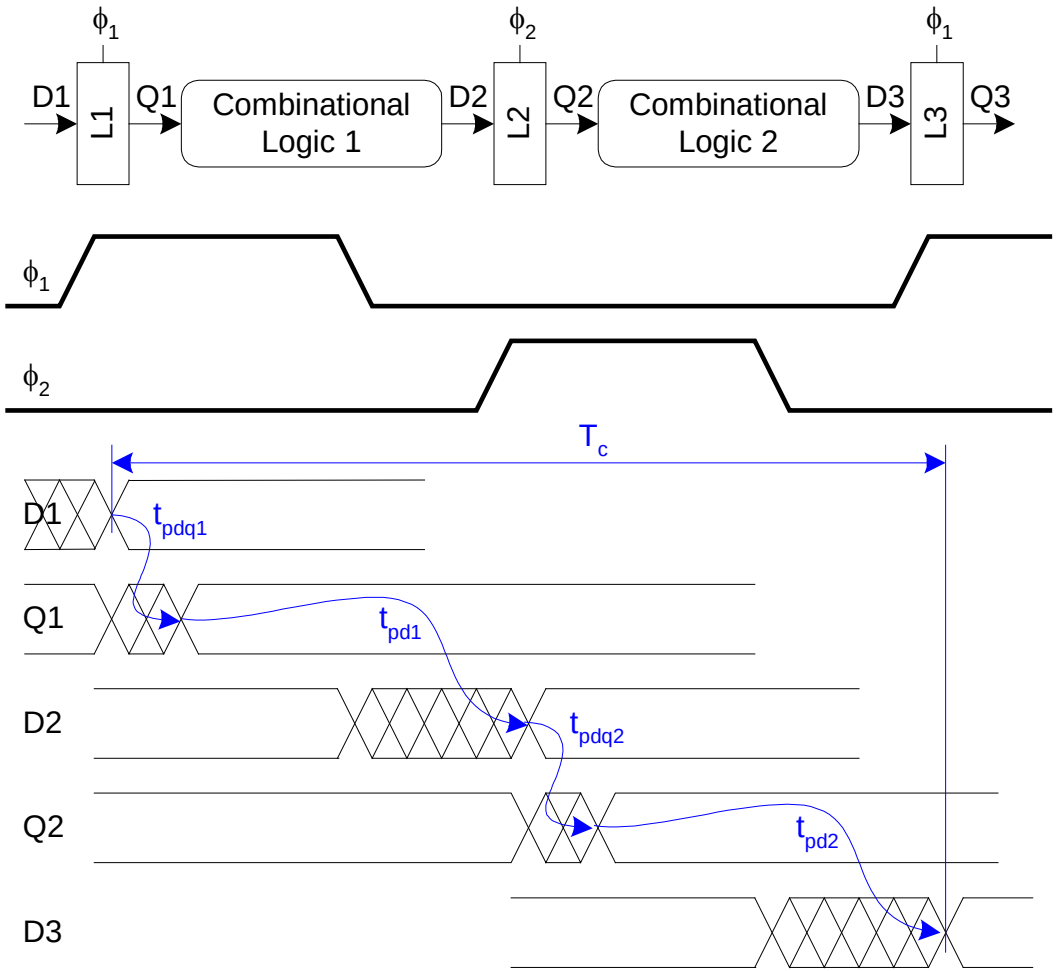
Max-Delay: Flip-Flops

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



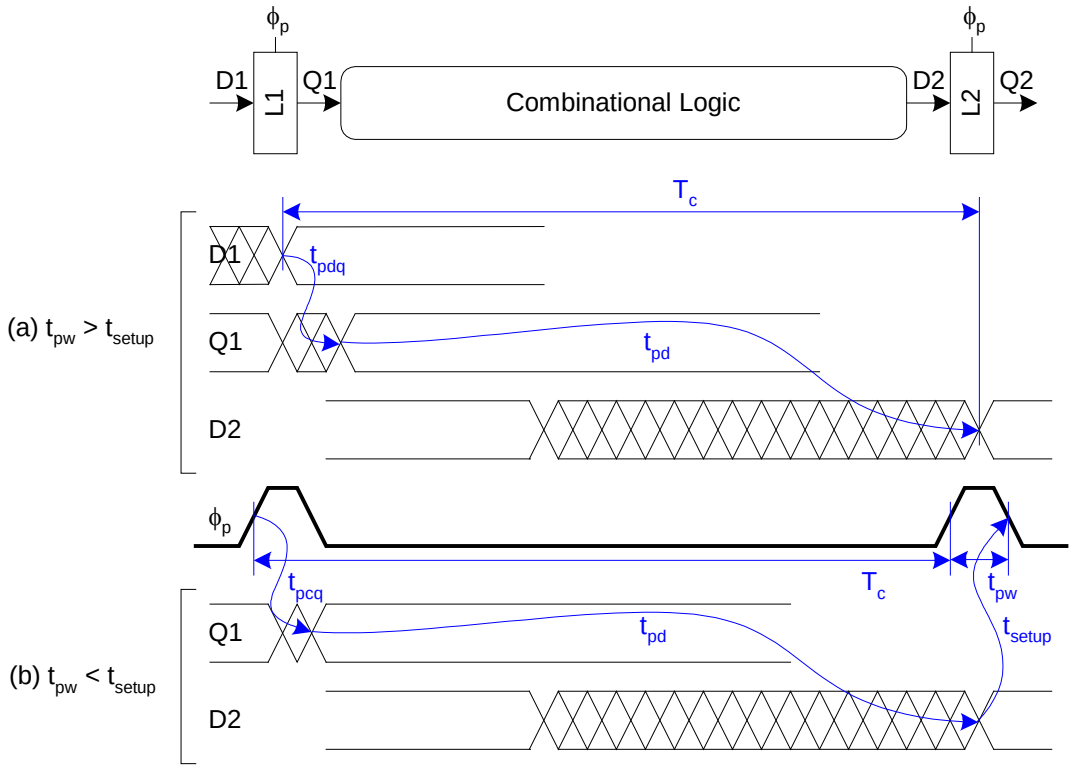
Max Delay: 2-Phase Latches

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



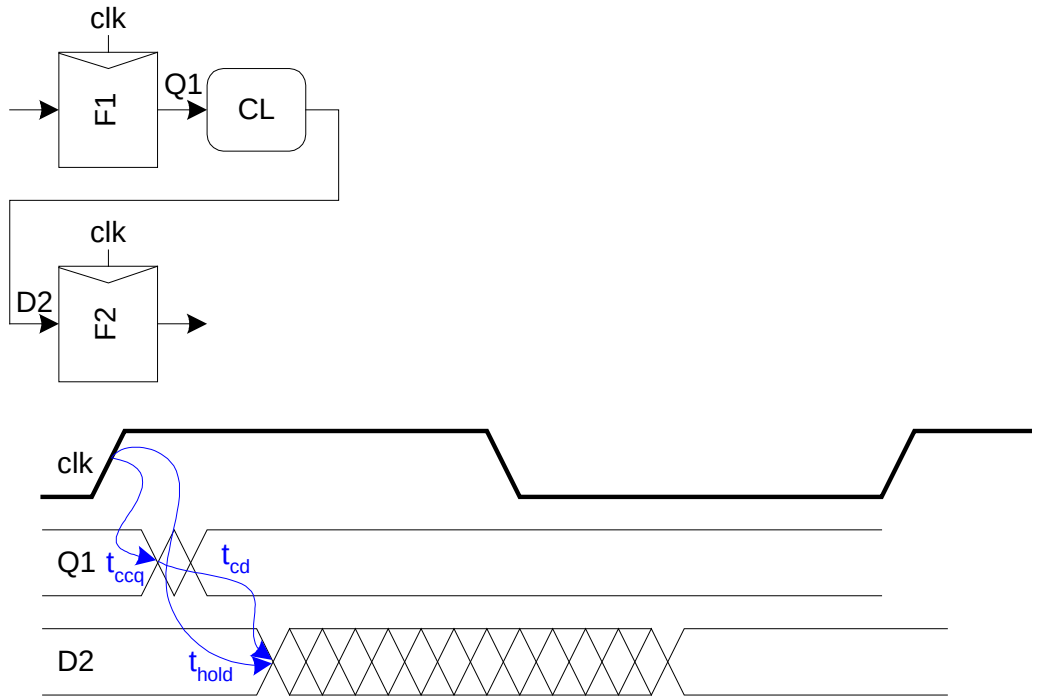
Max Delay: Pulsed Latches

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



Min-Delay: Flip-Flops

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



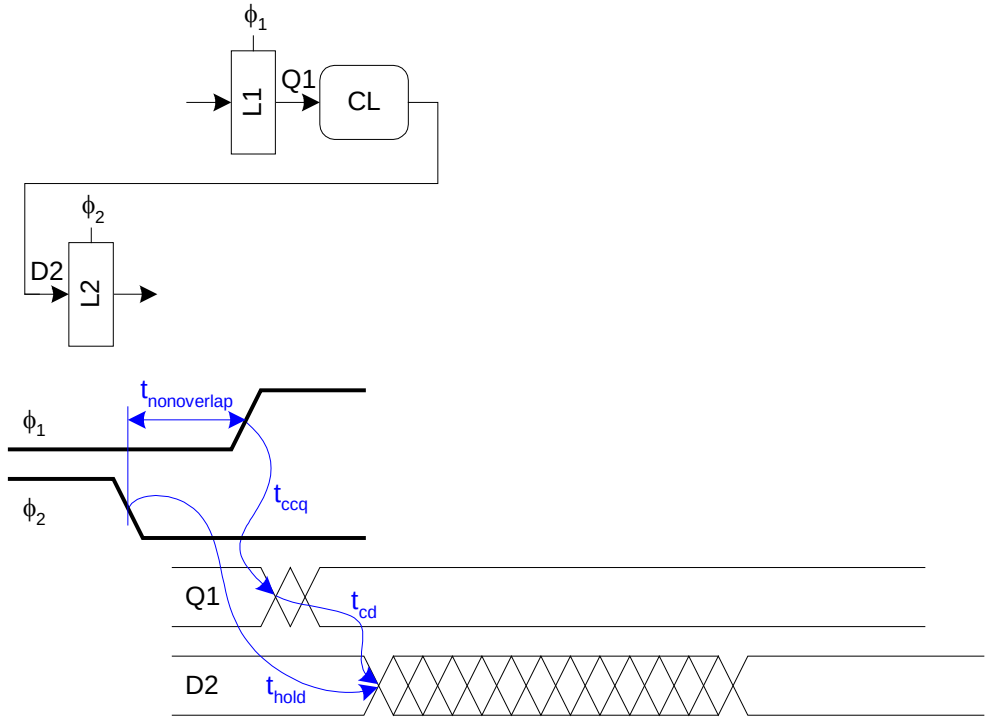
Min-Delay: 2-Phase Latches

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

Hold time reduced by nonoverlap

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

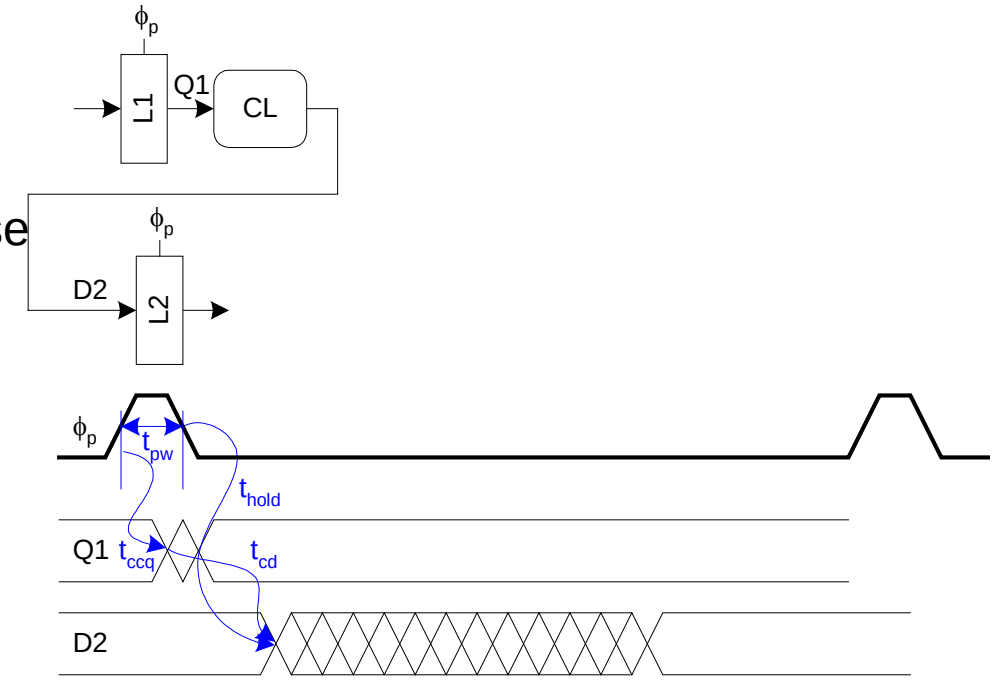
But a flop is made of two latches!



Min-Delay: Pulsed Latches

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

Hold time increased by pulse width

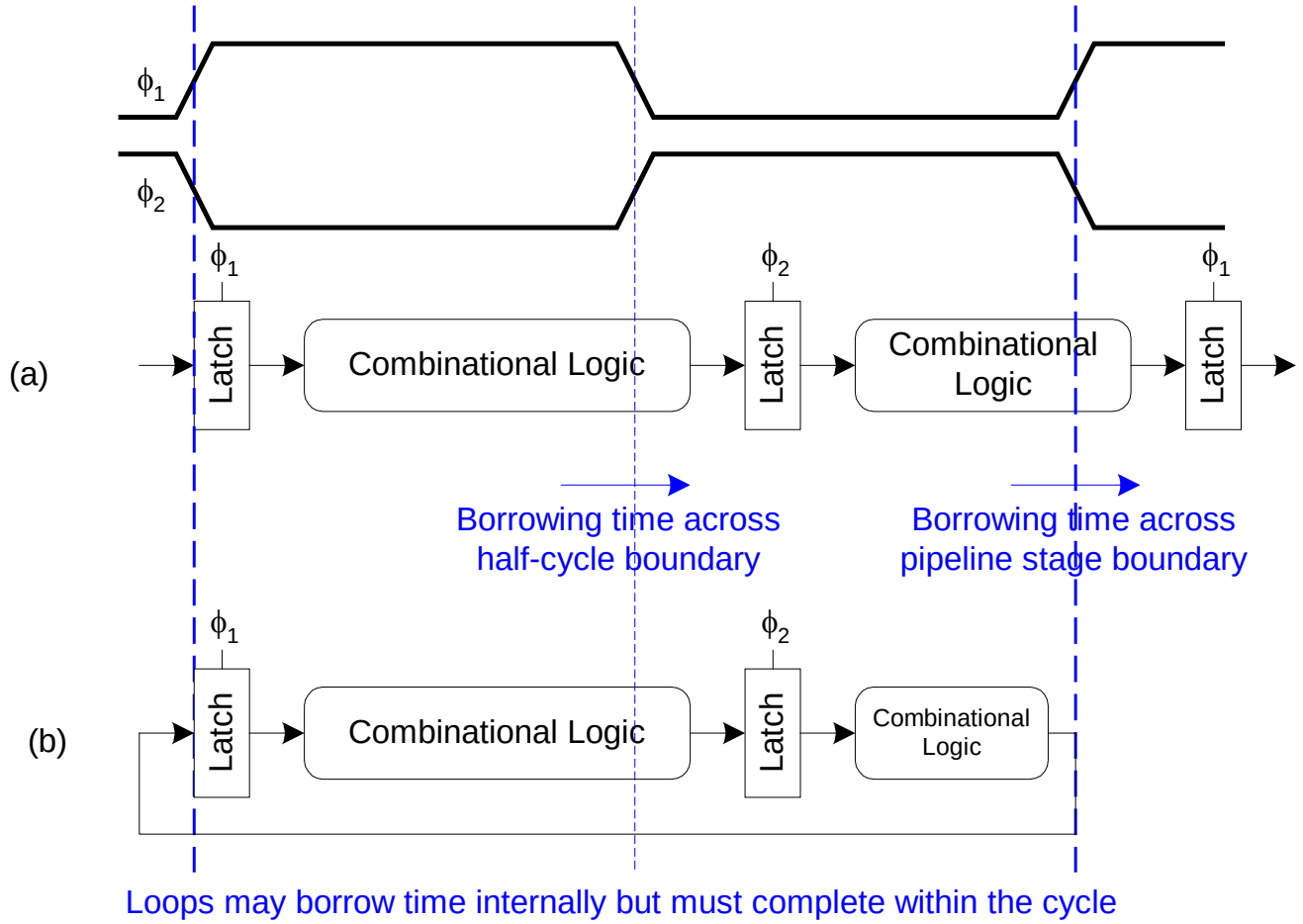


Time Borrowing

- 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



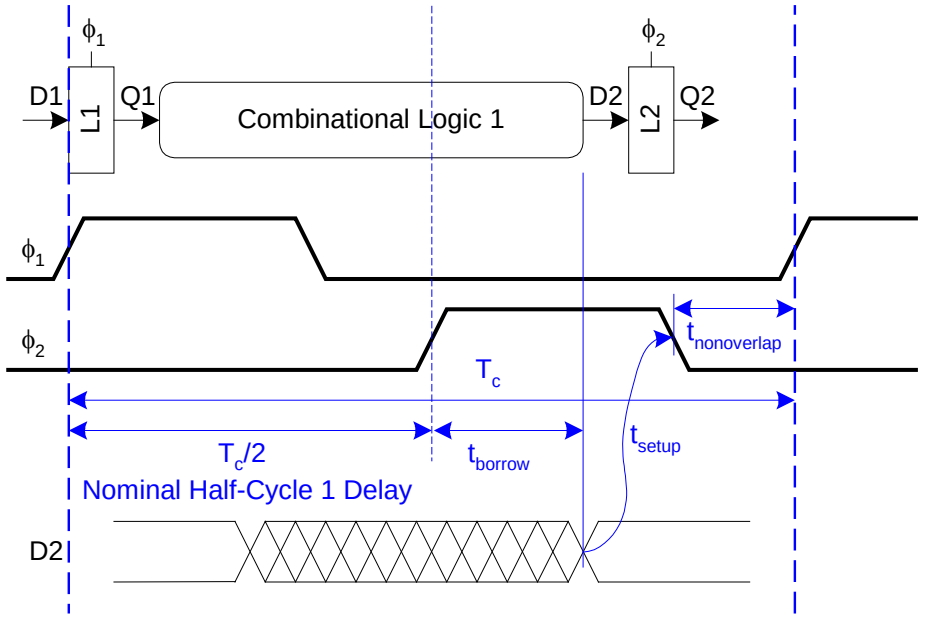
How Much Borrowing?

2-Phase Latches

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

Pulsed Latches

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



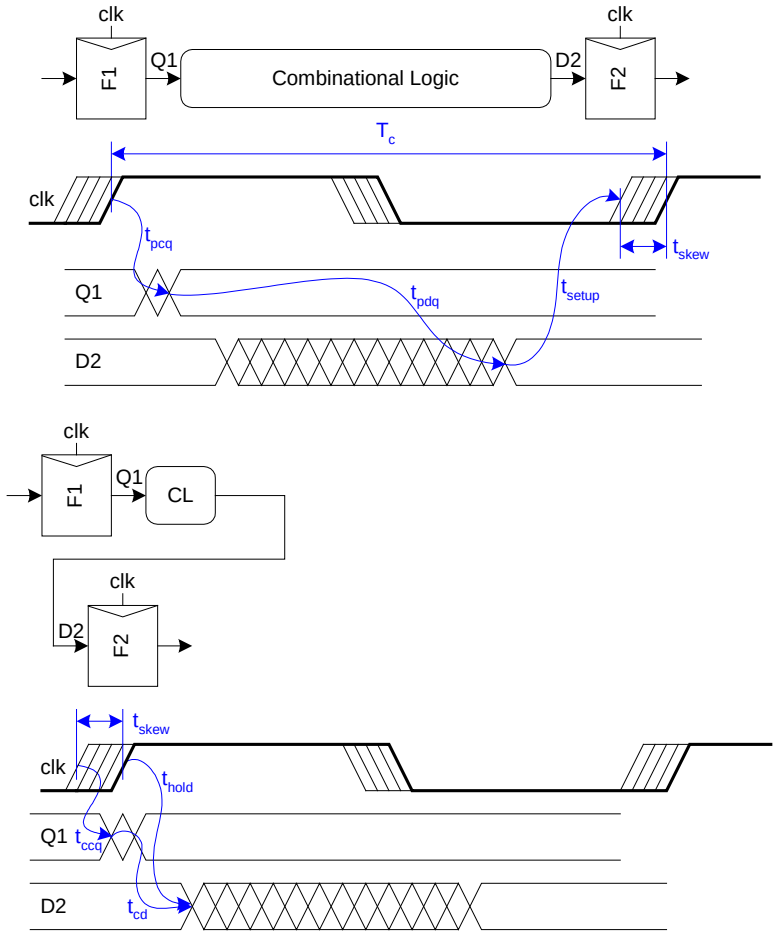
Clock Skew

- 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

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

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



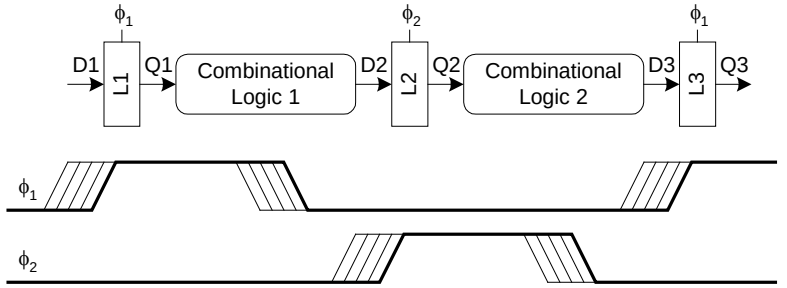
Skew: Latches

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

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

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

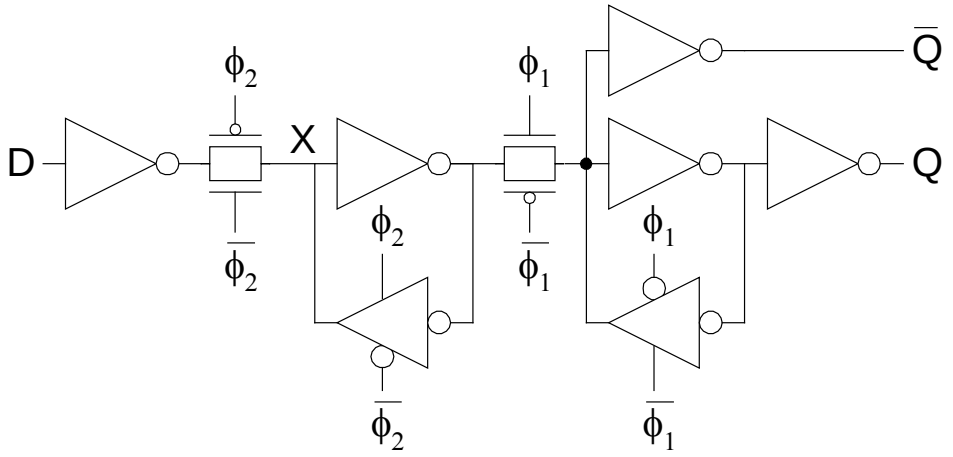
$$t_{\text{borrow}} \leq t_{pw} - \left(t_{\text{setup}} + t_{\text{skew}}\right)$$

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)

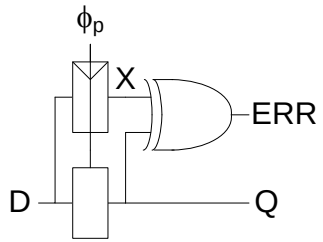
Safe Flip-Flop

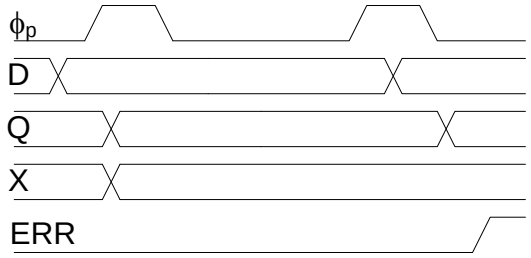
- Past years used flip-flop with nonoverlapping clocks
  - Slow – nonoverlap adds to setup time
  - But no hold times
- In industry, use a better timing analyzer
  - Add buffers to slow signals if hold time is at risk



Adaptive Sequencing

- 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	$\max\left(t_{pdq}, t_{pcq} + t_{setup} - t_{pw} + t_{skew}\right)$	$t_{hold} - t_{ccq} + t_{pw} + t_{skew}$	$t_{pw} - \left(t_{setup} + t_{skew}\right)$