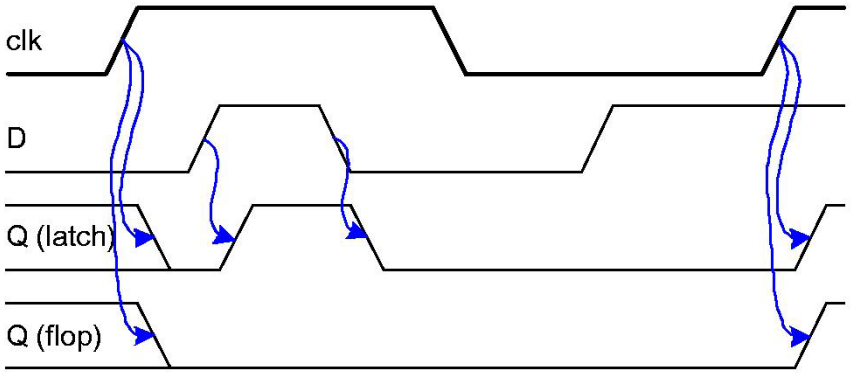
Chapter 7
Sequencing (Sequential Circuit Design)

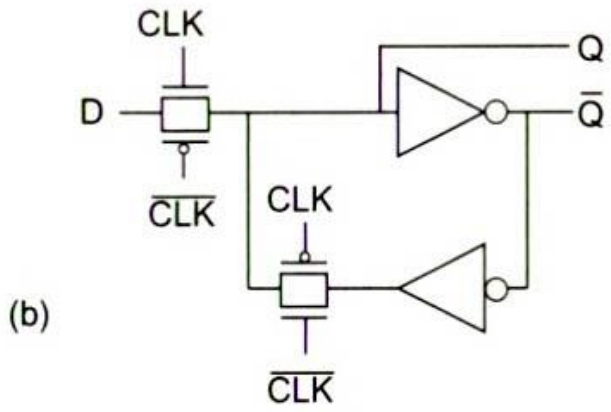
闕志達
台灣大學電機系

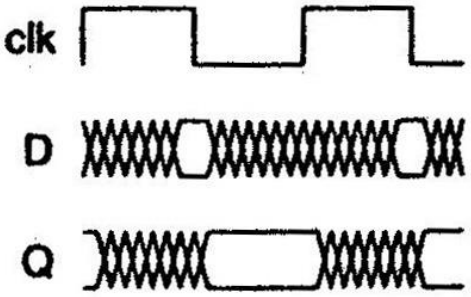
Sequencing Elements

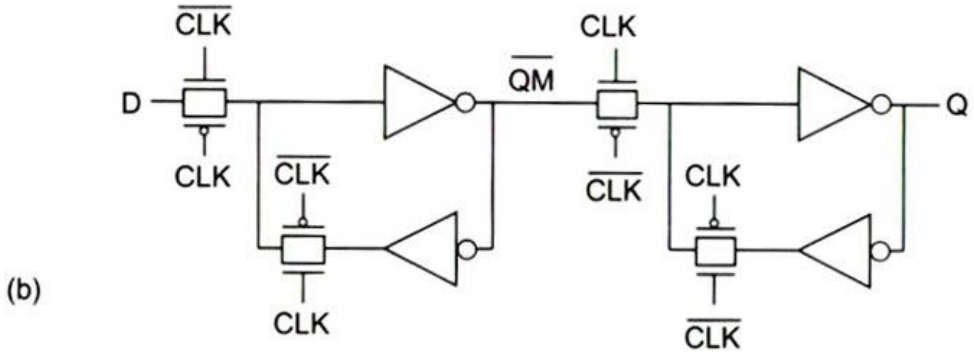
- **Latch**: Level sensitive
  - a.k.a. transparent latch, D latch
- **Flip-flop**: edge triggered
  - a.k.a. master-slave flip-flop, D flip-flop, D register
- Timing diagrams
  - Transparent/opaque

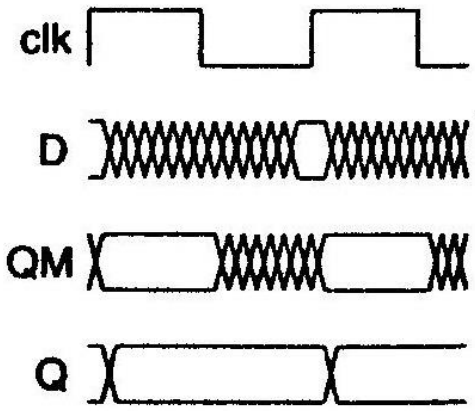


Latch/FF Circuits



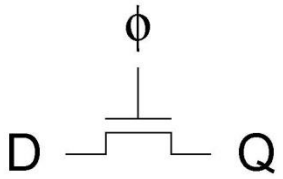




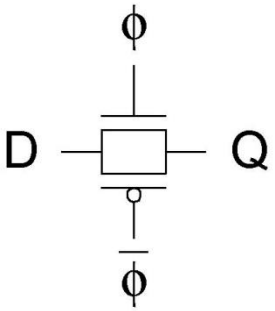


Latch --- Pass Transistor vs Transmission Gate

- Pass transistor
  - Simple, low clock loading
  - Vt drop
  - Non-restoring
  - Dynamic
  - Bilateral (back-driving),
  - Diffusion input (input at S/D),
  - Output noise (exposed output)

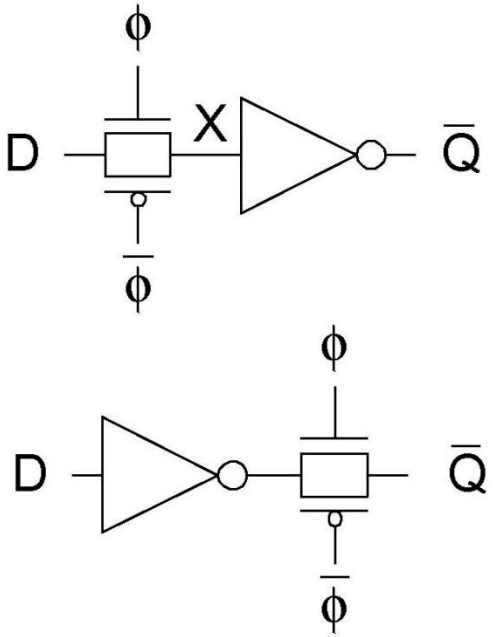


- Transmission gate
  - No Vt drop
  - Require complementary control inputs
  - Larger layout
  - Dynamic
  - Bilateral (back-driving), diffusion input, exposed output



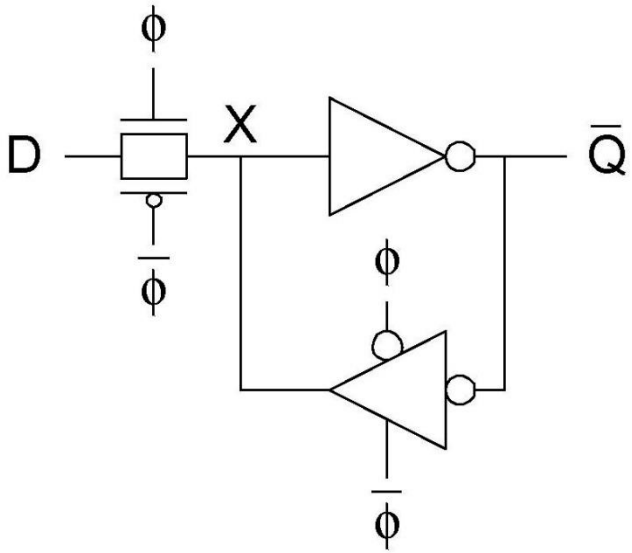
Latch --- Inverting Buffer

- Inverting buffer
  - Restoring
  - No backdriving
  - Fixes either output noise sensitivity or diffusion input
  - Inverted output
  - Still dynamic!



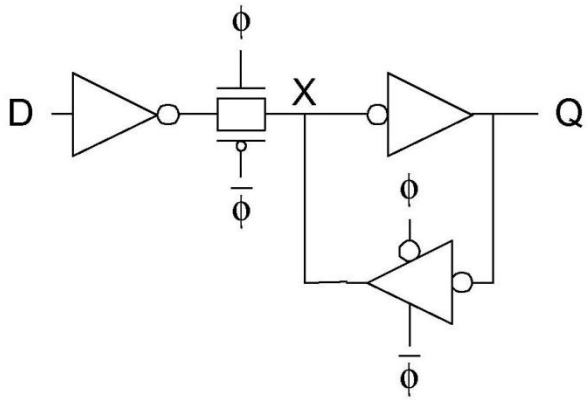
Static Latch

- Static
- Tri-state inverter feedback connection (Fig. 9.18)
- Back-driving risk
- Diffusion input

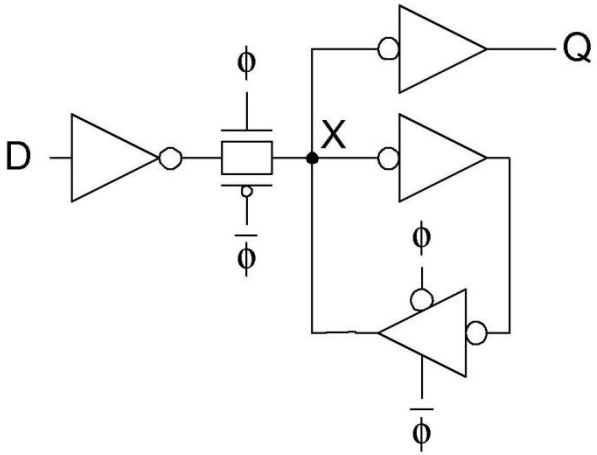


More Static Latches

- Buffer-input static
  - No diffusion input
  - Non-inverting
  - Back-driving

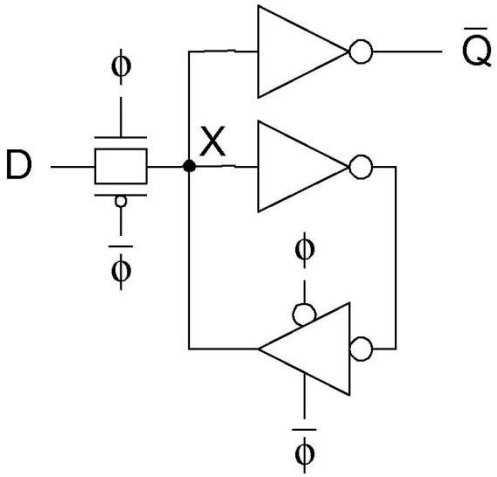


- Buffered-input/output static latch
  - No backdriving
  - Widely used in standard cells
  - Very robust (most important)
  - Rather large
  - Rather slow (1.5 – 2 FO4 delays)
  - High clock loading



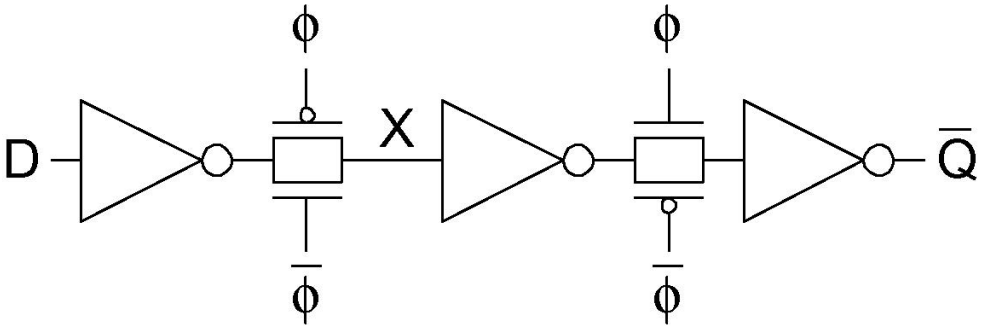
Datapath Latch

- Input noise can be controlled
- Buffered output, unbuffered input.
- Smaller, faster.



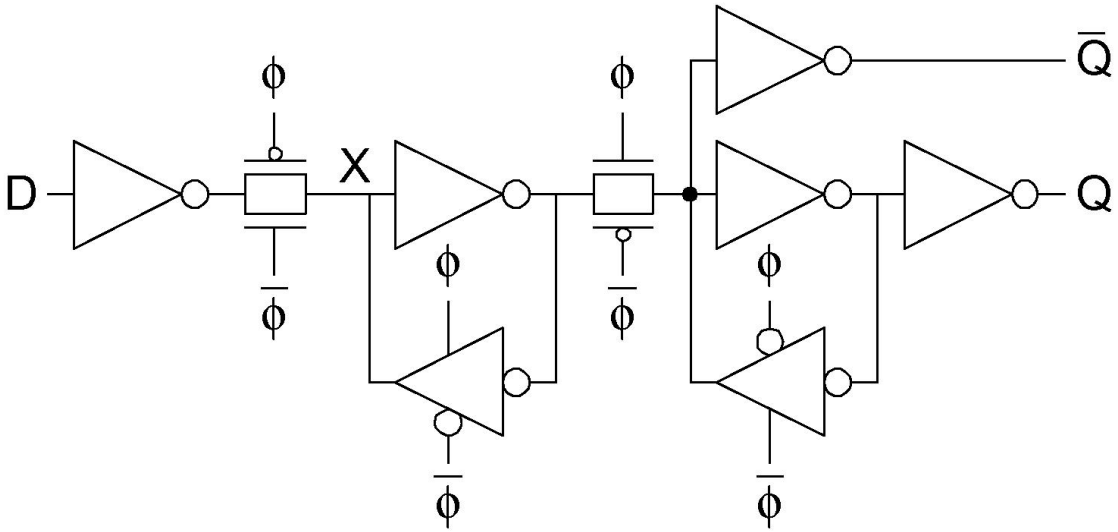
Flip-Flop Design

- Flip-flop is built as pair of back-to-back latches with complementary clocking signals.
- Dynamic FF
  - Much smaller and simpler
  - Still have clock skew problem that can cause race/hazard.
  - Mainly used in shift register.
  - The clock should go in opposite direction with the signal.



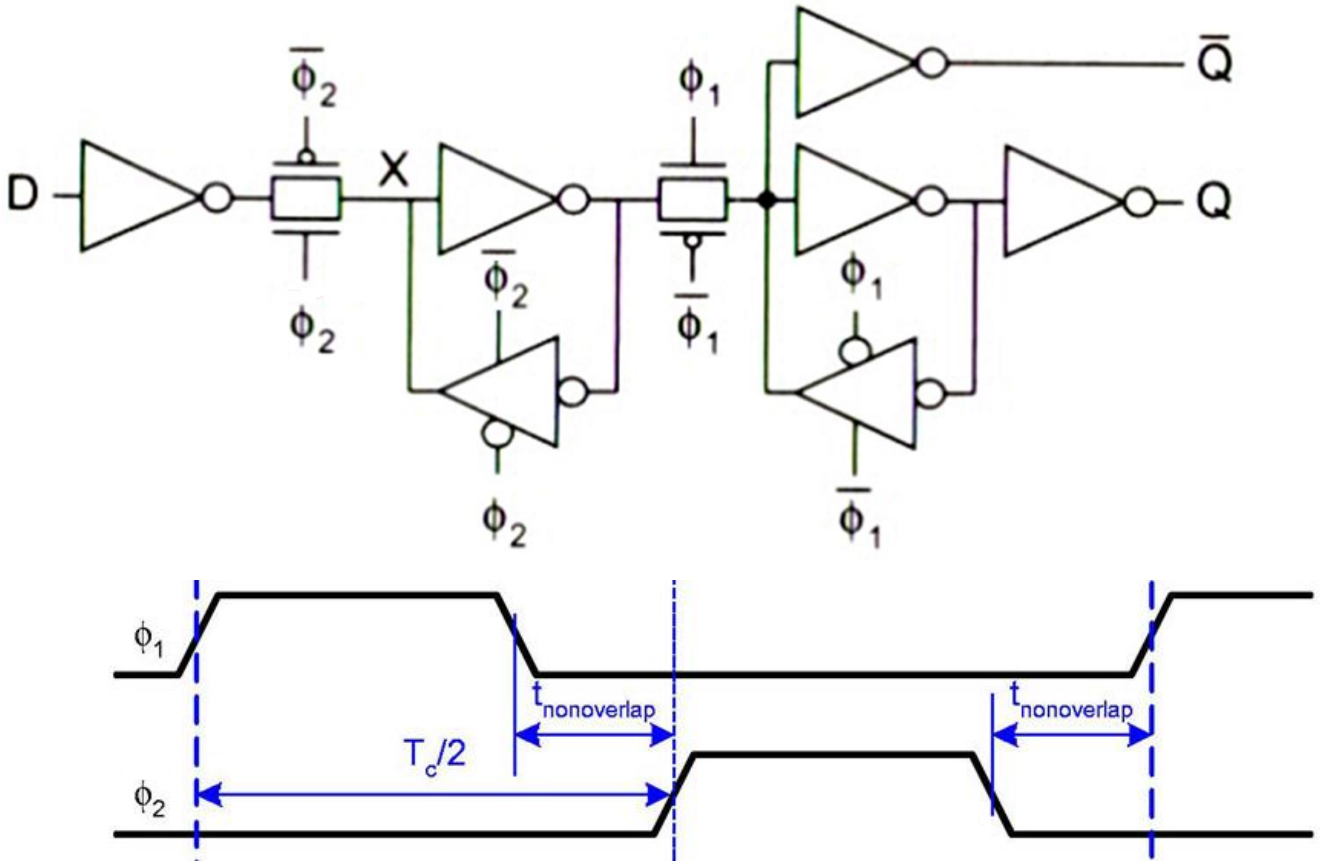
Static FF

- Static FF
  - Used in standard cell library
  - Very robust
  - Usually the complementary clock is generated locally by an inverter. Without sharp edge, there may be charge sharing.
  - May used extra buffer for clock signals to sharpen edges.



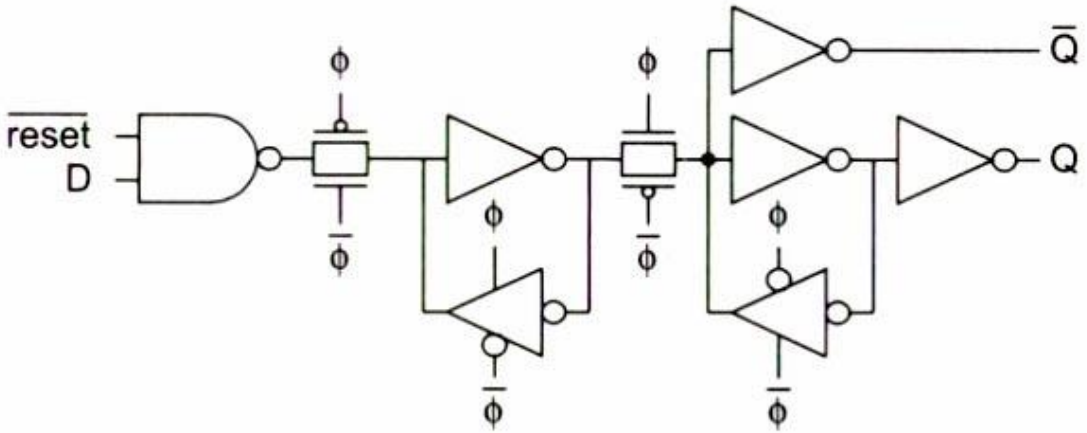
Two-Phase Clocking

- Avoid clock skew problem
- More routing/power for clock signal distribution.

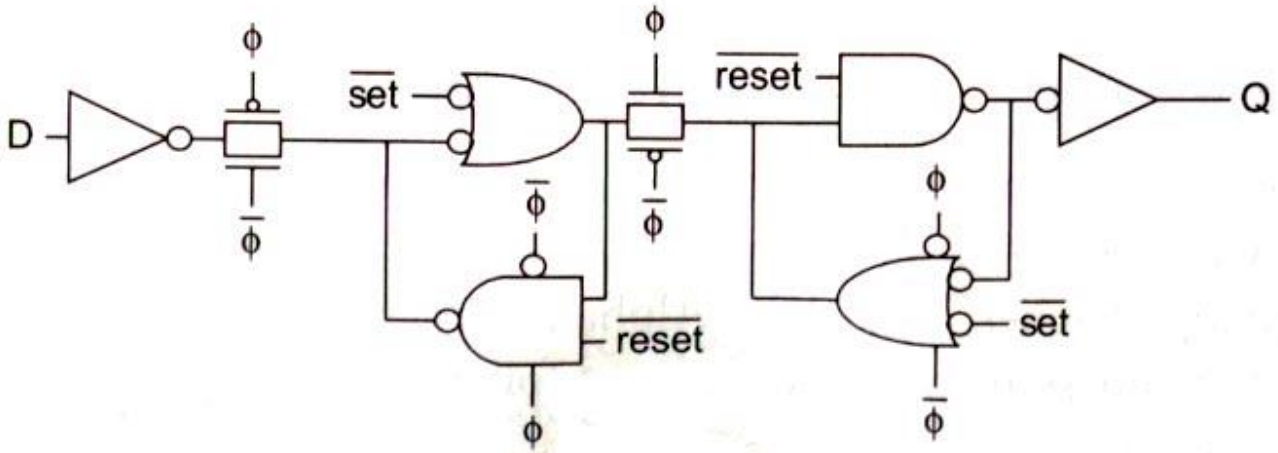


FF with Reset

- Synchronous reset FF

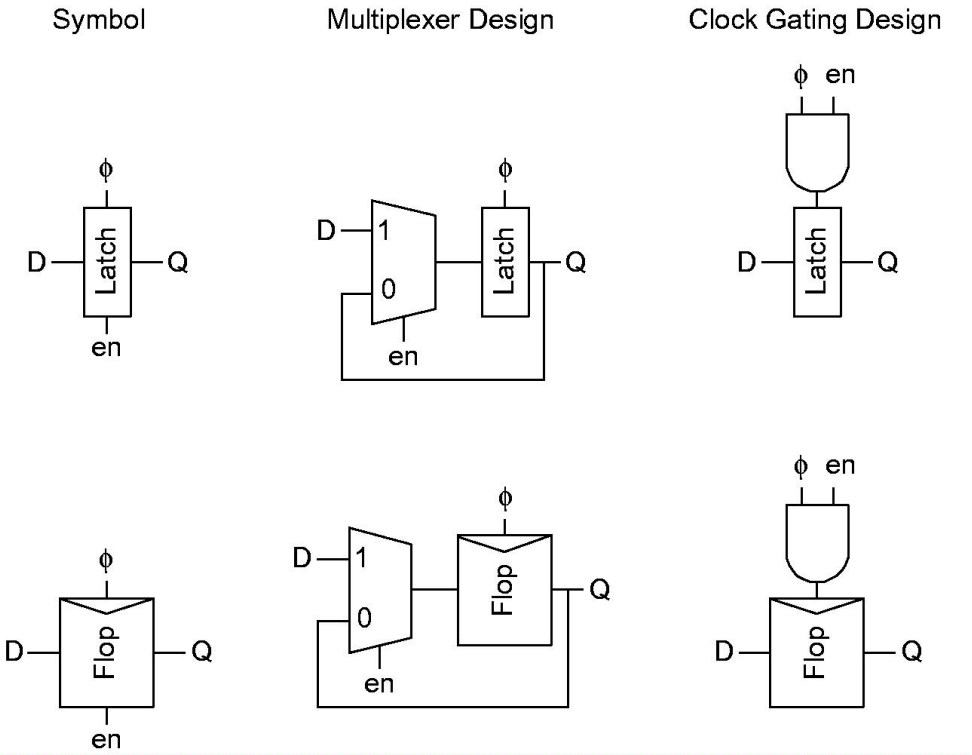


- Asynchronous set/reset FF



Enable in FF

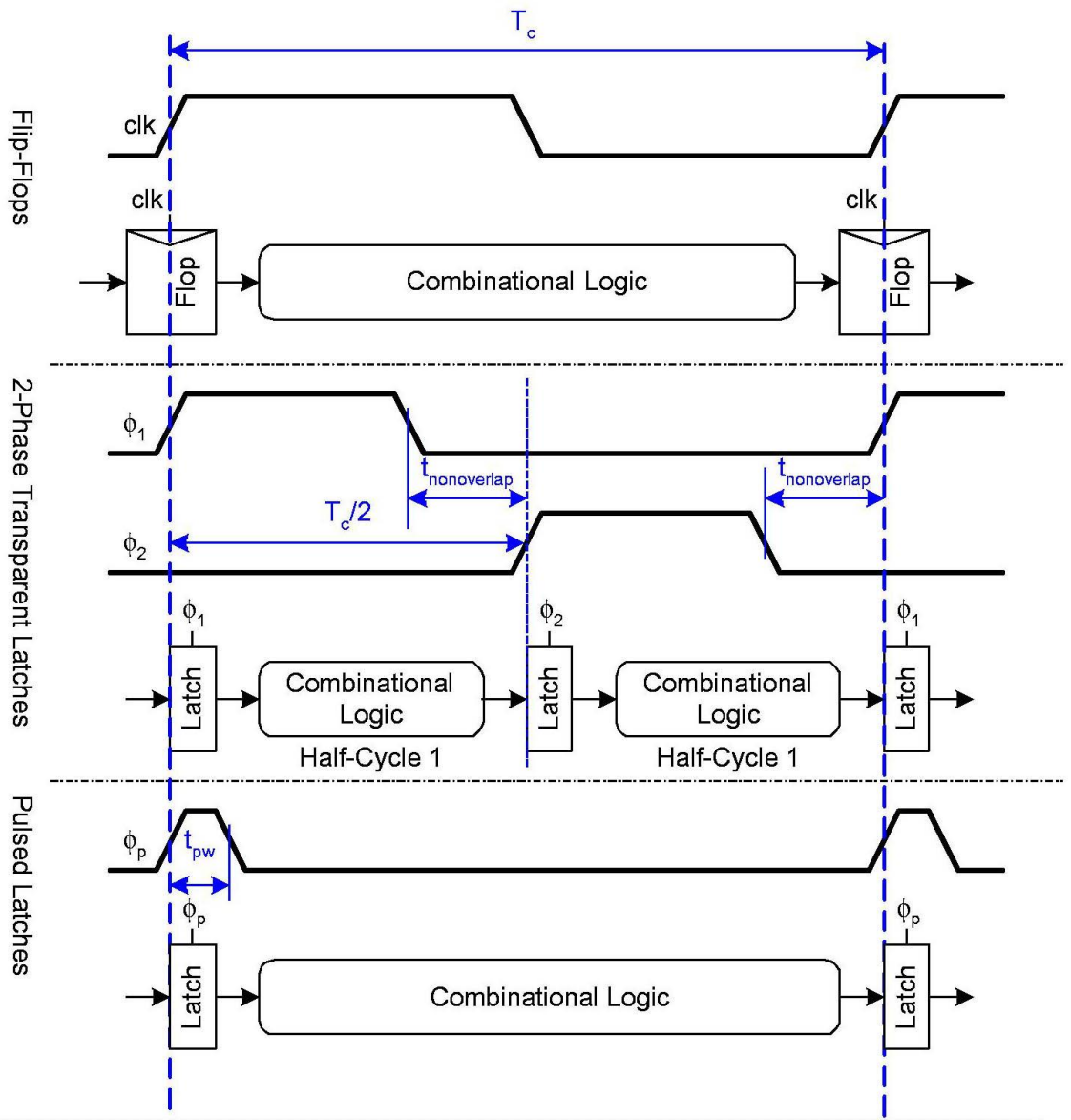
- Enable: ignore clock when en = 0
  - Mux: increase latch D-Q delay
  - Clock Gating: must make sure gated clock in CLEAN, 'en' must be stable when clk=1.



Sequencing Methods

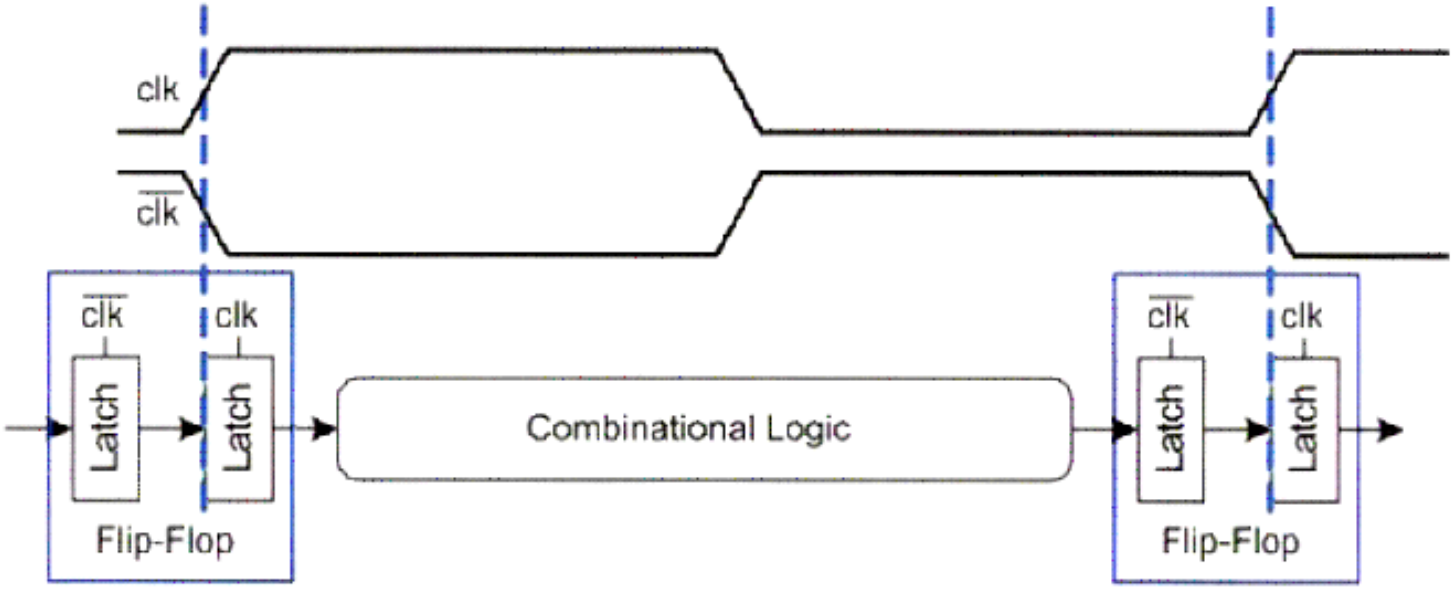
- Flip-flops
- 2-Phase Latches
- Pulsed Latches

- Similar to gates along a long corridor or canal.



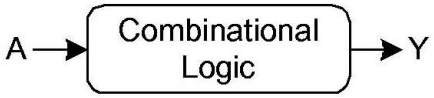
FF-based Sequencing

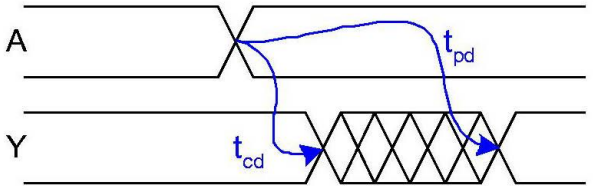
- Note that an FF is a pair of back-to-back latches (master-slave).
- At any given time in the FF-based and latch-based methods, one latch is transparent and one latch is opaque.

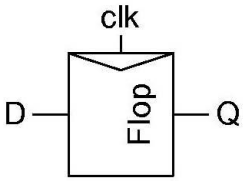


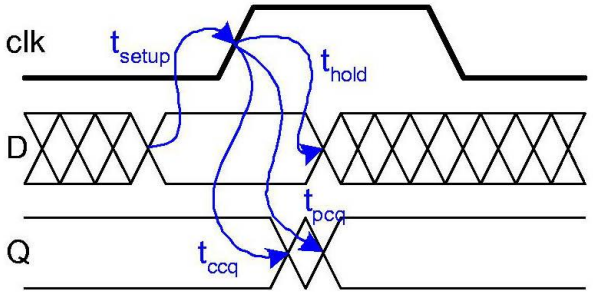
Timing Diagrams

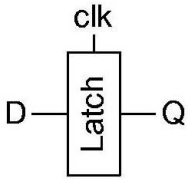
t_{pd}	Logic Prop. Delay
t_{cd}	Logic Cont. Delay
t_{pcq}	Latch/Flop Clk->Q Prop. Delay
t_{ccq}	Latch/Flop Clk->Q Cont. Delay
t_{pdq}	Latch D->Q Prop. Delay
t_{cdq}	Latch D->Q Cont. Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

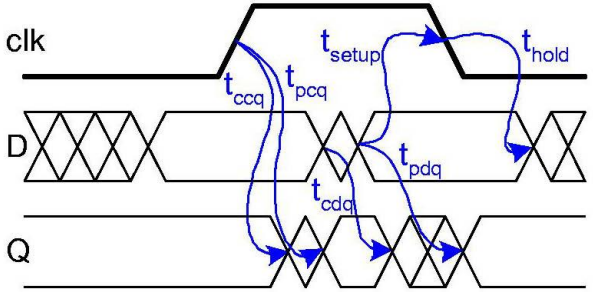








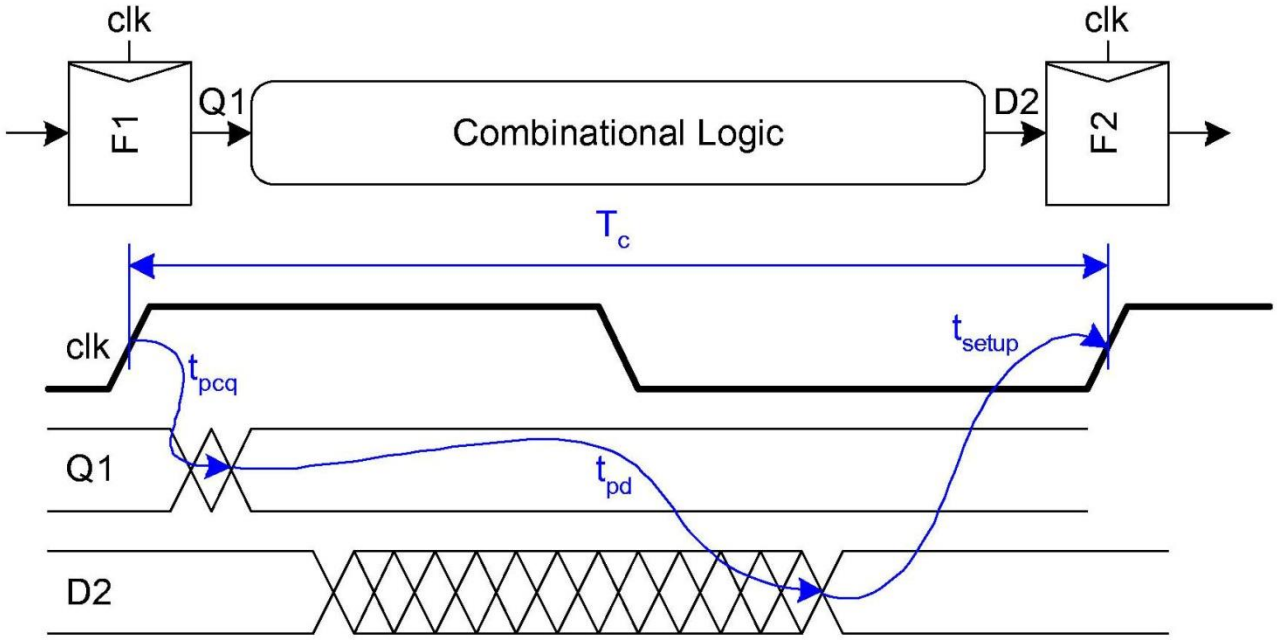




Max-Delay Constraints: FF

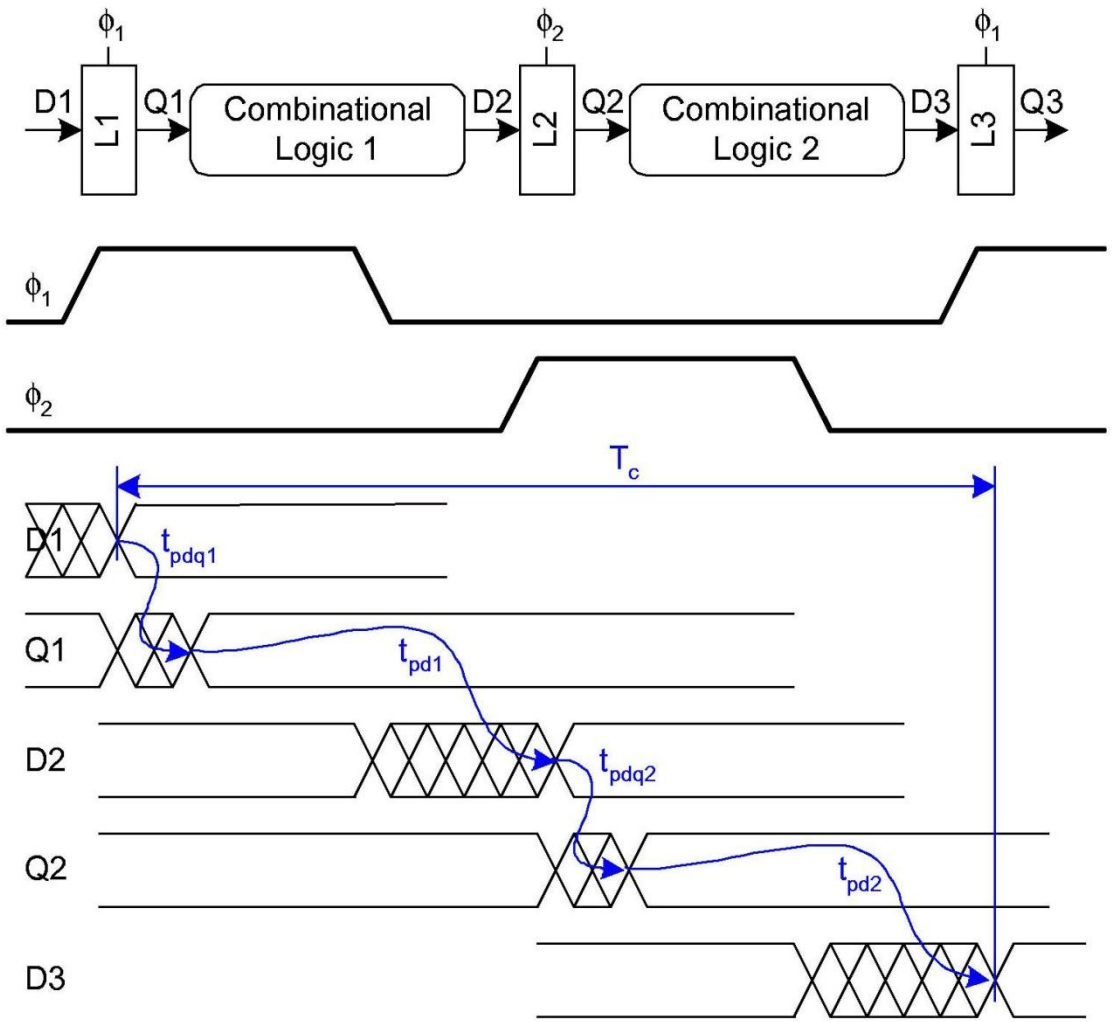
- Sequencing overhead decreases the available propagation delay for the combinational logic.

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



Max-Delay: 2-Phase Latch

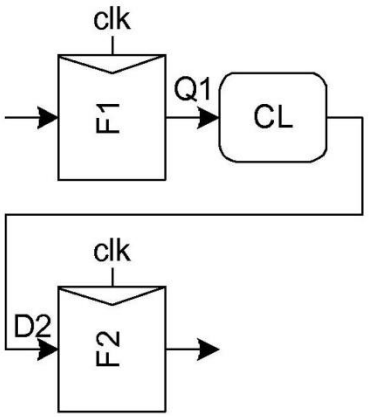
$$t_{pd} = t_{pd1} + t_{pd2} \le T_c - \underbrace{\left(2t_{pdq}\right)}_{\text{sequencing overhead}}$$

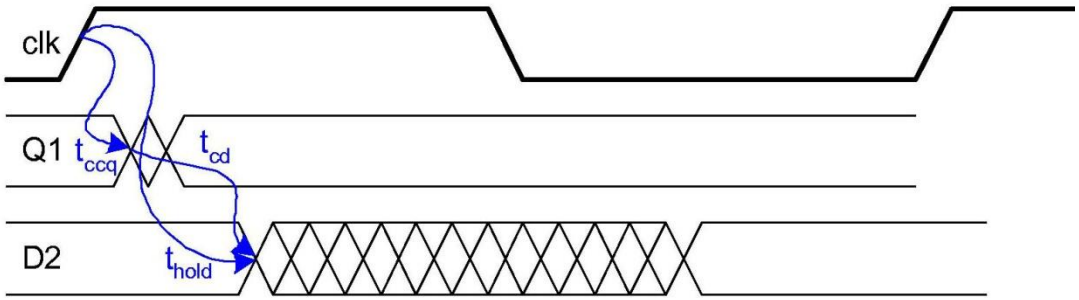


Min-Delay: Flip-Flop

- If the combinational logic is too fast, the signal can pass two FFs in one cycle.
- Race condition, hold-time failure.
- Worsen with clock skew, e.g. shift registers and scan chain.

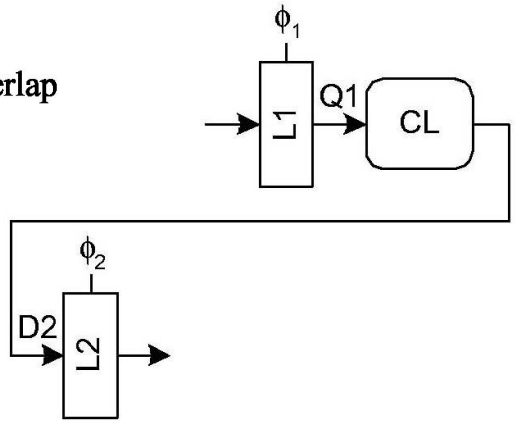
$$t_{cd} \geq t_{\text{hold}} - t_{ccq}$$

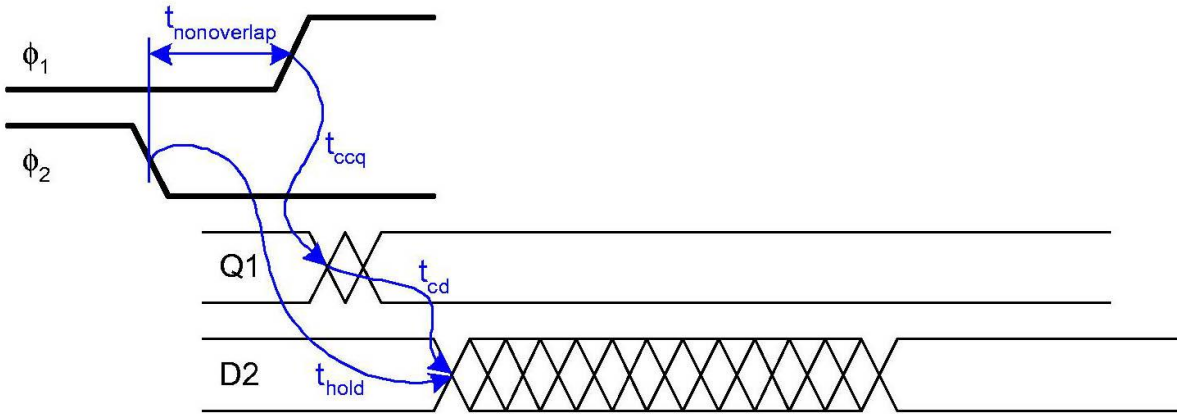




Min-Delay: 2-Phase Latch

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

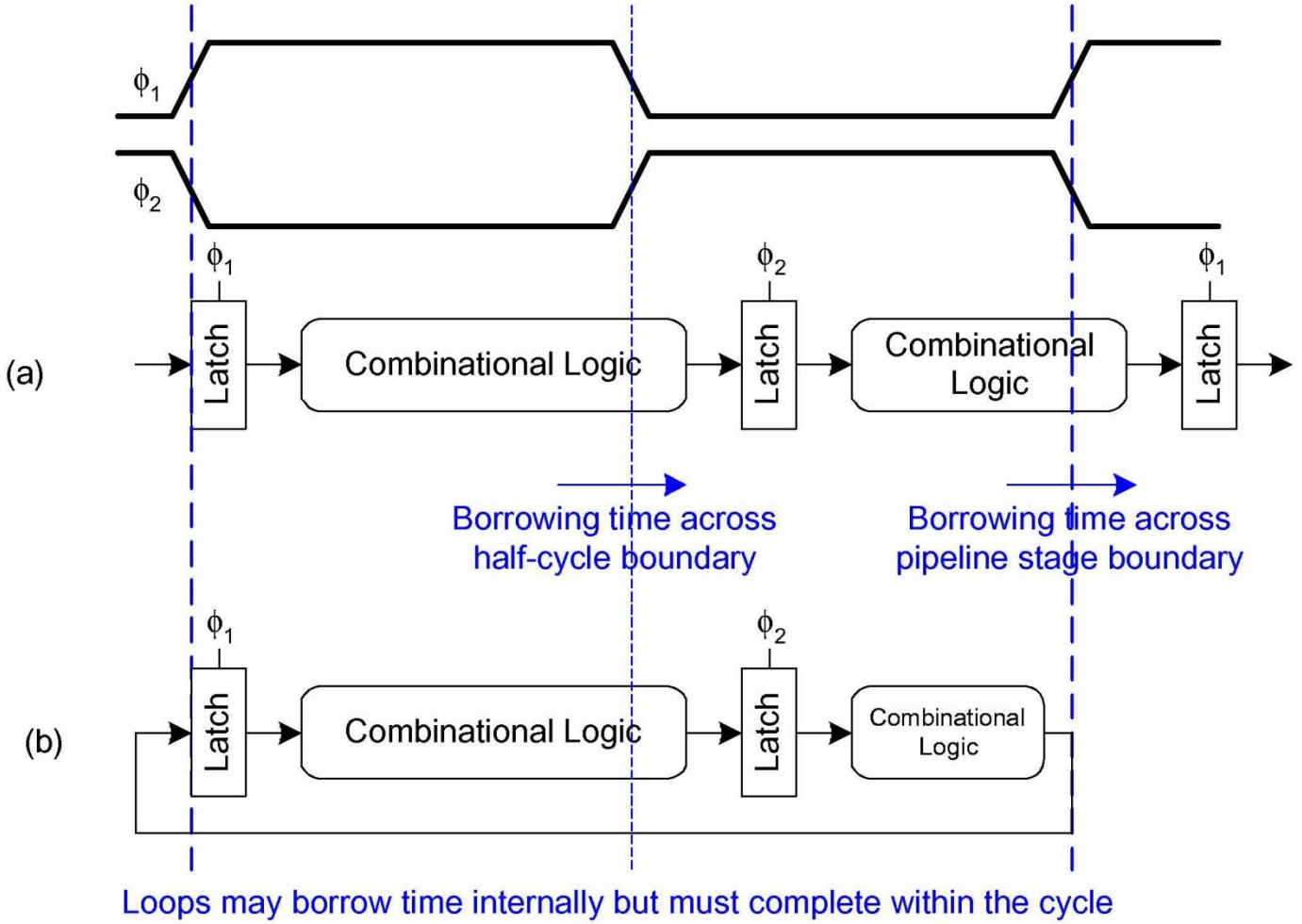




Time Borrowing

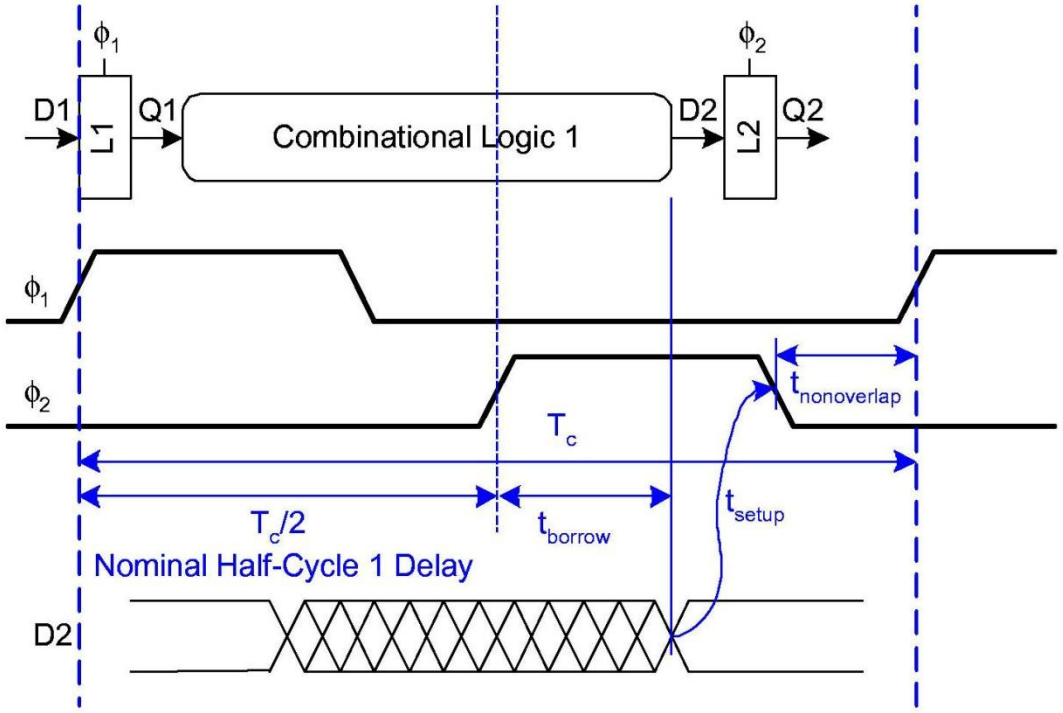
- In a flip-flop-based system:
  - A piece of new data launches on one rising edge
  - Must setup before next rising edge
  - If it arrives late, system fails
  - If it arrives early, time is wasted
- FFs have hard edges, i.e. each stage of logic has identical timing limits as others.
- In a latch-based system
  - Data can pass through latch while transparent
  - A stage of logic with longer delay can borrow time from the next stage with shorter delay
  - Must be careful to prevent the total borrowed time from longer than one stage

Time Borrowing Examples



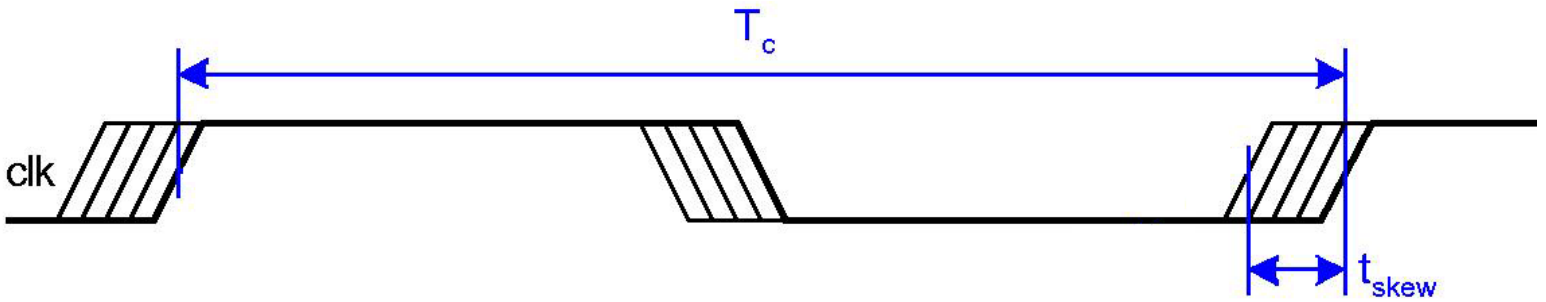
Maximum Time Borrowing

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



Clock Skew

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

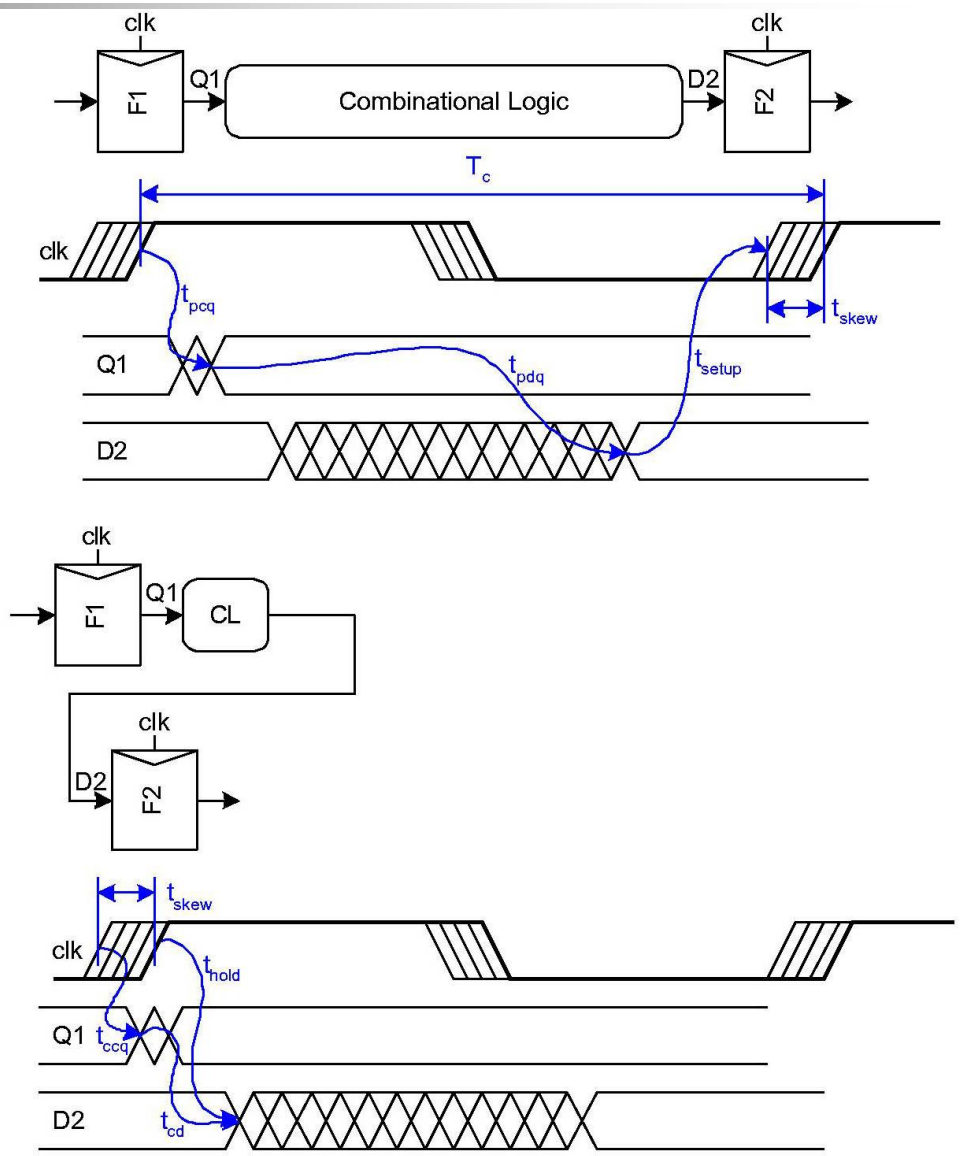


Skew: Flip-Flop

- Both delays are worsened by a clock skew time.

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

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



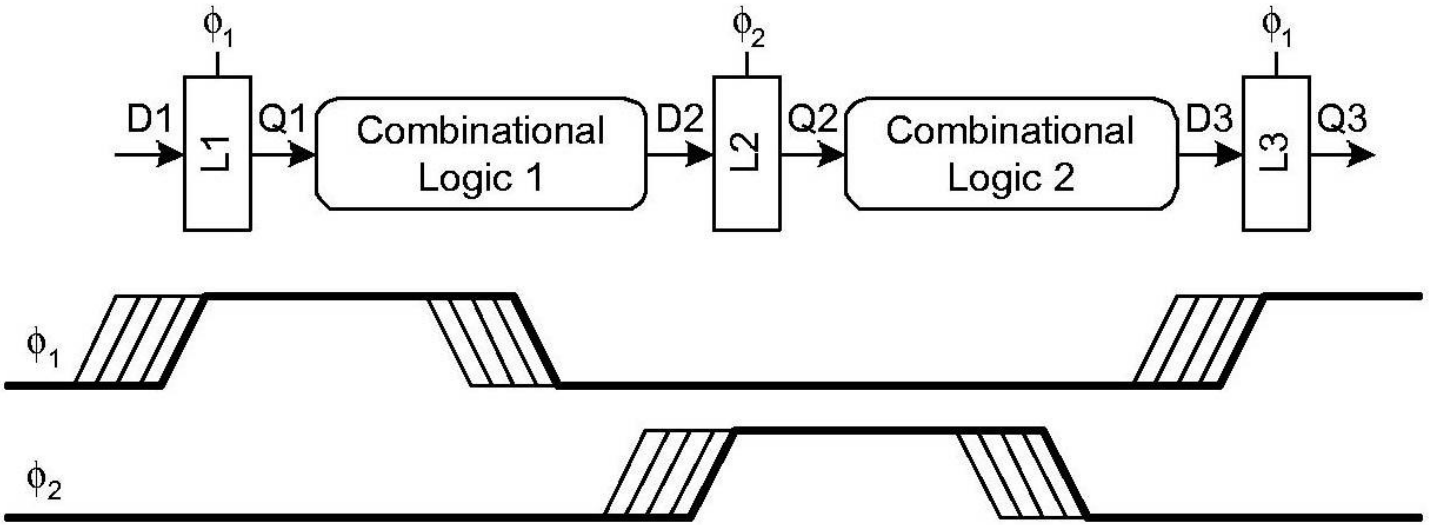
Skew: 2-Phase Latch

- Propagation delay unchanged.
- Skew tolerant.
- Other two times are worsened by a clock skew time.

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

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

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

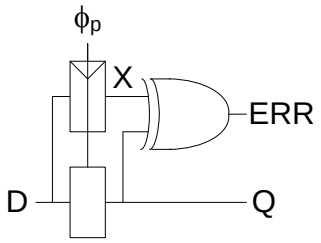


Two-Phase Clocking

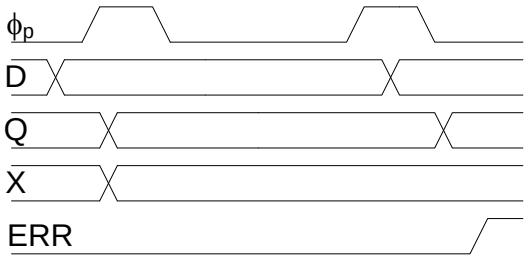
- If setup times are violated, increase clock period.
- If hold times are violated, chip fails at any speed.
- A conservative solution to guarantee hold times is to use 2-phase latches with big nonoverlap times.
- With large $t_{nonoverlap}$, the inequality will always be true.

For latch-based sequencing: $t_{cd1,}t_{cd2} \geq t_{\text{hold}} - t_{ccq} - t_{\text{nonoverlap}}$

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"
    - Basic idea is to sample "D" twice and if the outcomes are different, then "ERR"
    - 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