DIGITAL VLSI DESIGN

Rekha S S
Assistant professor
Department of Electronics and Communication Engg. ,PESU, RR campus

UNIT 4

Sequential Circuit Design

Sequential Circuits Design

Reference Books:
R2: CMOS VLSI design A Circuits and Systems Perspective, Neil Weste and David Harris, 3rd Edition. Chapter 10 (10.1 to 10.2) – as per Soft copy Sequencing static circuits, Sequencing Methods, Max-Delay Constraints, Min-Delay Constraints, Time Borrowing, Clock Skew, Problems on Max and Min Delay Constraints at design level **R1:** CMOS Digital Integrated Circuits Analysis and Design, Sung-Mo (Steve) Kang,3rd Edition. Introduction, Behavior of Bi-stable Elements, The SR Latch Circuit (8.1,8.2, 8.3) Clocked Latch and Flip Flop Circuits (8.4) CMOS D-Latch and Edge-Triggered Flip-Flop (8.5)

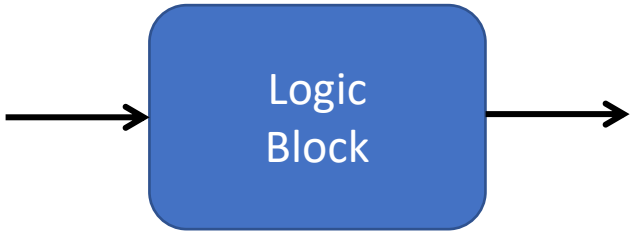
INTRODUCTION

Digital Circuits

- **Combinational Circuits**
- **Sequential Circuits**

- Output is a function of the current inputs

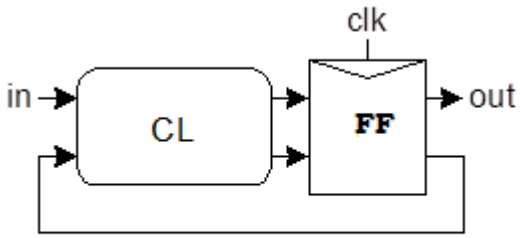
X y = f(X)



- Output depends on previous as well as current inputs
- Such circuit are said to have state (like present state, next state, previous state)
- FSM (finite state machine) and pipeline are two examples of sequential circuits.

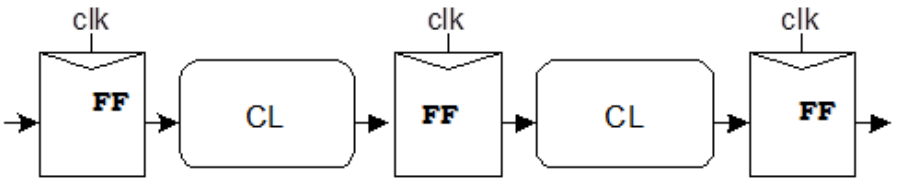
INTRODUCTION

- **FSM**

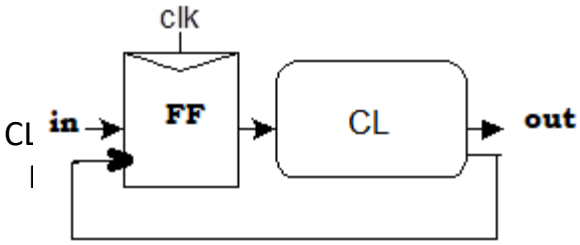


Finite State Machine

- **Pipeline**

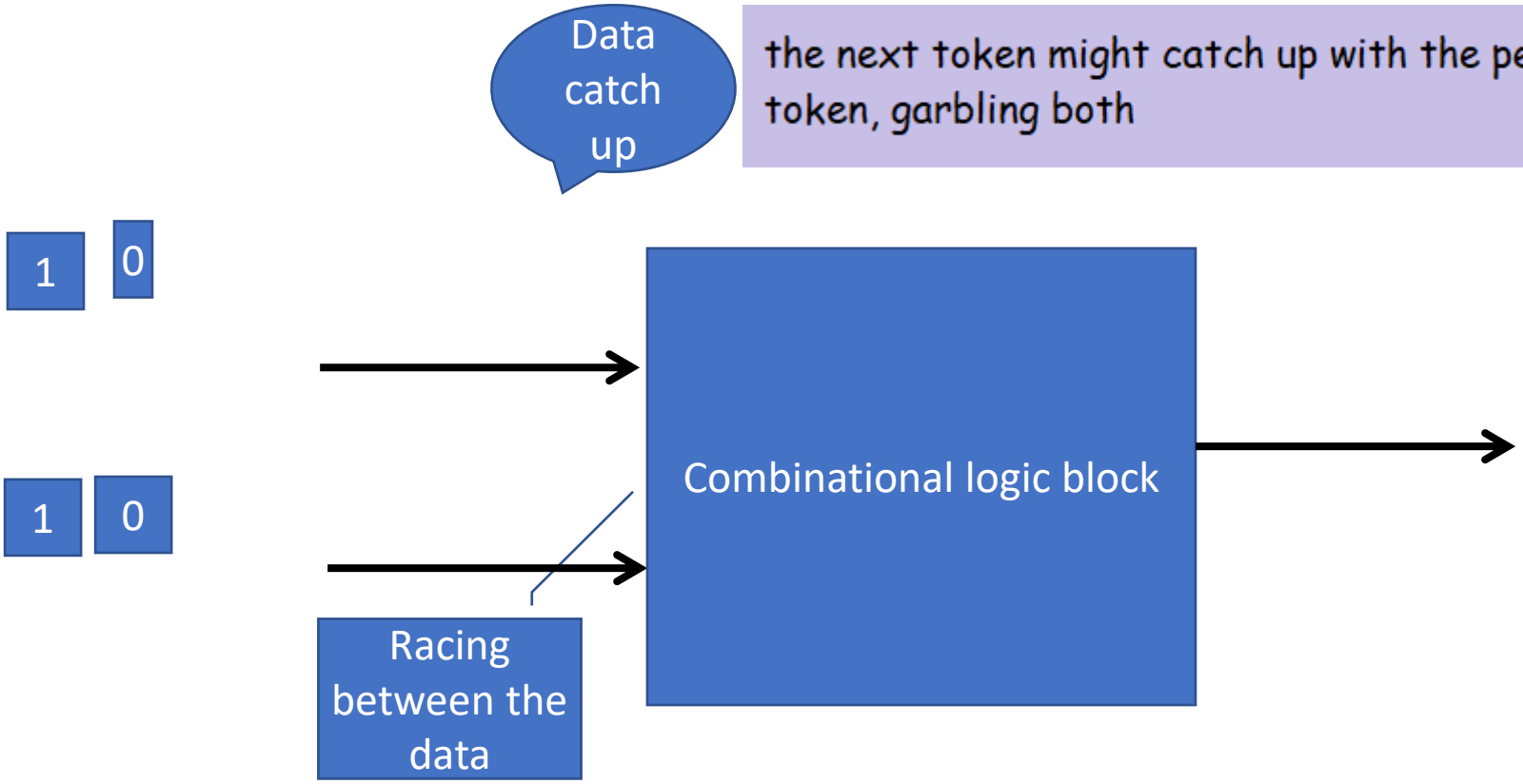


Pipeline



Sequential Circuits Design

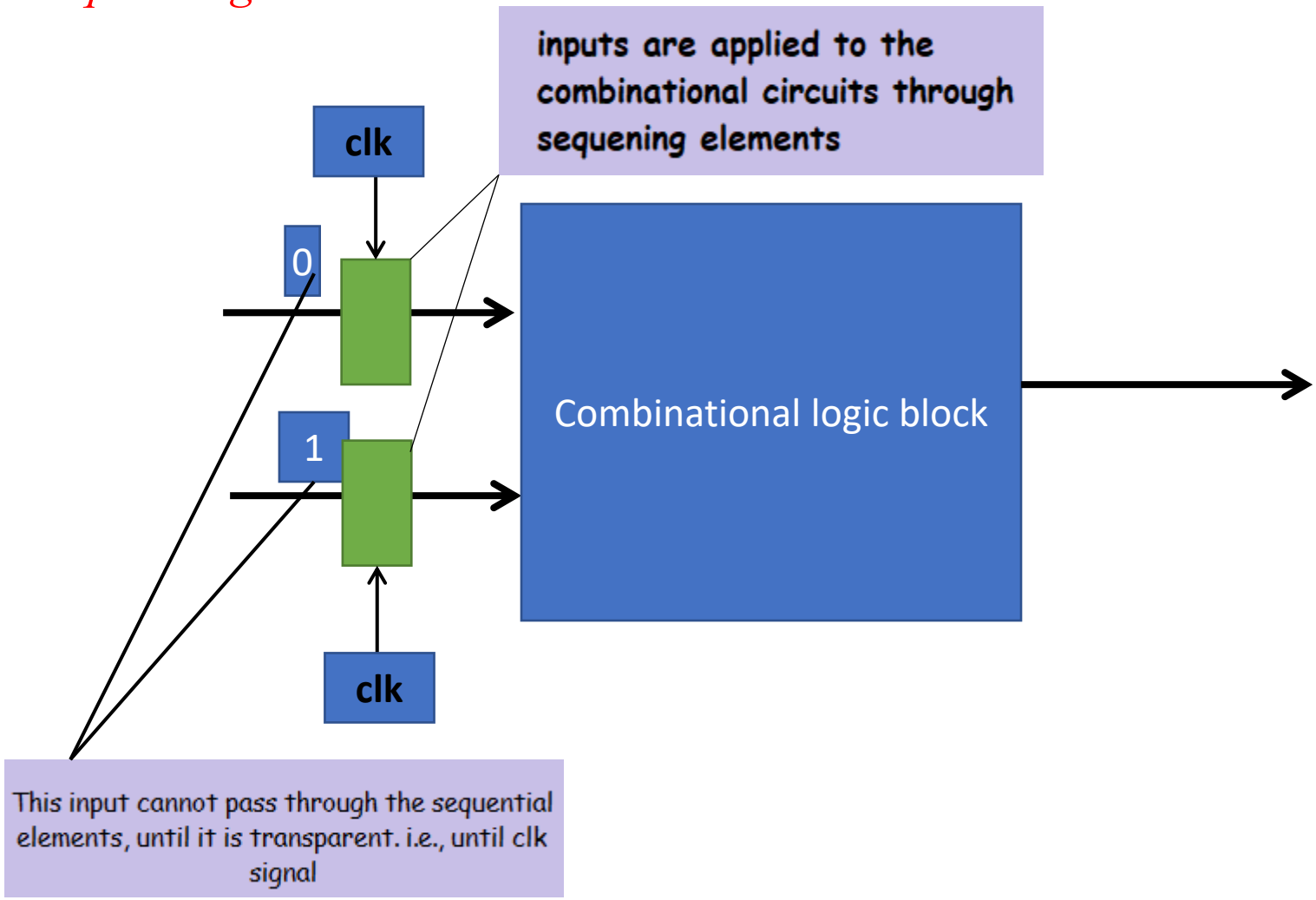
- Flip flops or latches, are sometimes called memory elements, that hold data called tokens.



- The purpose of these (Flip flop or latch)elements is not really memory; instead, it is to enforce sequence, to distinguish the ***current token*** from the ***previous or next token*** *-- Sequencing elements.*

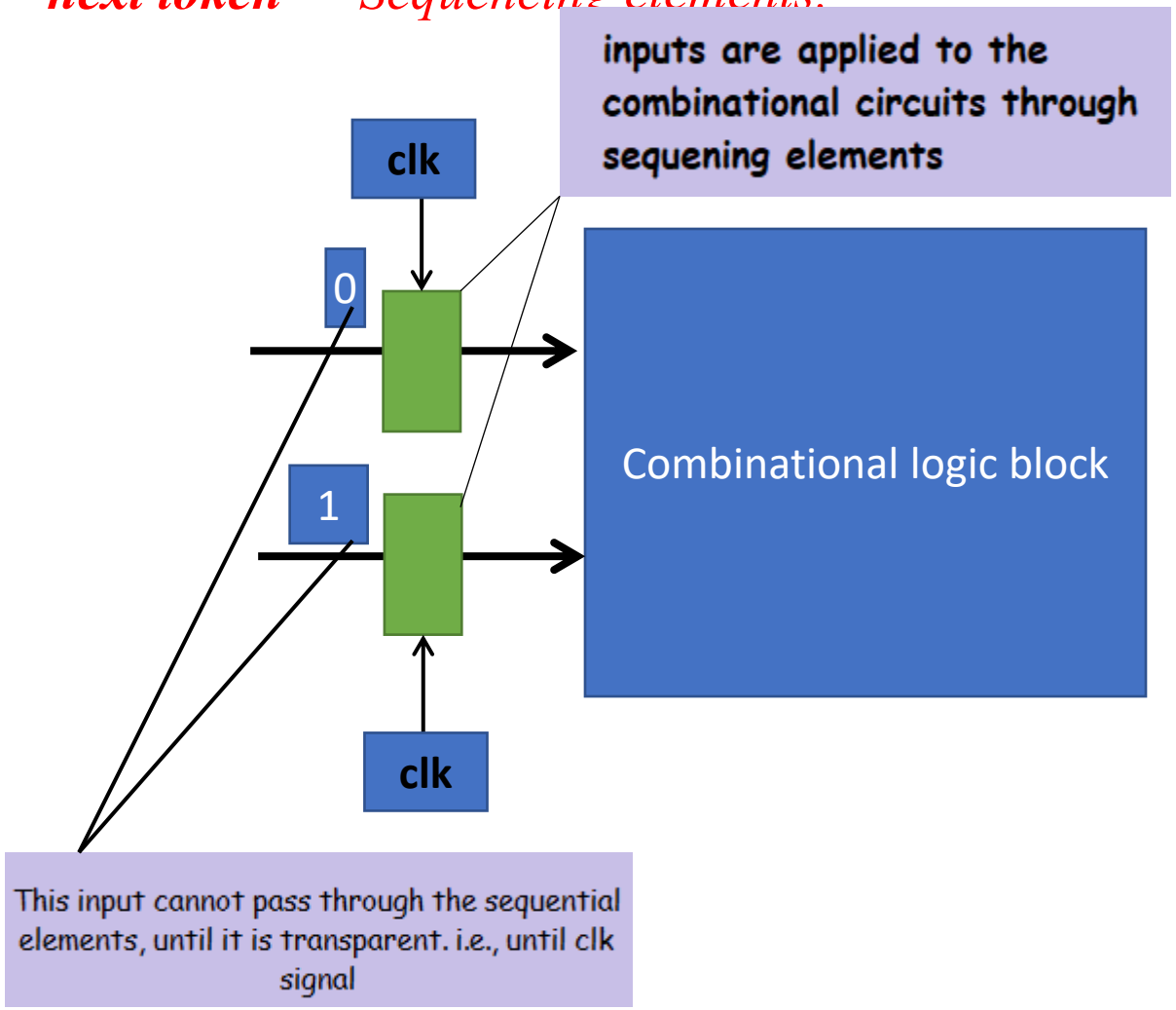
Sequential Circuits Design

- The purpose of these elements is not really memory; instead, it is to enforce sequence, to distinguish the ***current token*** from the ***previous or next token*** -- *Sequencing elements.*



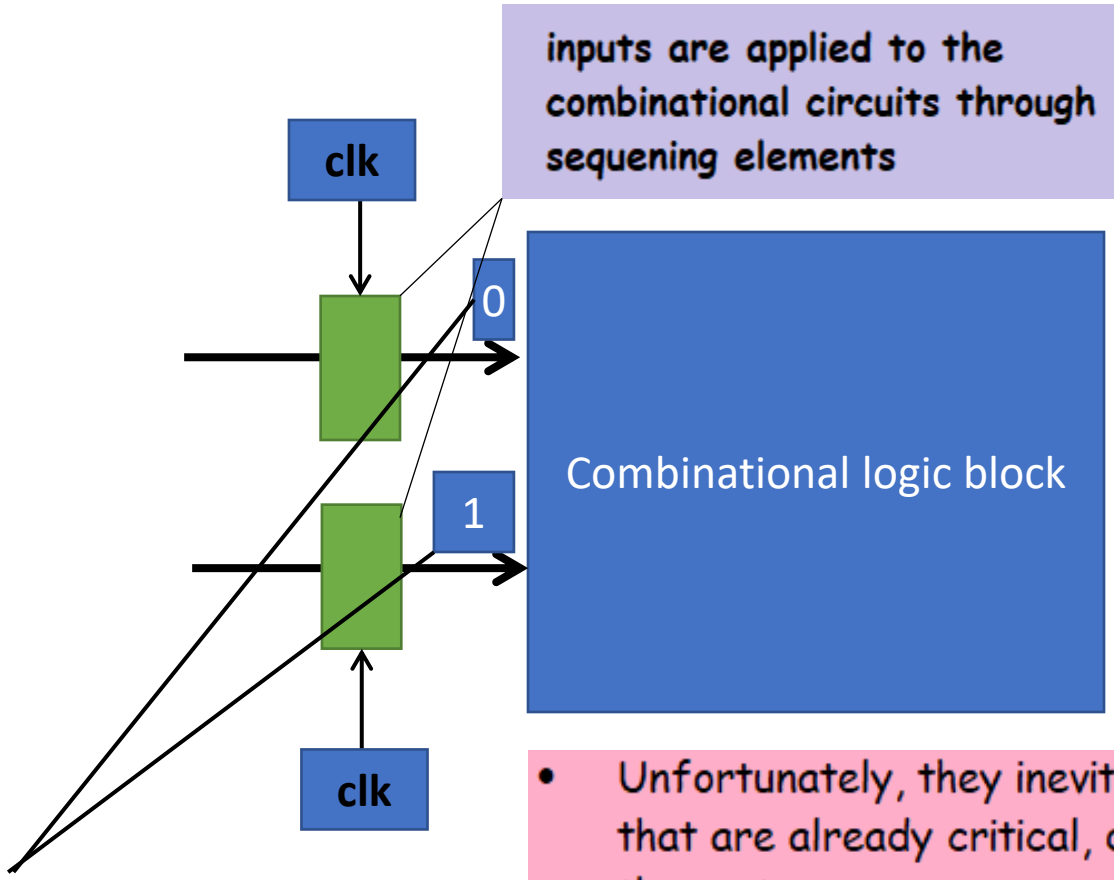
Sequential Circuits Design

- The purpose of these elements in not really memory; instead, it is to enforce sequence, to distinguish the ***current token*** from the ***previous or next token*** -- *Sequencing elements.*



Sequential Circuits Design

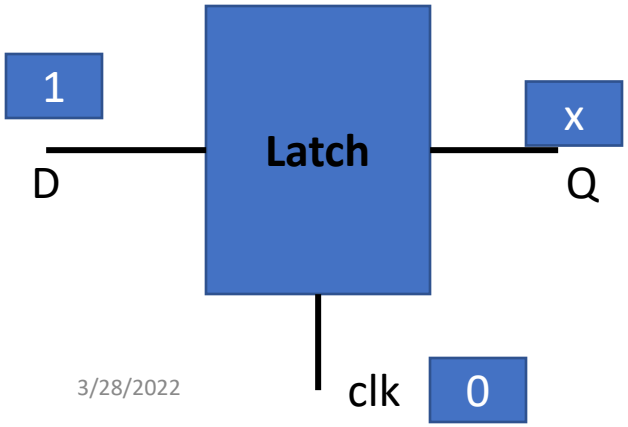
- without sequencing elements, the next token might catch up with the pervious token, garbling both.
  - Sequencing elements delay tokens that arrive too early, preventing them from catch up with the pervious tokens



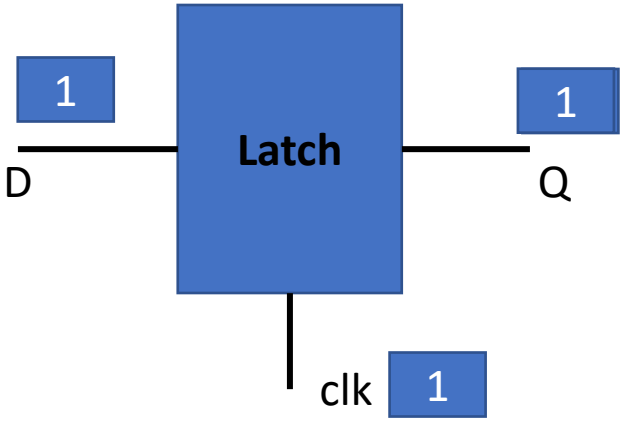
- Unfortunately, they inevitably add some delay to tokens that are already critical, decreasing the performance of the system.
- This extra delay is called ***Sequencing Overhead***

Sequencing static circuits

- Flip Flop and Latches are most commonly used sequencing elements.
- Both have three terminals
  - Data input (D)
  - Data output (Q)
  - Clock Input (clk)
- Latch is transparent when the clock is high and opaque when the clock is low

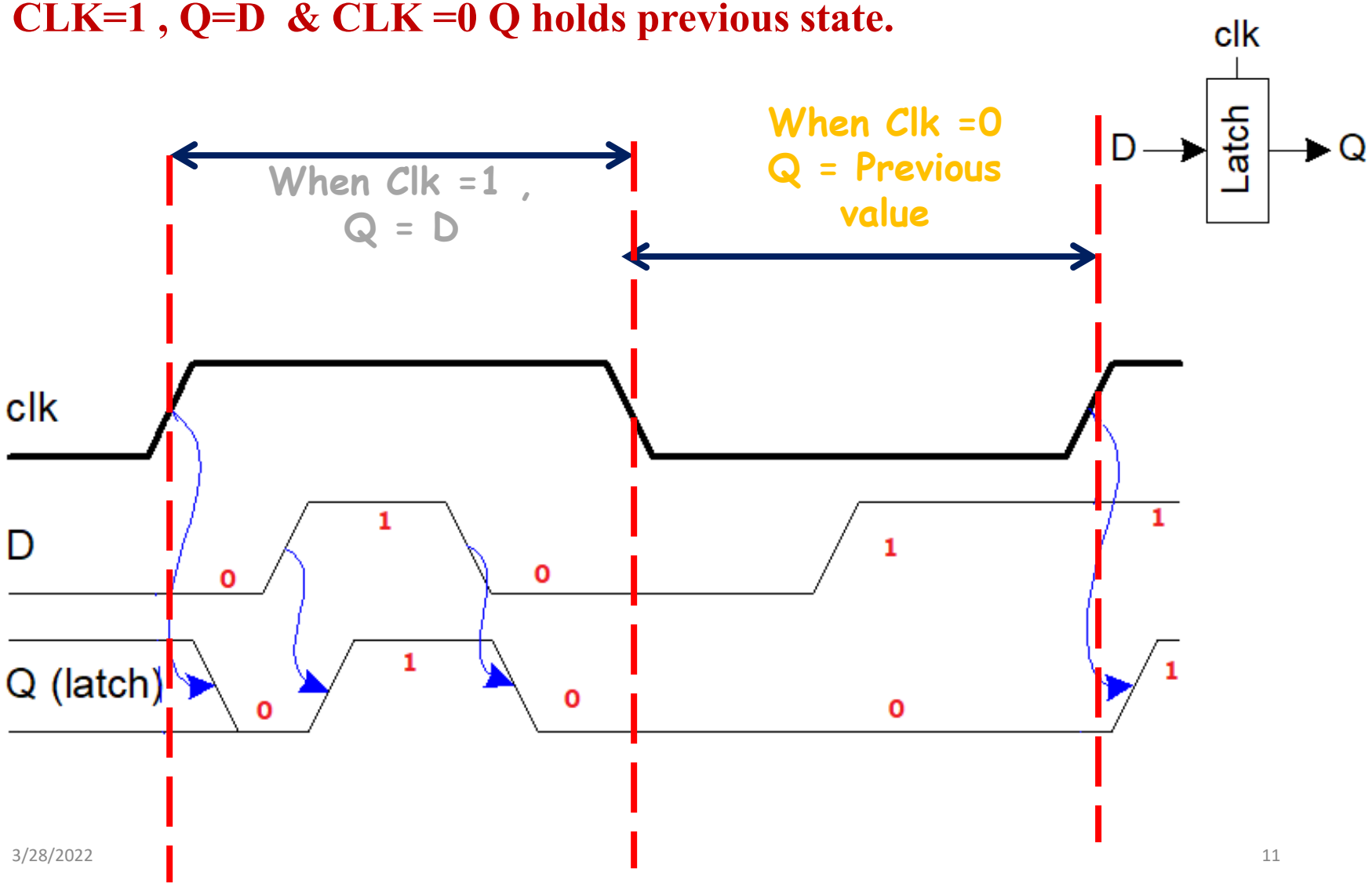


Clk	Q
1	D
0	Previous state



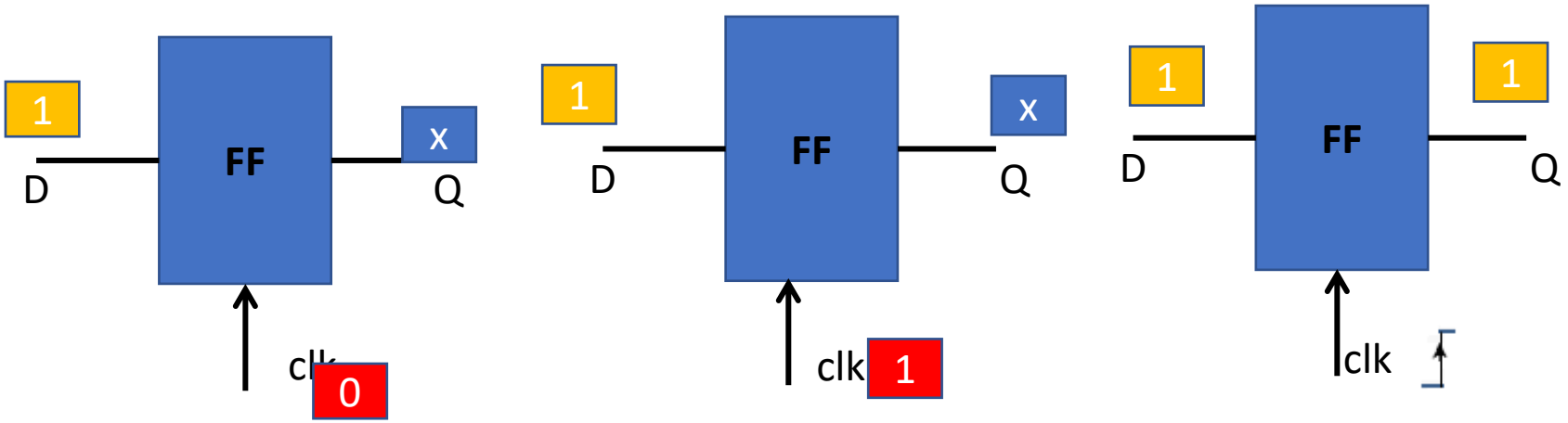
Sequencing static circuits

- **Latch:** **is transparent when clock is HIGH else retains previous state CLK=1 , Q=D & CLK =0 Q holds previous state.**



Sequencing static circuits

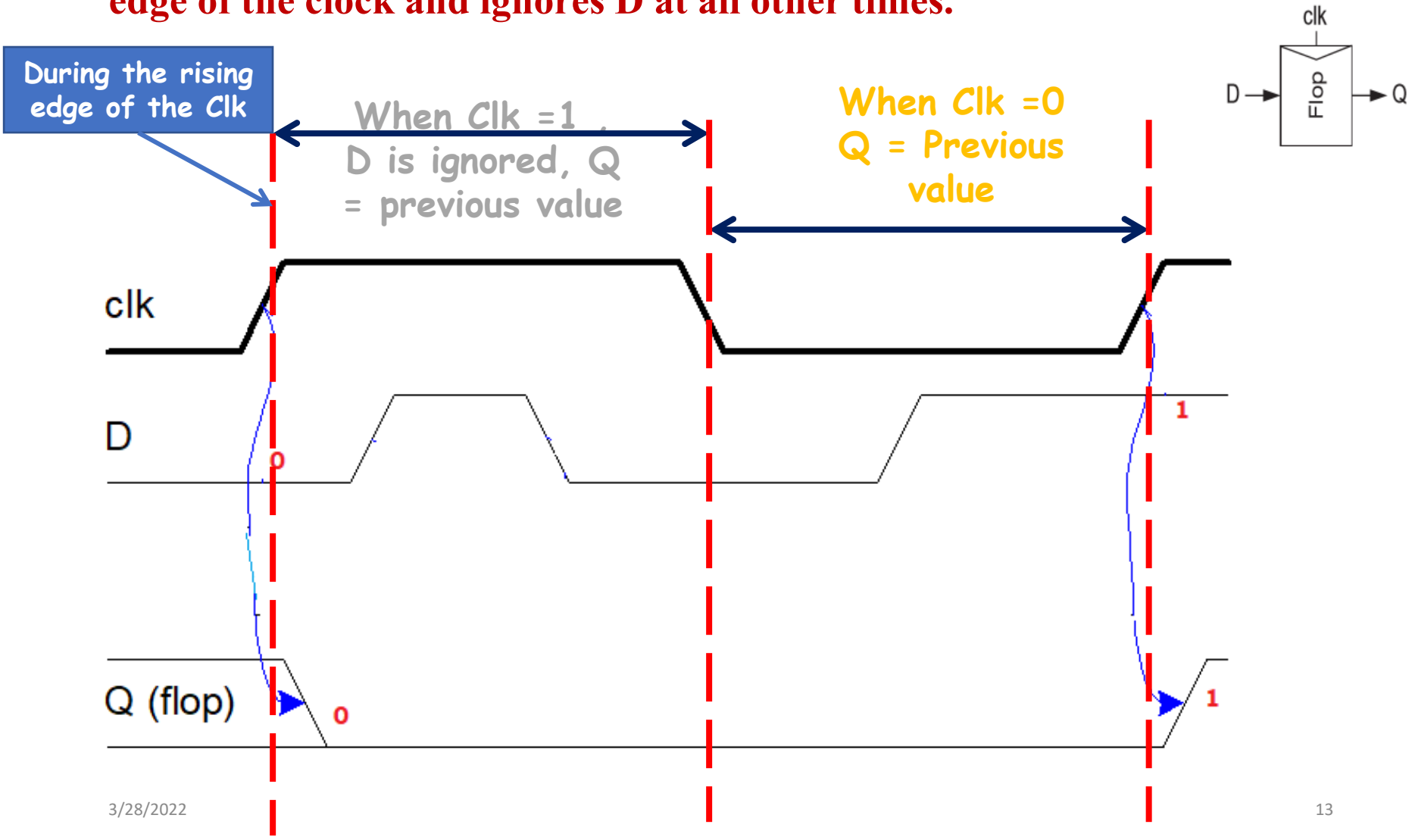
- **Flip Flop**: **is a +ve edge-triggered device, that copies D to Q on the rising edge of the clock and ignores D at all other times.**



Clk	Q
1	No change
0	No change
↑	D

Sequencing static circuits

- **Flip Flop:** **is a +ve edge-triggered device, that copies D to Q on the rising edge of the clock and ignores D at all other times.**



Observations

- The purpose of FF and latch is not really memory; instead, it is to enforce sequence, to distinguish the ***current token*** from the ***previous or next token*** *-- Sequencing elements.*
- without sequencing elements, the next token might catch up with the pervious token, garbling both.
- Sequencing elements delay tokens that arrive too early, preventing them from catch up with the pervious tokens
- Unfortunately, they inevitably add some delay to tokens that are already critical, decreasing the performance of the system.
- This extra delay is called ***Sequencing Overhead***

Sequencing Methods of Combinational Logic

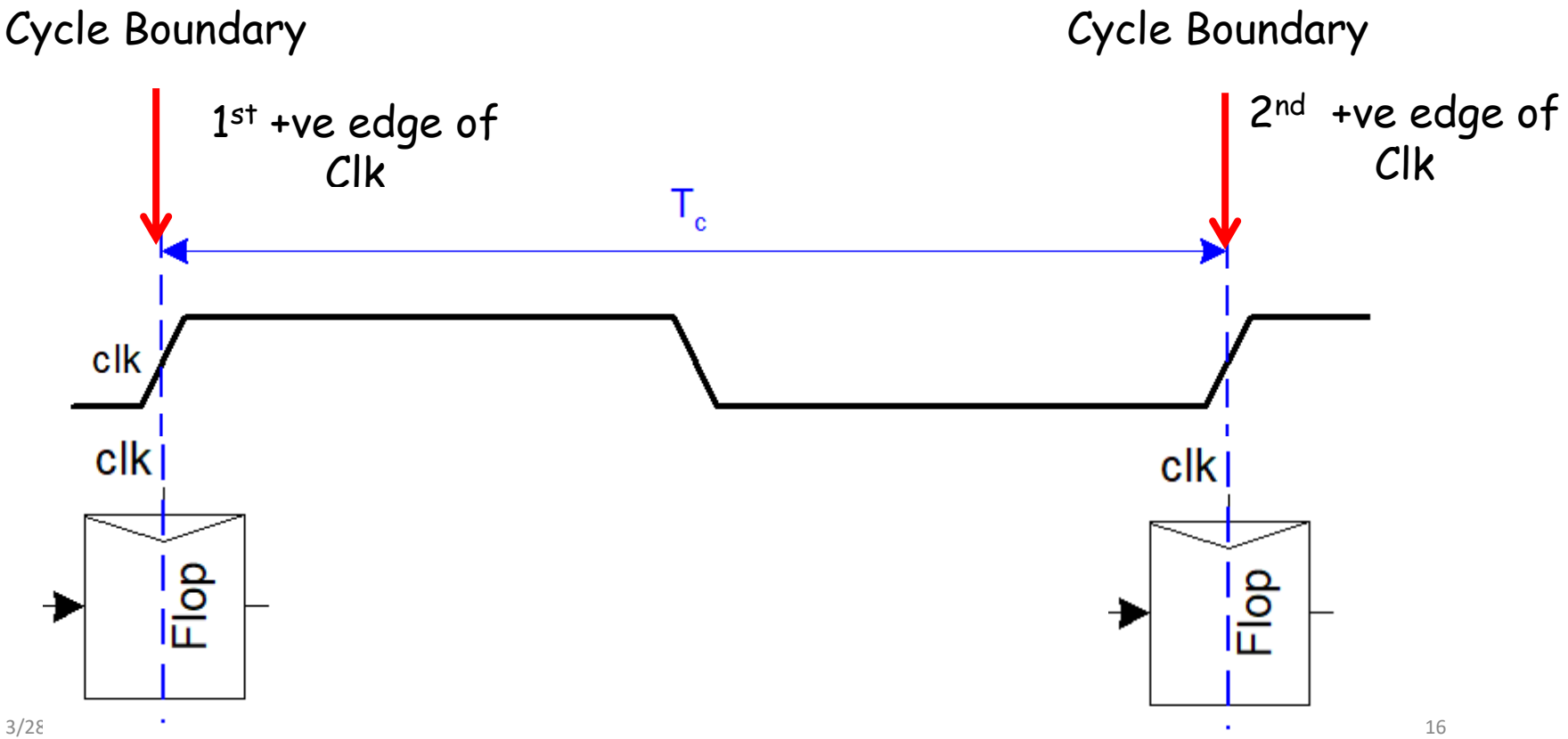
- There are three methods of sequencing blocks of combinational logic.

1. **Using Flip- Flop as Sequencing Element**
2. **Using Two phase Latches as Sequencing Element**
3. **Using Pulsed Latches as Sequencing Element**

Sequencing Methods of Combinational Logic

1. Using Flip- Flop as Sequencing Element

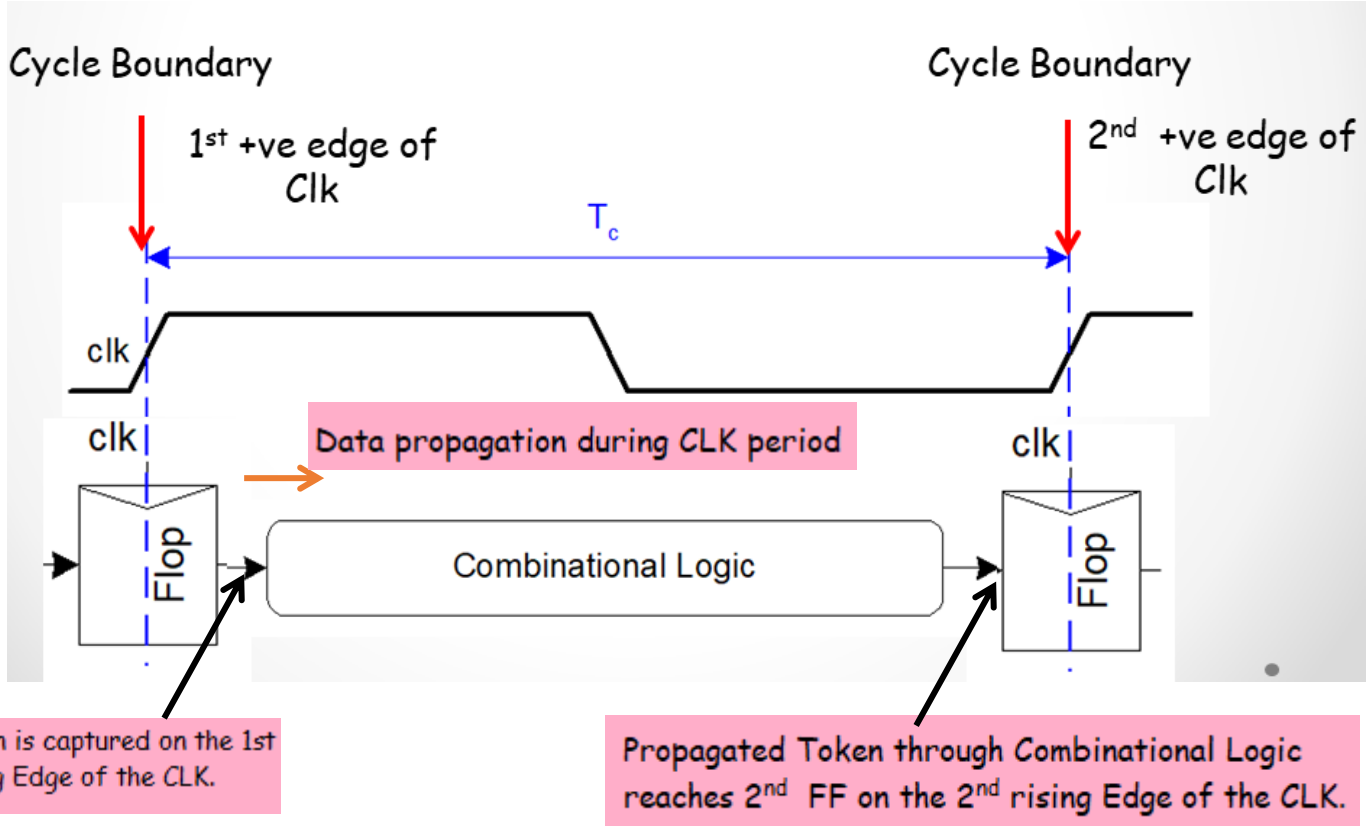
- Uses One Flip-Flop at each Cycle Boundary



Sequencing Methods of Combinational Logic

1. Using Flip- Flop as Sequencing Element

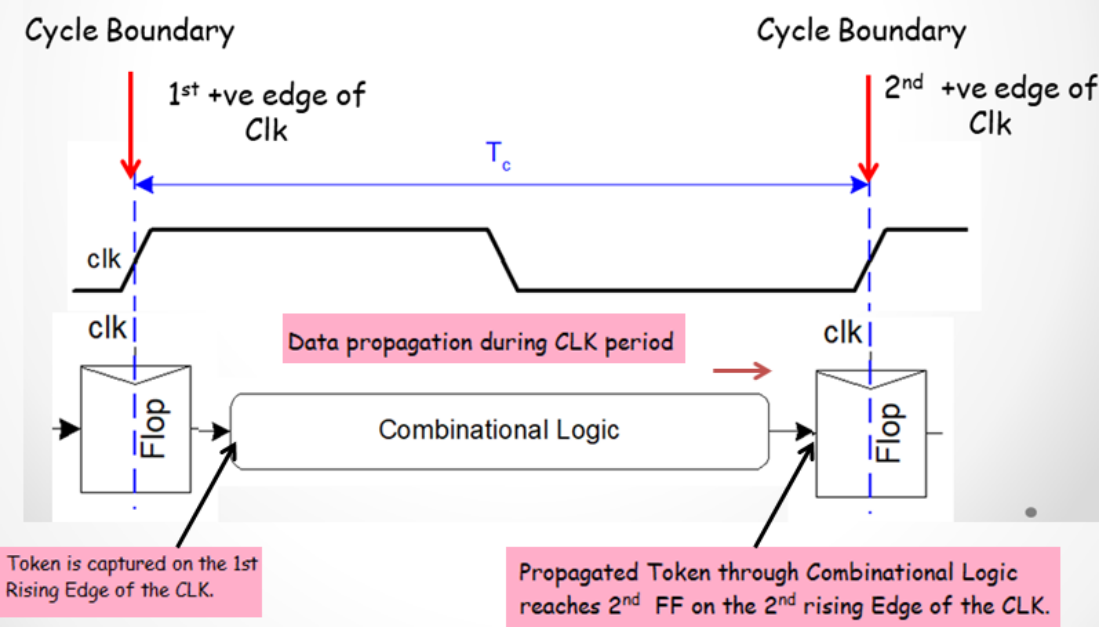
- Uses One Flip-Flop at each Cycle Boundary
- Tokens advances from one cycle to next on rising edge. If the token arrives early , it waits at the flip-flop unit the next cycle (rising edge)



Sequencing Methods of Combinational Logic

1. Using Flip- Flop as Sequencing Element

- Uses One Flip-Flop at each Cycle Boundary
- Tokens advances from one cycle to next on rising edge. If the token arrives early , it waits at the flip-flop unit the next cycle (rising edge)

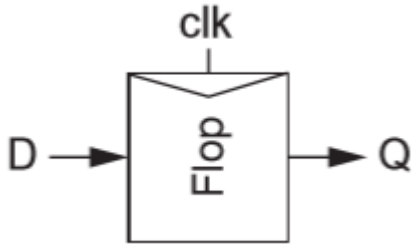


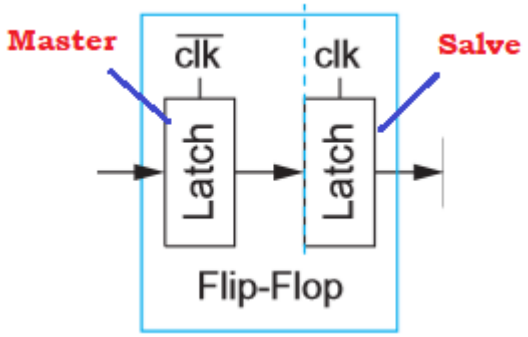
- So, data is captured in the first flip-flop on the first rising edge of the clock. Data propagates through combinational logic and reaches second flip-flop on the second rising edge of the clock.

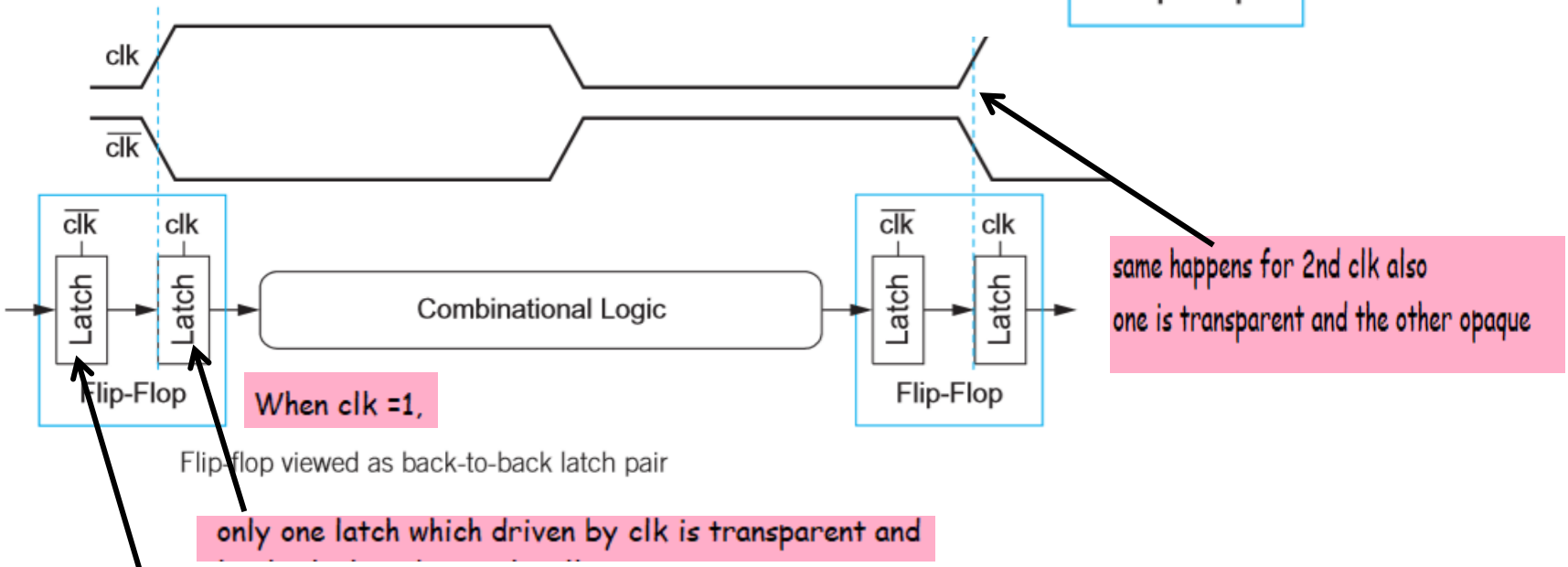
Sequencing Methods of Combinational Logic

2. Using Two phase Latches as Sequencing Element

Since Flip Flop can be constructed using latch as flows



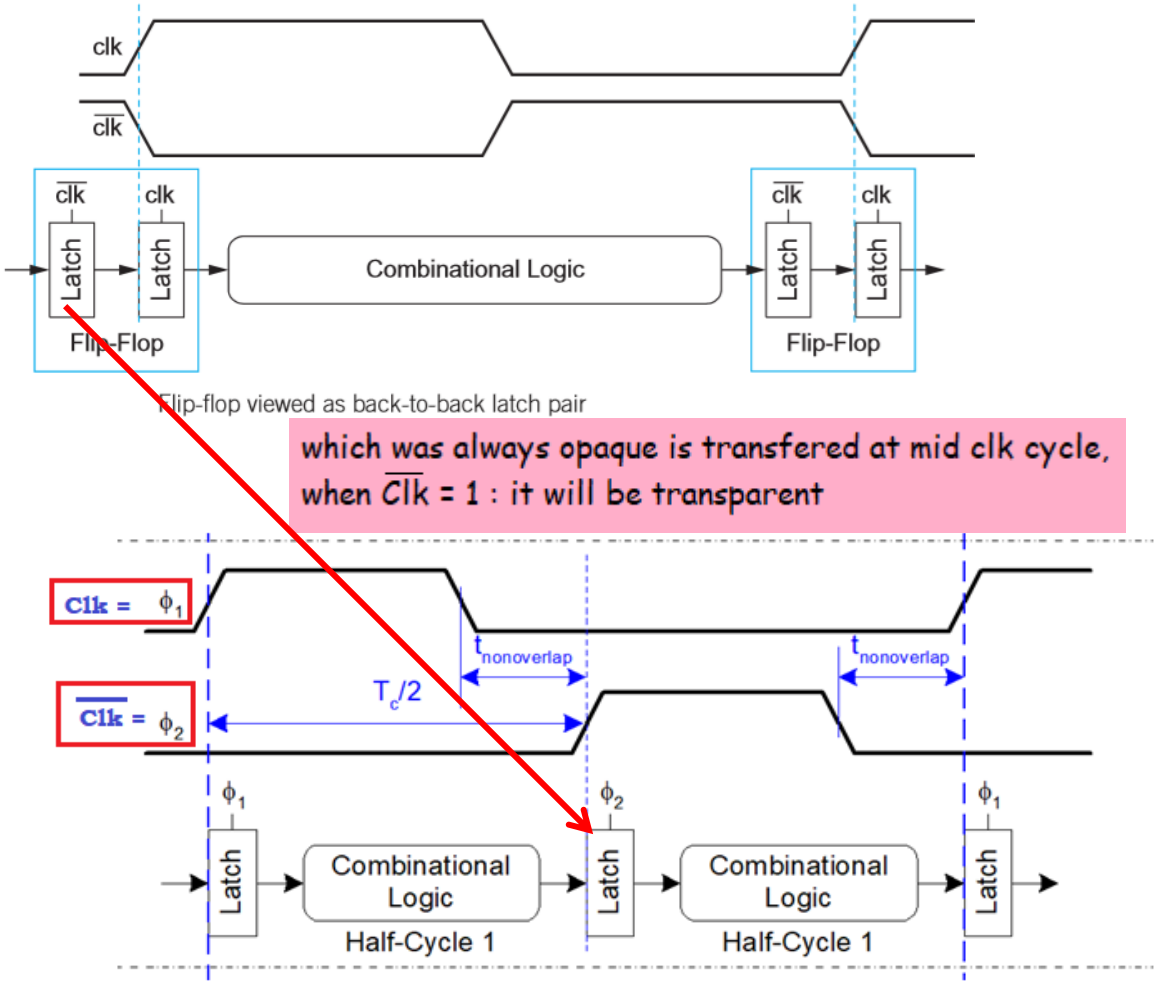




Sequencing Methods of Combinational Logic

2. Using Two phase Latches as Sequencing Element

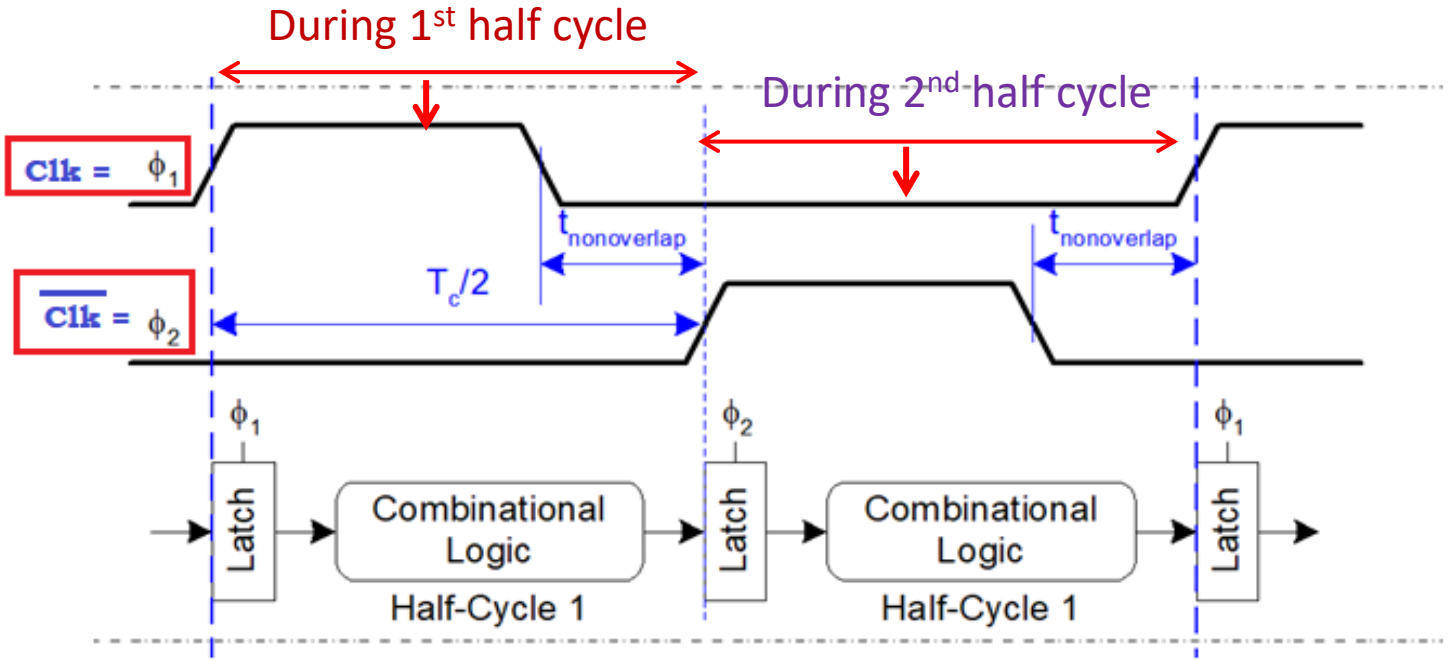
Since Flip Flop can be constructed using latch as flows



Sequencing Methods of Combinational Logic

2. Using Two phase Latches as Sequencing Element

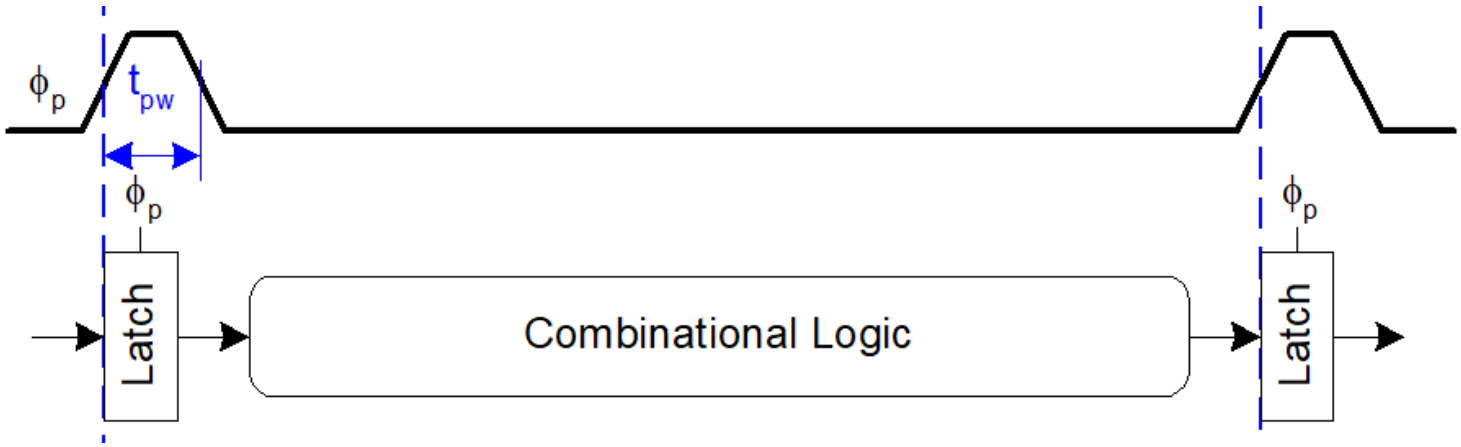
- Divide the full cycle of combinational logic into two phases, sometimes called *half-cycles*.
- Non Overlapping Two phase Clocks are used to Control two Latches and are referred as Φ1 and Φ2.
- At any given time, at least one clock is LOW and the corresponding latch is opaque, preventing one token from catching up with another.



Sequencing Methods of Combinational Logic

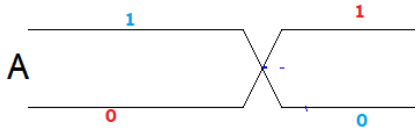
3. Using Pulsed Latches as Sequencing Element

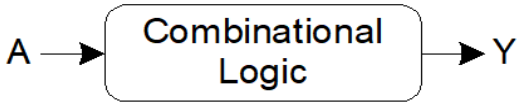
- Pulsed latch systems eliminate one of the latches from each cycle and apply a brief pulse to the remaining latch.
- If the pulse is shorter than the delay through the combinational logic, we can still expect that a token will only advance through one clock cycle on each pulse.



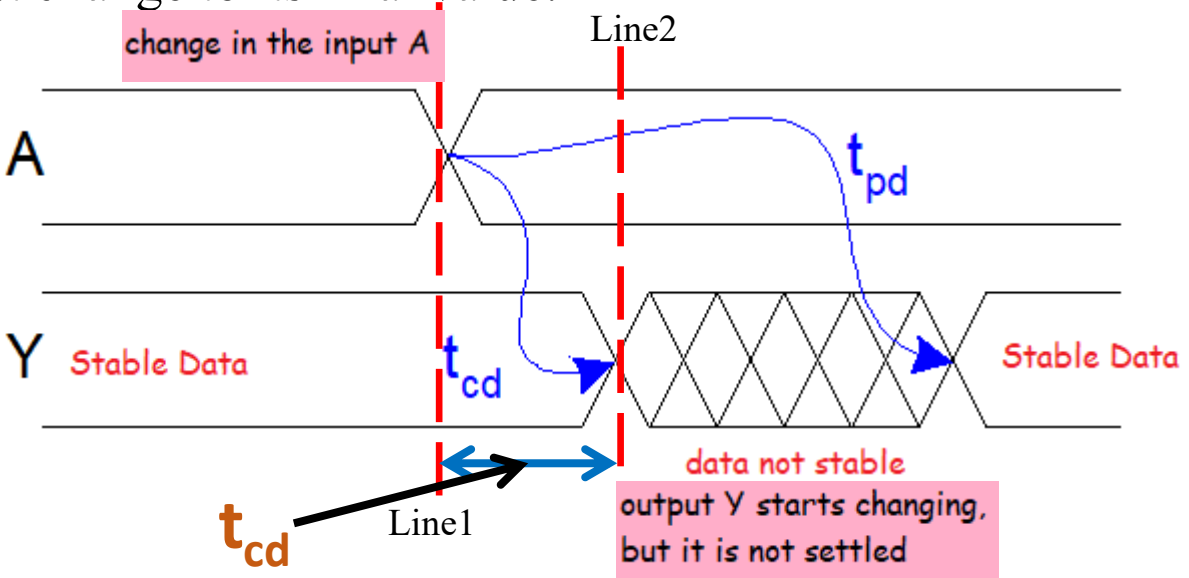
Delays and Timing Constraints of the combinational Logic

- Let the input *A* changing from one arbitrary value to another.



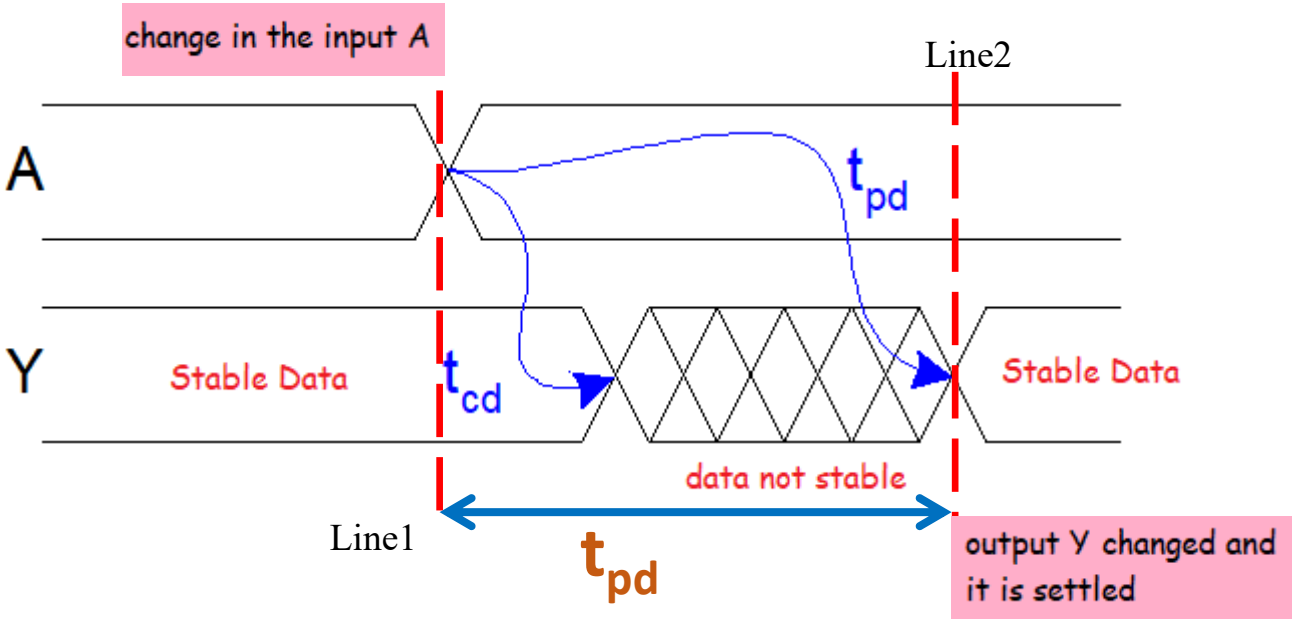


- But the output Y will not respond immediately. There are two different delays associated with comb. Logic
- **Contamination Delay (t_{cd}):** minimum amount of time from when an input changes until any output start to change its value, but the value has not reached the stable condition.
- **Propagation Delay (t_{pd})**: max. amount of time from when an input changes until output change to its final value.



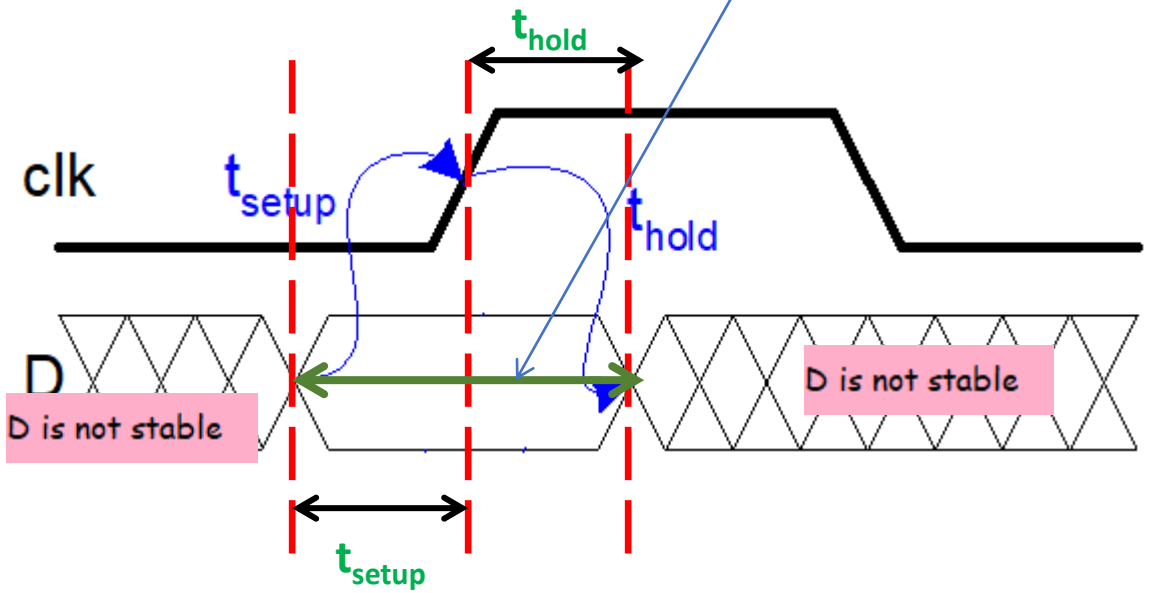
Delays and Timing Constraints of the combinational Logic

- **Propagation Delay (t_{pd})**: max. amount of time from when an input changes until output change to its final value.



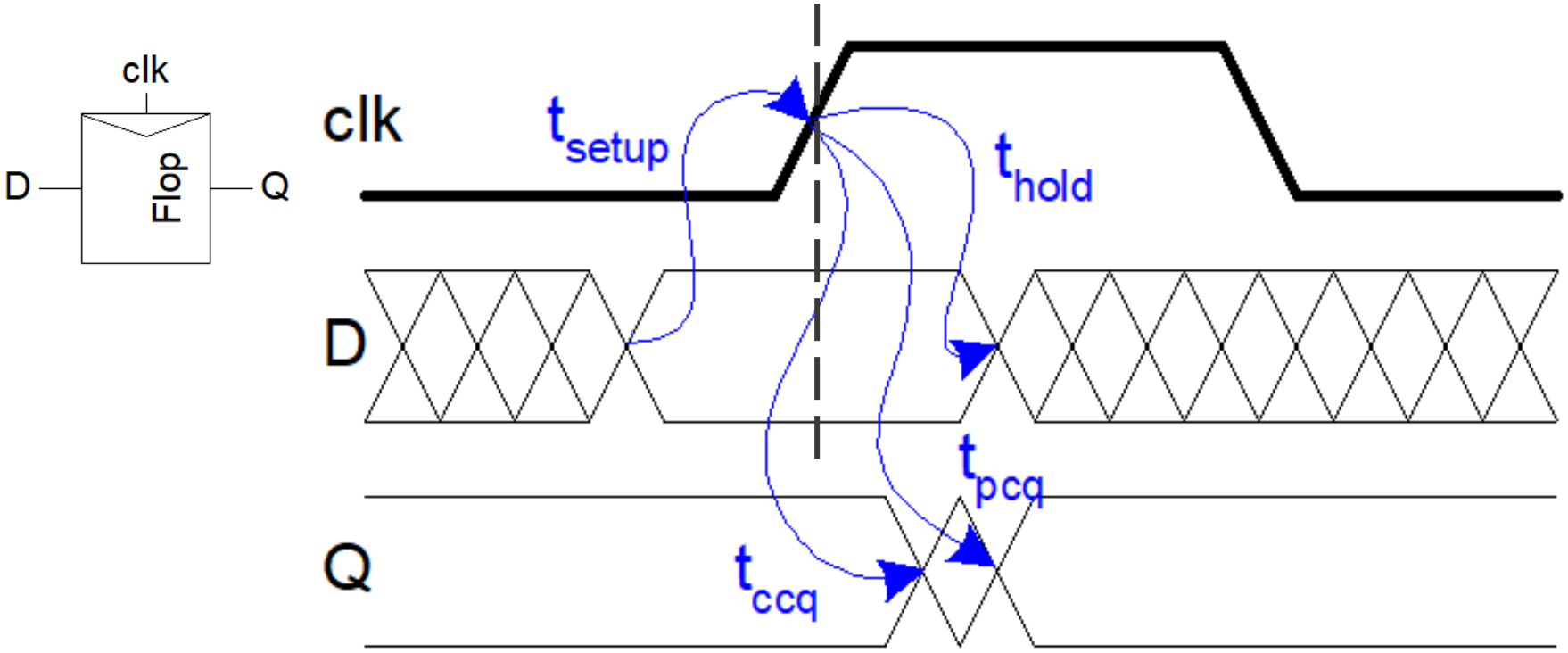
Delays and Timing Constraints of the Flip Flop

- The data input must be stable for some window around the rising edge of the flip flop if it is to be accurately sampled.
- Thant means, a minimum amount of time where the data should be stable (should not change) before clock goes high is known as **Setup time (t_{setup})** of FF.
- A minimum amount of time where the data should be stable after clock goes high is called as **Hold time (t_{hold})** of FF.

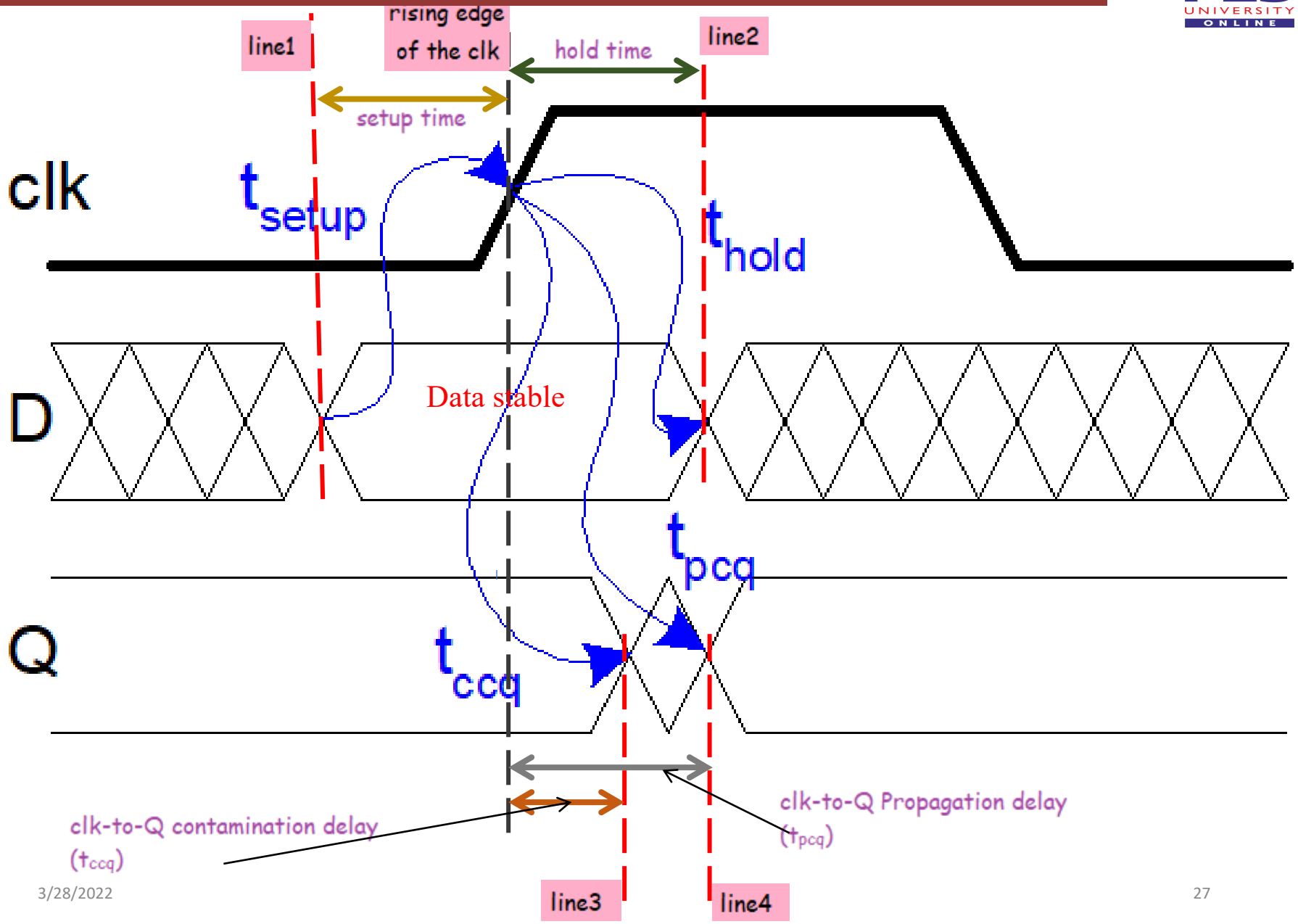


Delays and Timing Constraints of the Flip Flop

- Since in FF change in output occurs with respect to the rising edge of the clk.
- The output of FF begins to change after clk-to-Q contamination delay (t_{ccq}) and completely settles after a clk-to-Q propagation dealy (t_{pcq}).

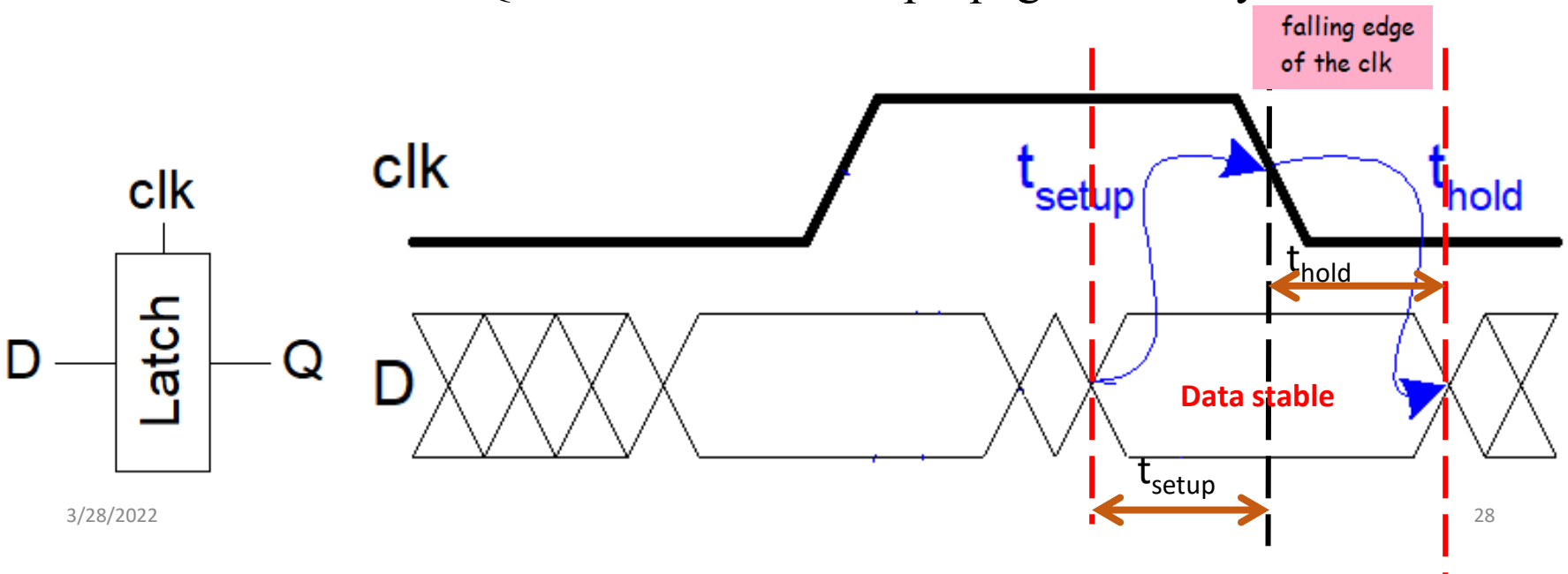


Sequential Circuits Design



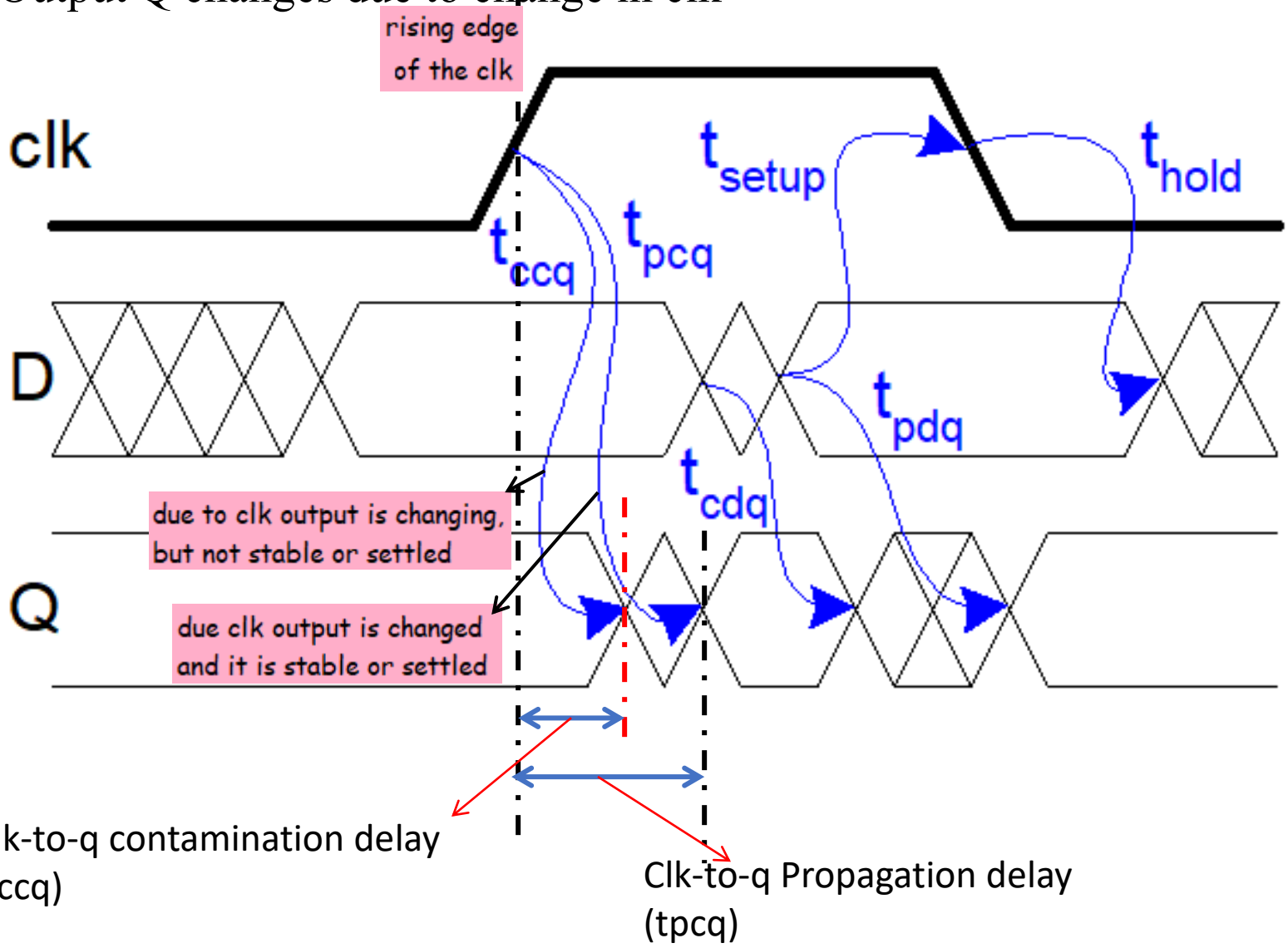
Delays and Timing Constraints of the Latch

- Since D latch is level sensitive, that is, it is changes in D is reflected at Q when clk =1 and opaque when clk=0.
- Hence input D must setup and hold around the falling edge that defines the end of the sampling period.
- Since output Q changes due to the clk, so it is associated with clk-to-Q contamination and propagation delay.
- And also output Q changes due to change in D when Clk =1, so it is associated with D-to-Q contamination and propagation delay.



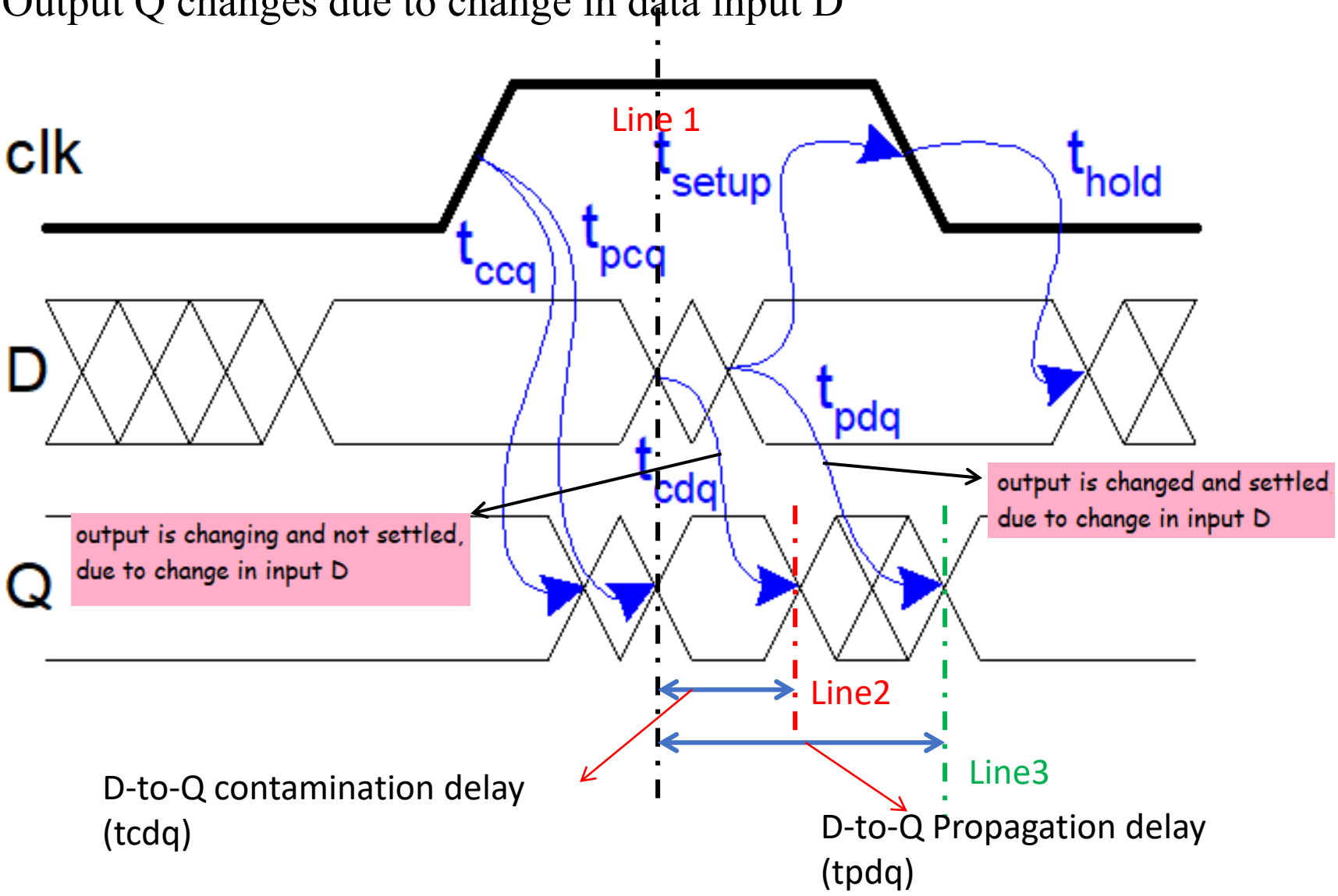
Sequential Circuits Design

- Output Q changes due to change in clk



Sequential Circuits Design

- Output Q changes due to change in data input D



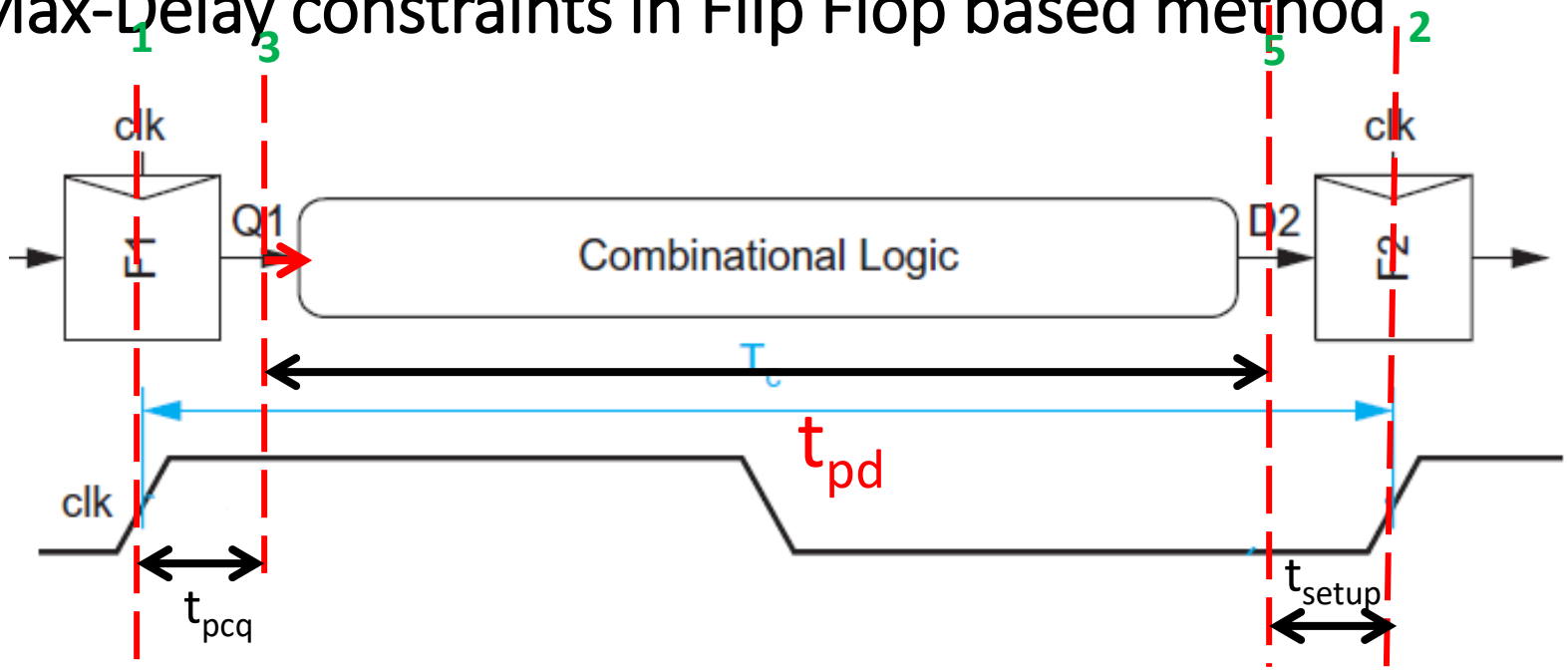
Sequencing element timing notation

Term	Name
t_{pd}	Logic Propagation Delay
t_{cd}	Logic Contamination Delay
t_{pcq}	Latch/Flop Clock-to-Q Propagation Delay
t_{ccq}	Latch/Flop Clock-to-Q Contamination Delay
t_{pdq}	Latch D-to-Q Propagation Delay
t_{cdq}	Latch D-to-Q Contamination Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

Max-Delay constraints

- Ideally, the entire clock cycle would be available for computations in the combinational logic
- However, in the real logic, the entire clock cycle is not available for the computations in the combinational logic.
- This is due to the sequencing overheads.
- **Setup time failure or max-delay failure**: if the combinational logic delay is too great, the receiving element will miss its setup time and sample the wrong value. This is called a setup time failure or max-delay failure.

Max-Delay constraints in Flip Flop based method



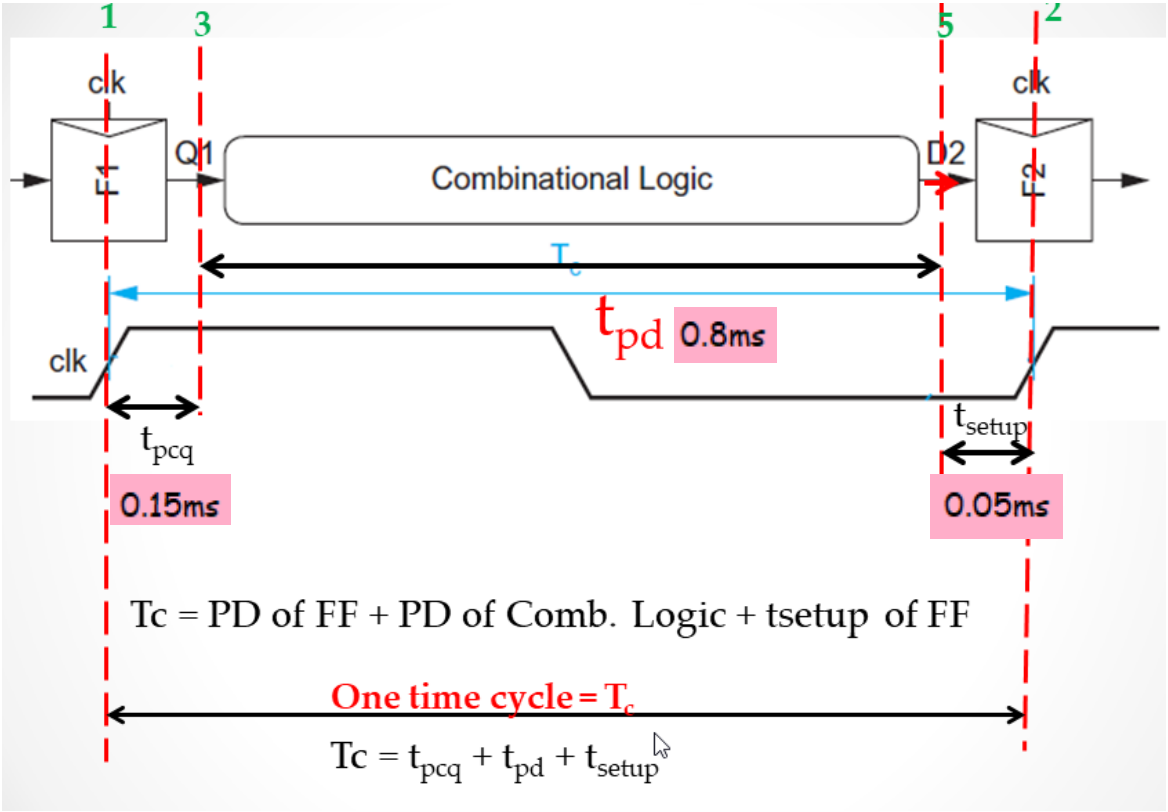
Tc = PD of FF + PD of Comb. Logic + tsetup of FF

One time cycle = T_c

$Tc = t_{pcq} + t_{pd} + t_{setup}$

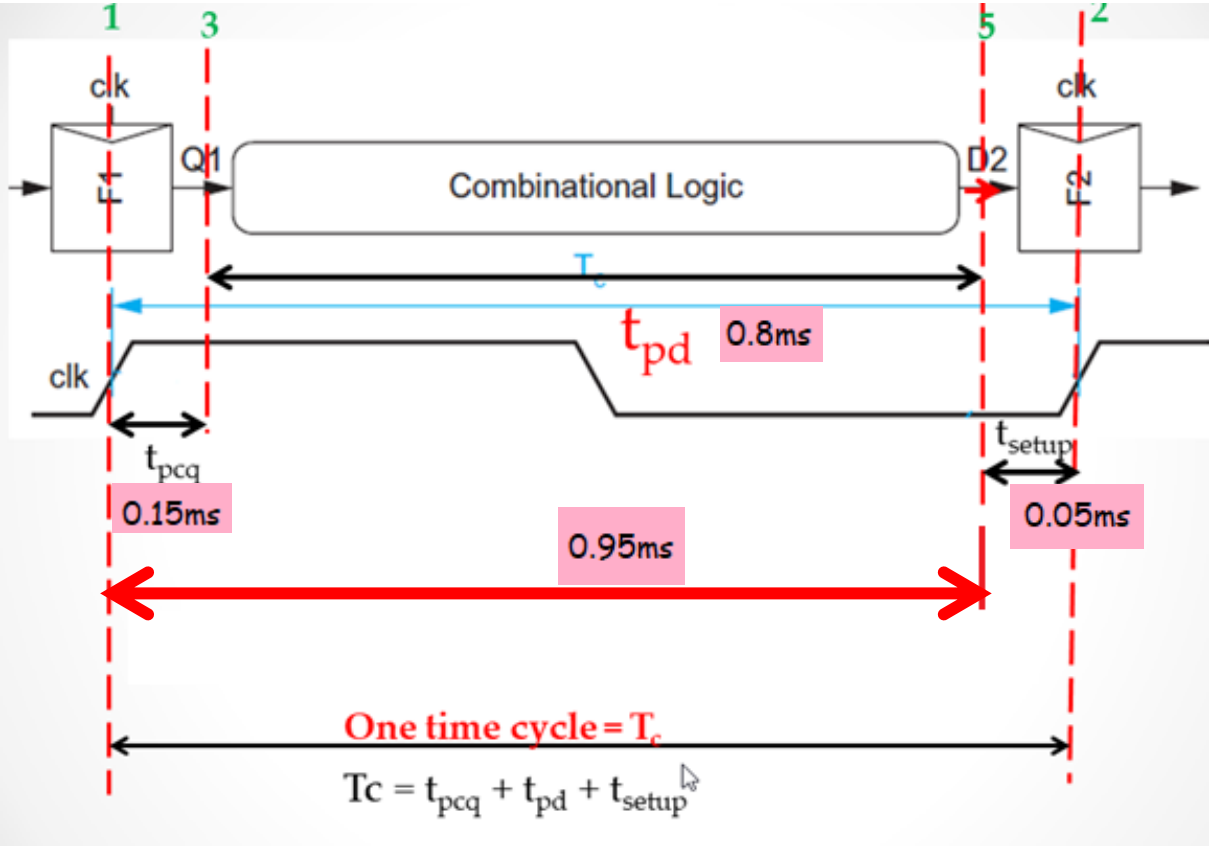
Max-Delay constraints in Flip Flop based method

For example: if Tc = 1ms, tpcq = 0.15ms and tsetup =0.05ms



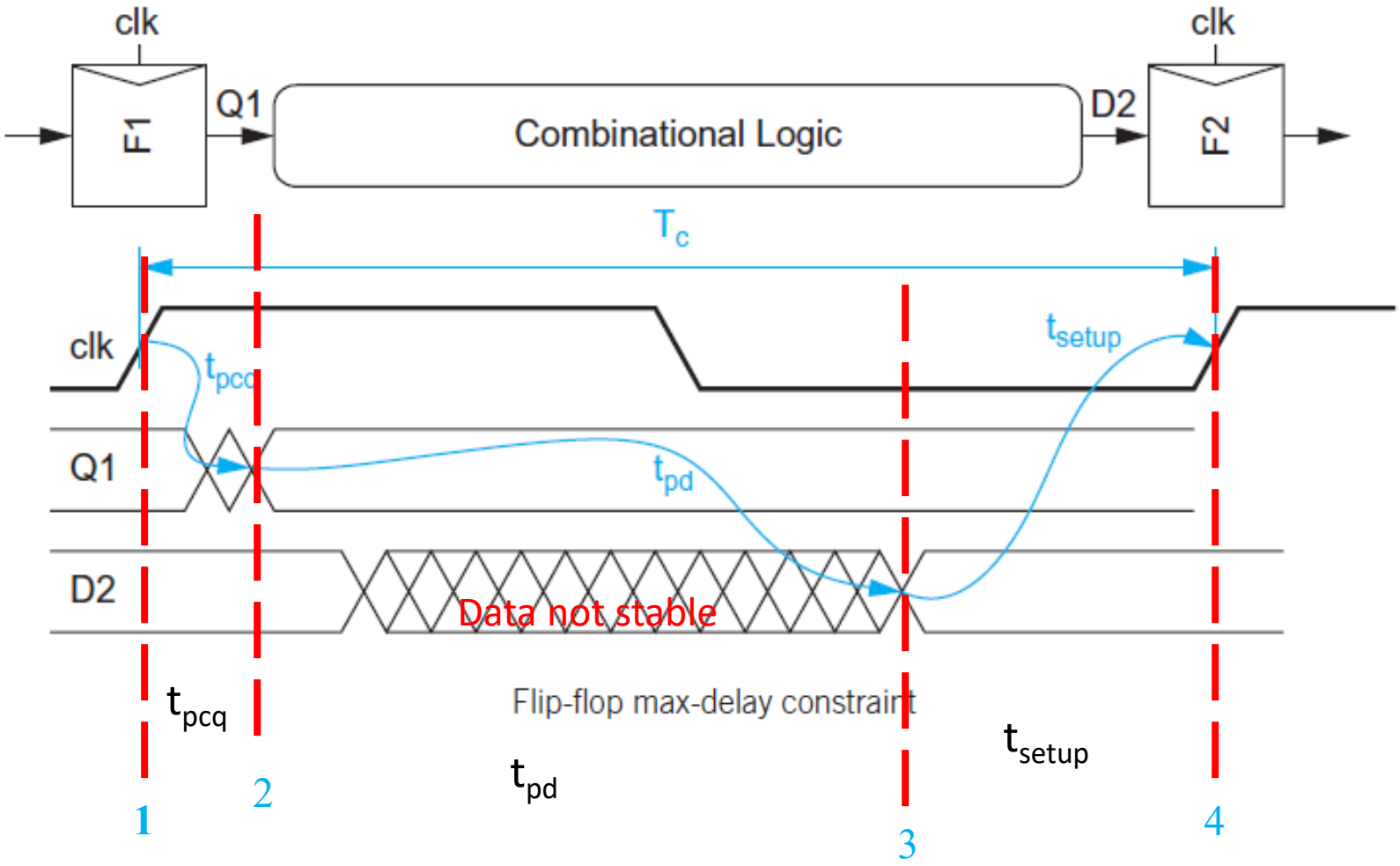
Time available for combinational logic block to compute the entire function is 0.8ms i., tpd = 0.8ms

Max-Delay constraints in Flip Flop based method



Now, output of the comb. Logic is not ready by 0.95ms, the receiving element (F2) will miss its setup time and sample the wrong value. This called setup time failure or max-delay failure.

Max-Delay constraints in Flip Flop based method



Max-Delay constraints in Flip Flop based method

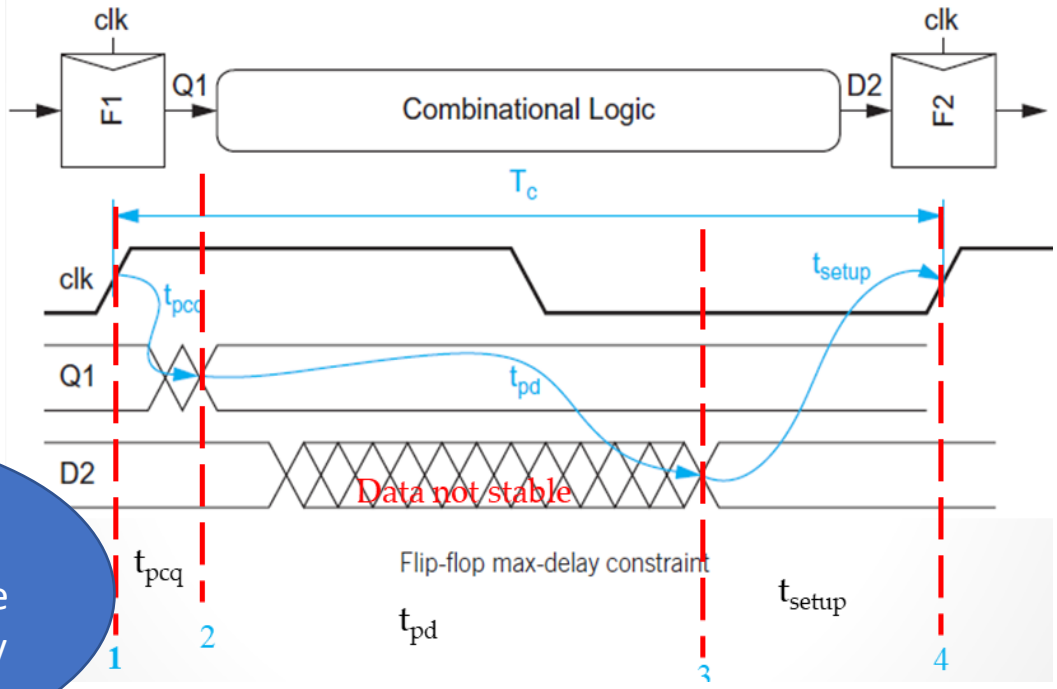
- Thus, the clock period must be at least

$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

- Thus, the maximum allowable logic delay is

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

If this condition is satisfied, there won't be setup time failure or max-delay failure



- **Example 10.1**

For the fig 10.6, the propagation delays and contamination delays of the path are given in Table 10.2. Suppose the registers are built from flip-flops with a setup time of 62ps, hold time of –10 ps, propagation delay of 90ps, and contamination delay of 75ps. Calculate the minimum cycle time *Tc* at which the ALU self-bypass path will operate correctly.

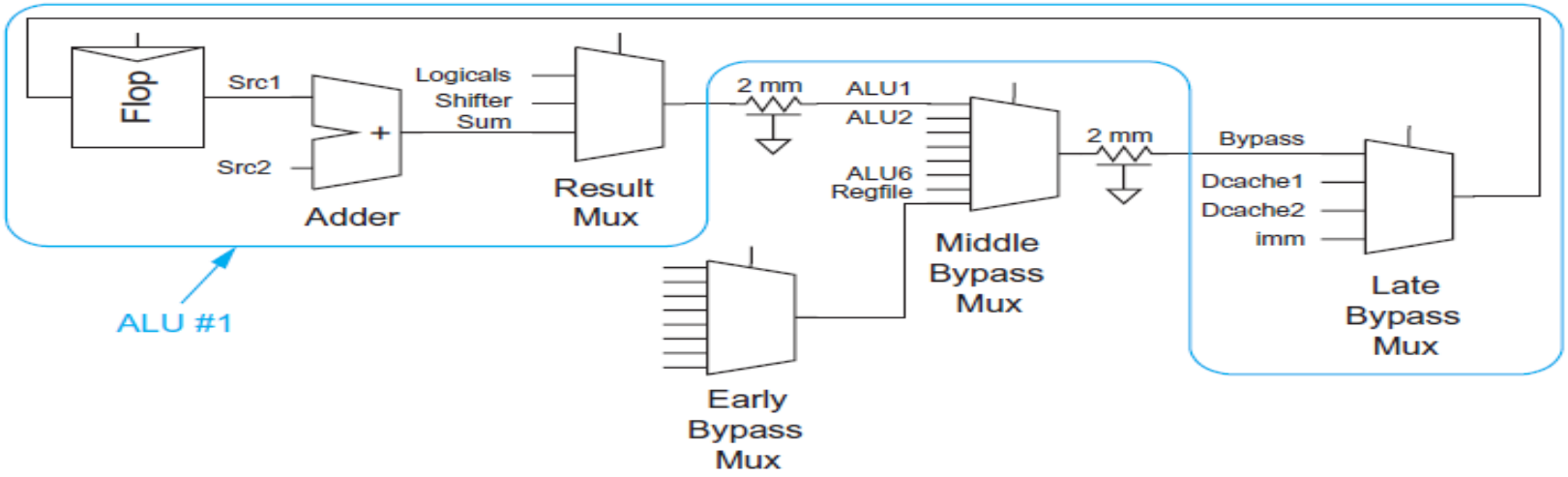
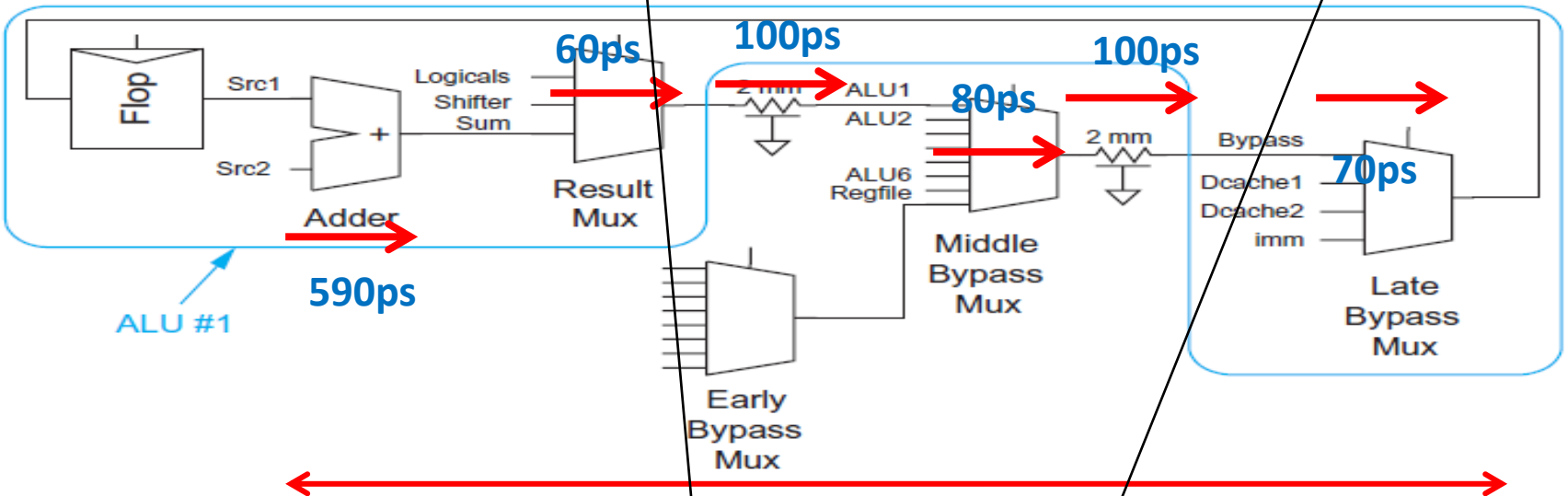


TABLE 10.2 Combinational logic delays

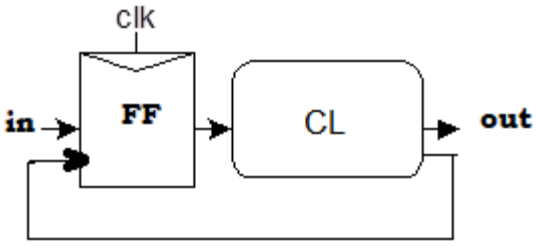
Element	Propagation Delay	Contamination Delay
Adder	590 ps	100 ps
Result Mux	60 ps	35 ps
Early Bypass Mux	110 ps	95 ps
Middle Bypass Mux	80 ps	55 ps
Late Bypass Mux	70 ps	45 ps
2-mm Wire	100 ps	65 ps

- **Example 10.1**

For the fig 10.6, the propagation delays and contamination delays of the path are given in Table 10.2. Suppose the registers are built from flip-flops with a setup time of 62ps, hold time of –10 ps, propagation delay of 90ps, and contamination delay of 75ps. Calculate the minimum cycle time *Tc* at which the ALU self-bypass path will operate correctly.



Total comb. Logic delay = 590+60+100+80+100+70 = 1000ps

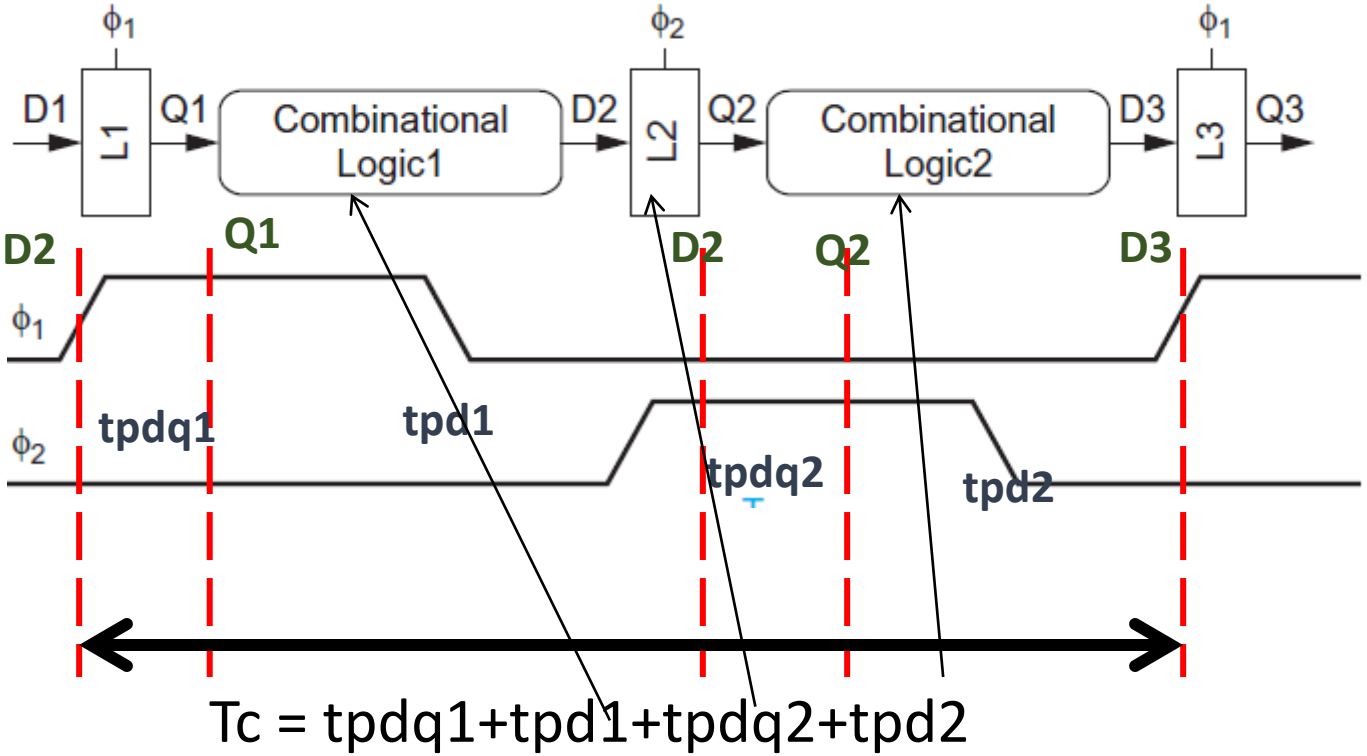


$$T_c \geq t_{pcq} + t_{pd} + t_{setup}$$

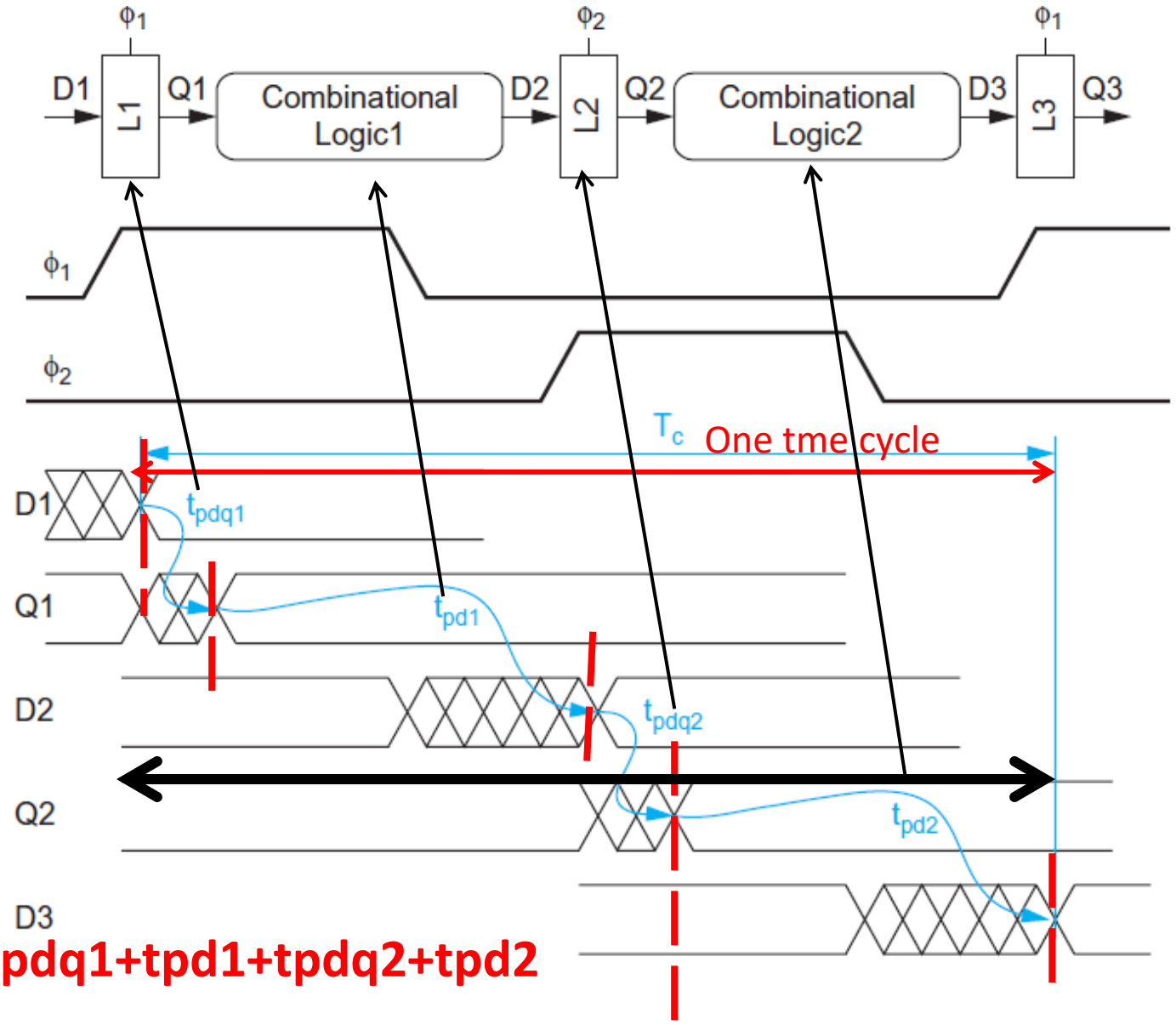
Tc = 90ps+1000ps+62ps = 1152ps

Max-Delay constraints in Two Phase Latch

- Let us assume that data $D1$ arrives at $L1$ while the latch is transparent ($\Phi 1$ high).
- The data propagates through $L1$, the first block of combinational logic, $L2$, and the second block of combinational logic.
- Technically, $D3$ could arrive as late as a setup time before the falling edge of $\Phi 1$ and still be captured correctly by $L3$.



Sequential Circuits Design



Tc = tpdq1+tpd1+tpdq2+tpd2

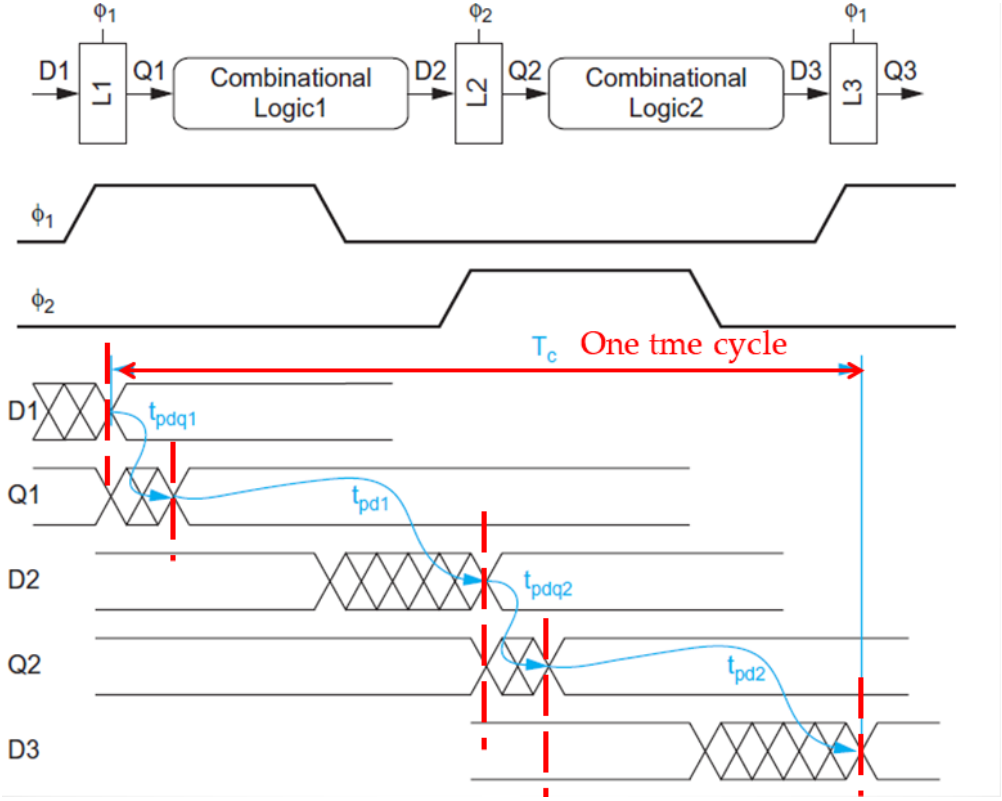
Max-Delay constraints in Two Phase Latch

Thus, the clock period must be at least

$$T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}$$

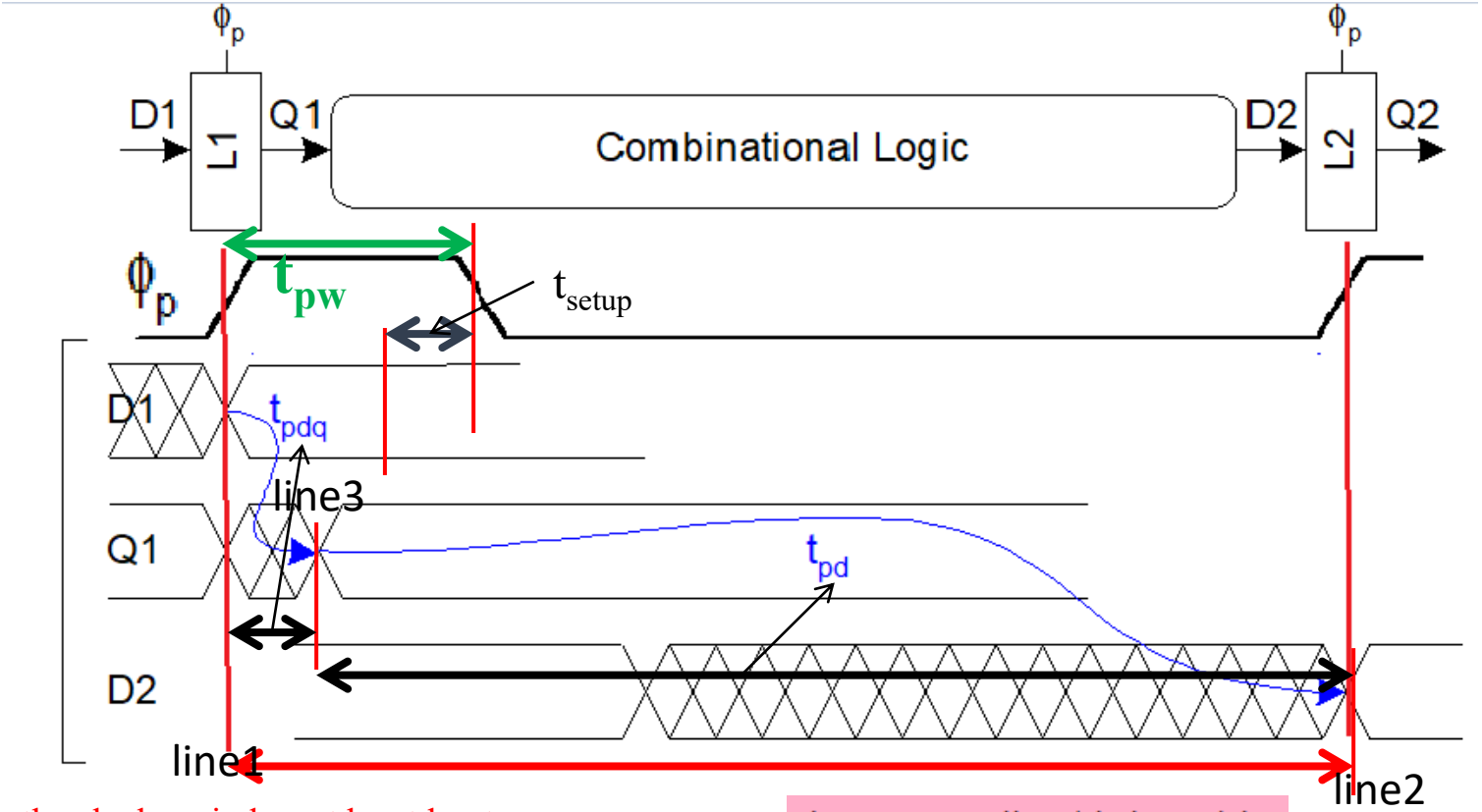
Thus, the maximum allowable logic delay

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



Max-Delay constraints in Phased Latch

Case 1: $T_{pw} > t_{setup}$



the clock period must be at least

$Tc \geq tpdq + tpd$

the maximum allowable logic delay

$Tpd \leq Tc - tpdq$

Max-Delay constraints in Phased Latch

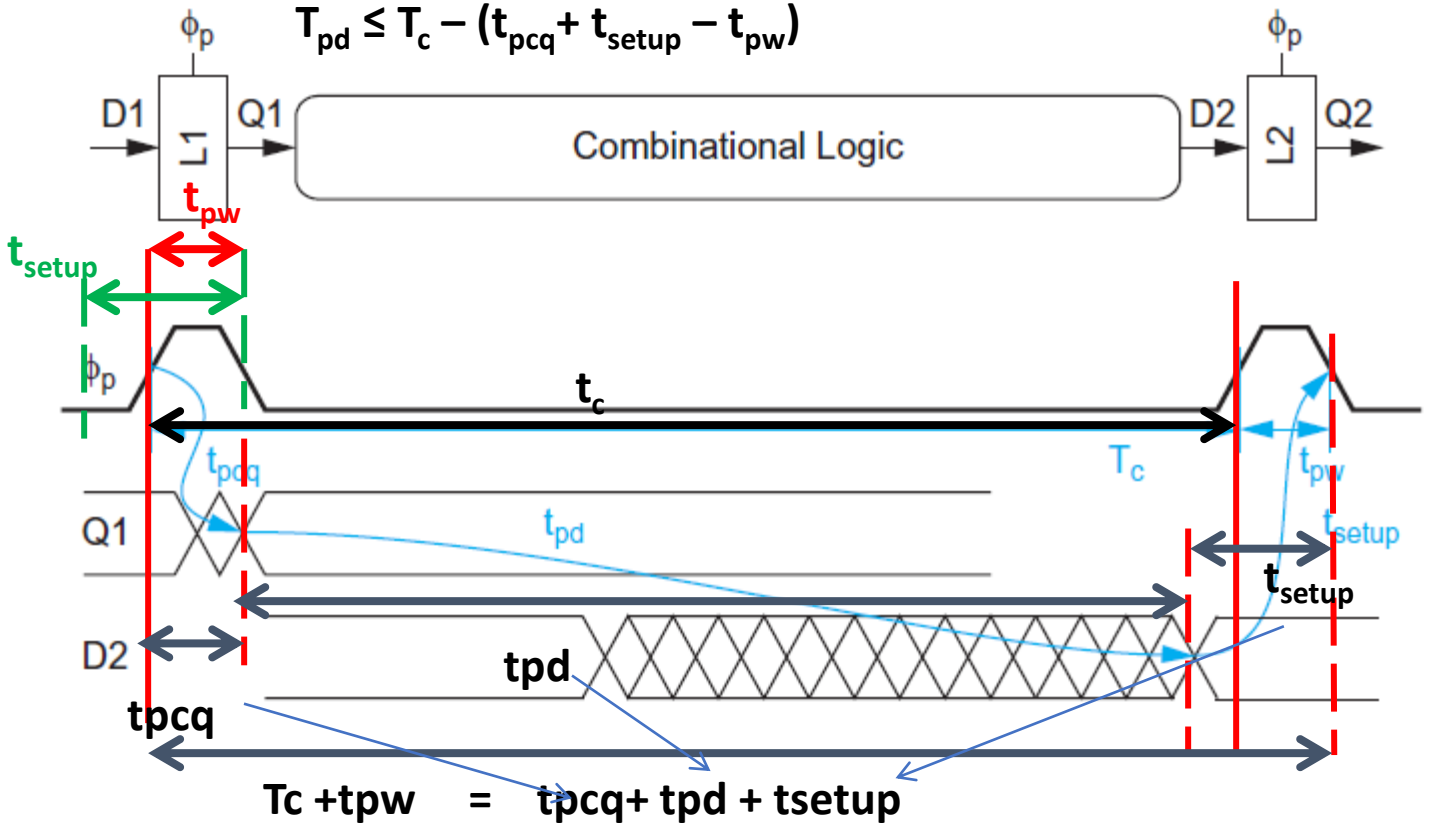
Case 2: $T_{pw} < t_{setup}$

Thus, the clock period must be at least

$$Tc \geq t_{pcq} + t_{pd} + t_{setup} - t_{pw}$$

Thus, the maximum allowable logic delay is without setup time failure

$$T_{pd} \leq T_c - (t_{pcq} + t_{setup} - t_{pw})$$



Problems on Max-Delay

- **Example 10.2**: Recompute the ALU self-bypass path cycle time if the flip-flop is replaced with a pulsed latch. The pulsed latch has a pulse width of 150 ps, a setup time of 40 ps, a hold time of 5 ps, a *clk*-to-*Q* propagation delay of 82 ps and contamination delay of 52 ps, and a *D*-to-*Q* propagation delay of 92 ps.

Solution: The propagation delay of the combinational block is still 1000ps (same as example 10.1)

Since pulsed width of the latch is greater than setup time,

so According to equation $T_c \geq t_{pdq} + t_{pd}$

The cycle time must be at least 92 + 1000 = 1092 ps.

Max-Delay constraints

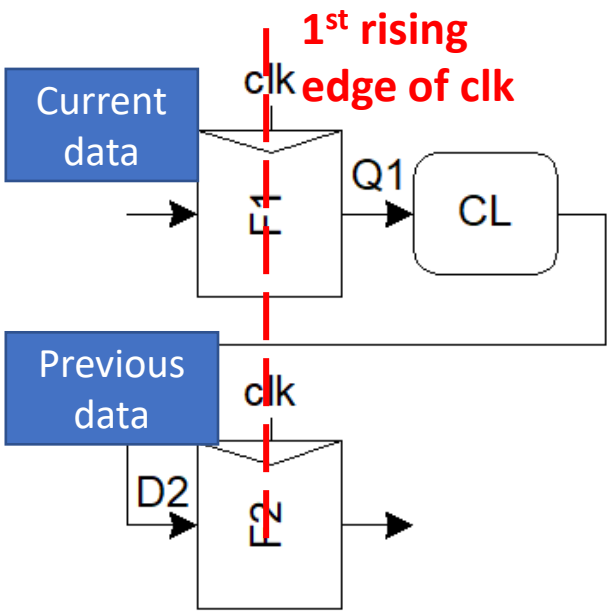
- To avoid Max delay constraints or setup time failure in the design

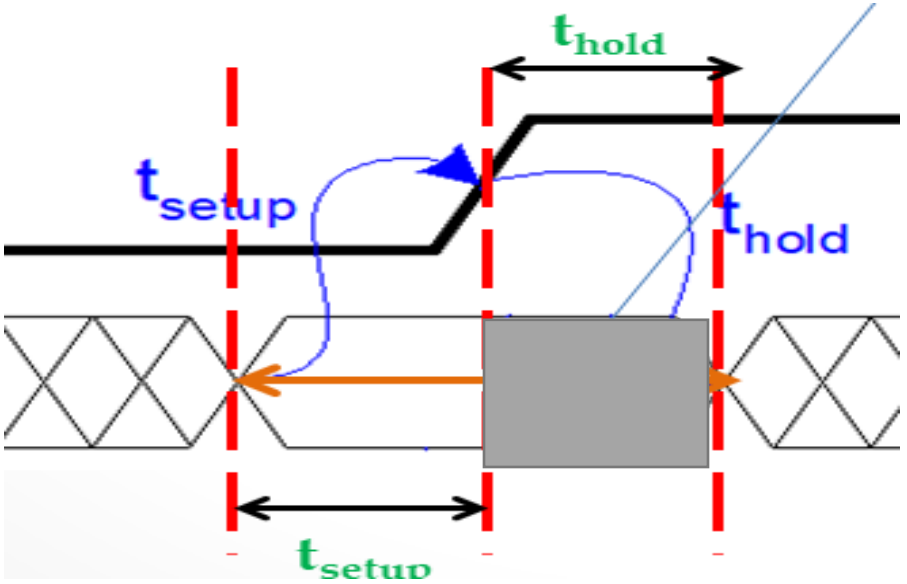
➢**Increase the time cycle**

➢**Decrease the propagation delay of the combinational logic.**

Min-Delay Constraints

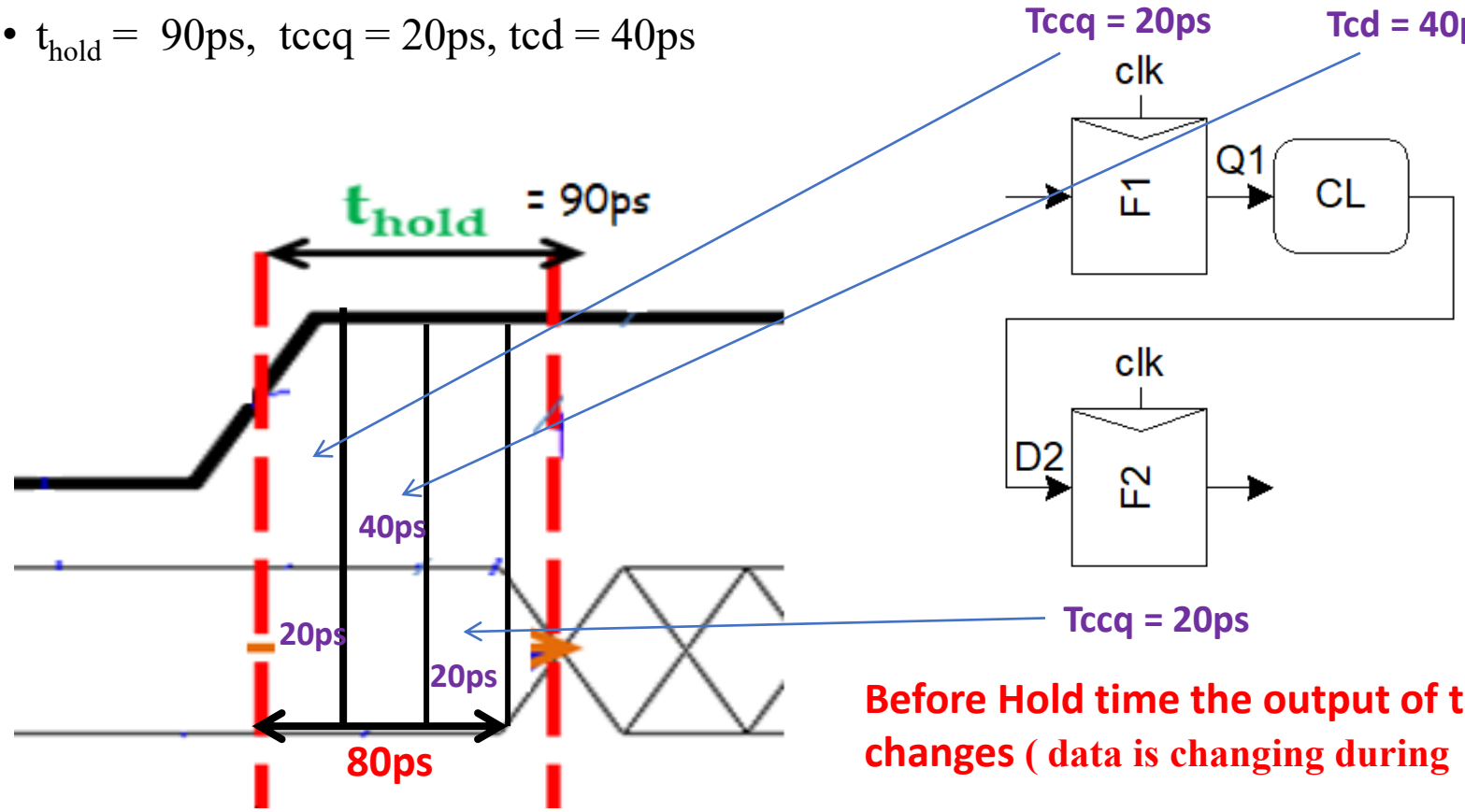
- If the hold time is large and the contamination delay is small, data can incorrectly propagate through two successive elements on one clock edge, corrupting the state of the system. This is called a *race condition*, *hold-time failure*, or *min-delay failure*.





Min-Delay Constraint in Flip Flop

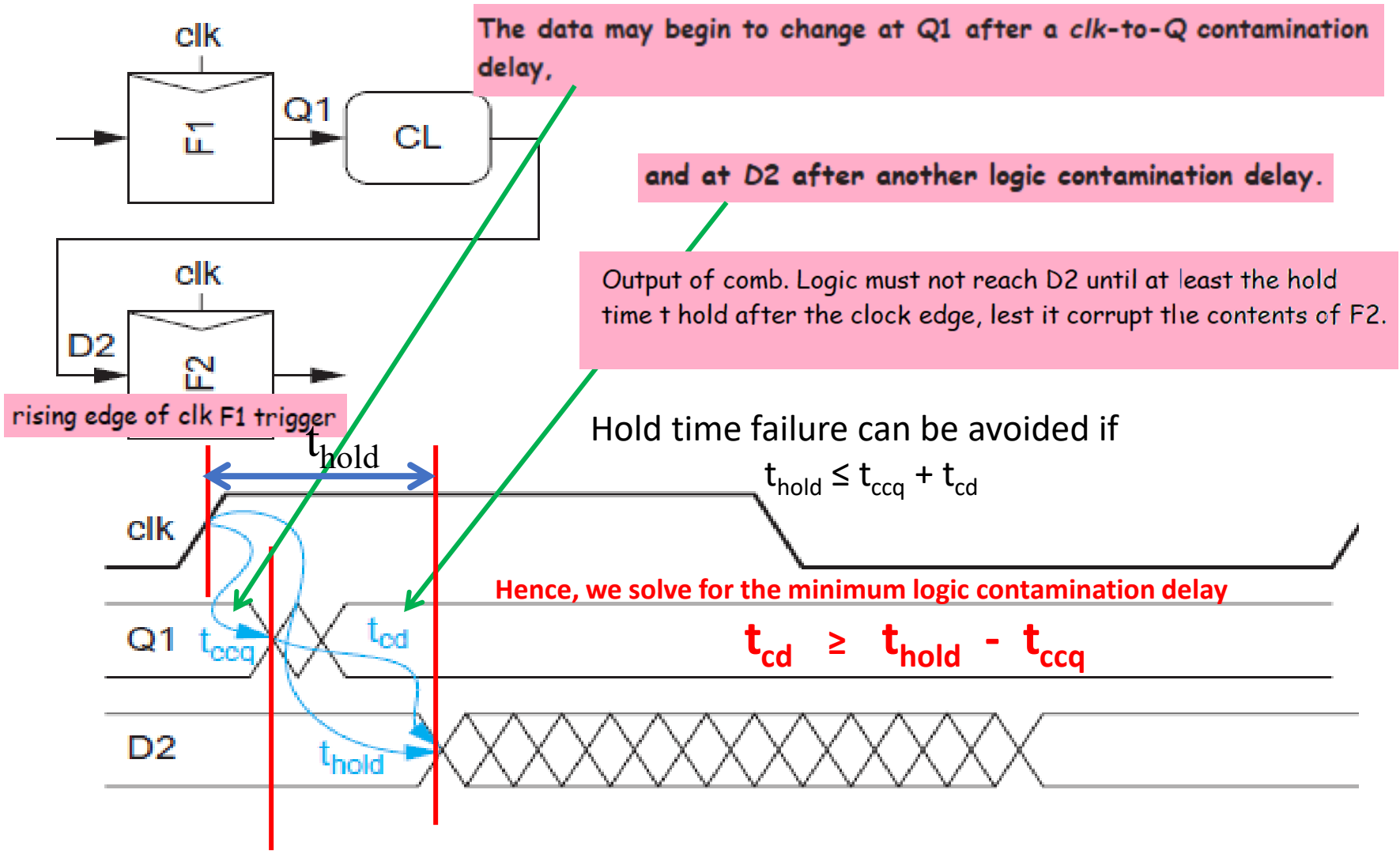
- If the hold time is large and the contamination delay is small, data can incorrectly propagate through two successive elements on one clock edge, corrupting the state of the system. This is called a *race condition*, *hold-time failure*, or *min-delay failure*.
- Assume
- t_{hold} = 90ps, tccq = 20ps, tcd = 40ps



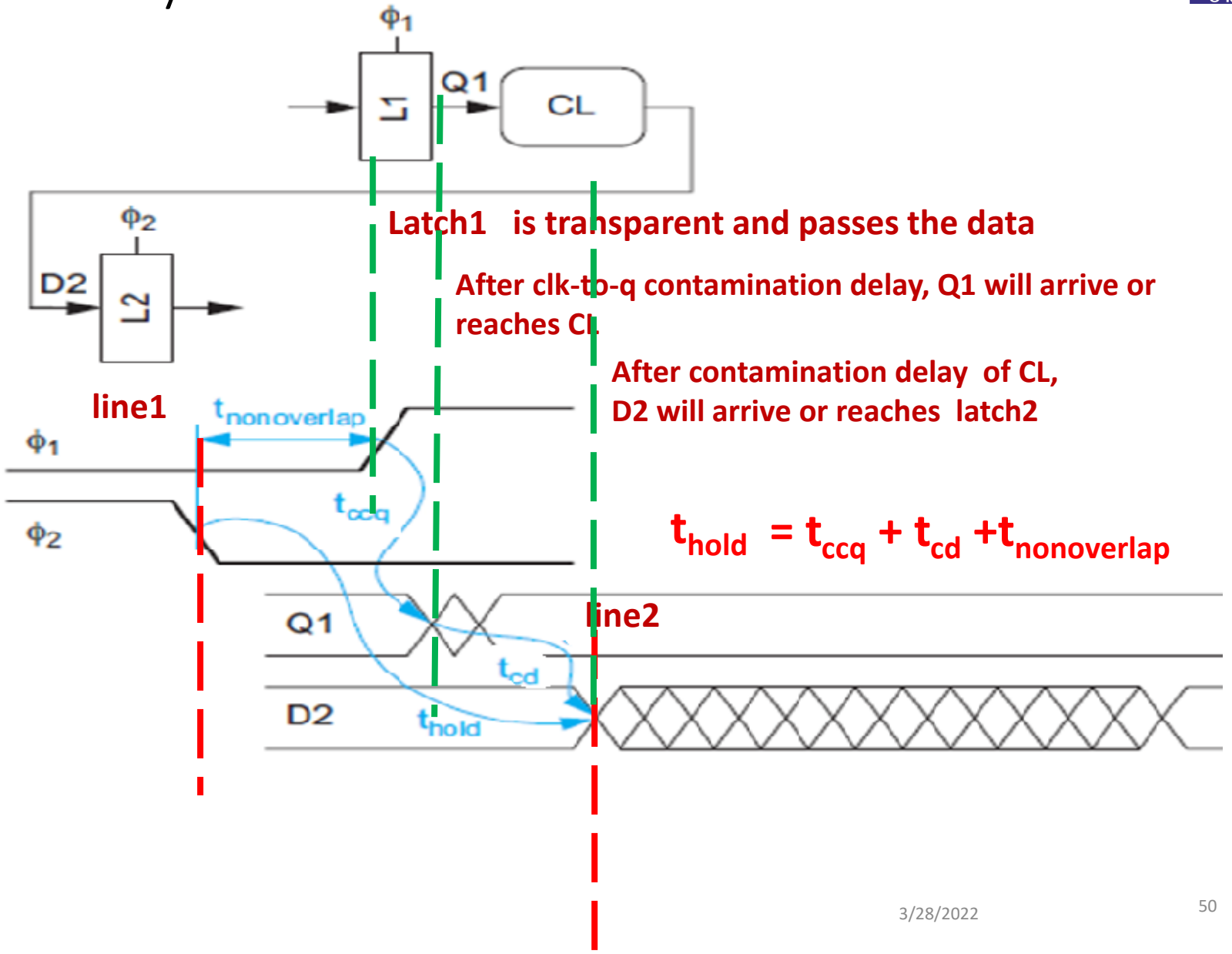
Before Hold time the output of the FF changes (data is changing during hold time)

Min-Delay Constraint in Flip flop

- The path begins with the rising edge of the clock triggering $F1$.



Min-Delay Constraint in Latch

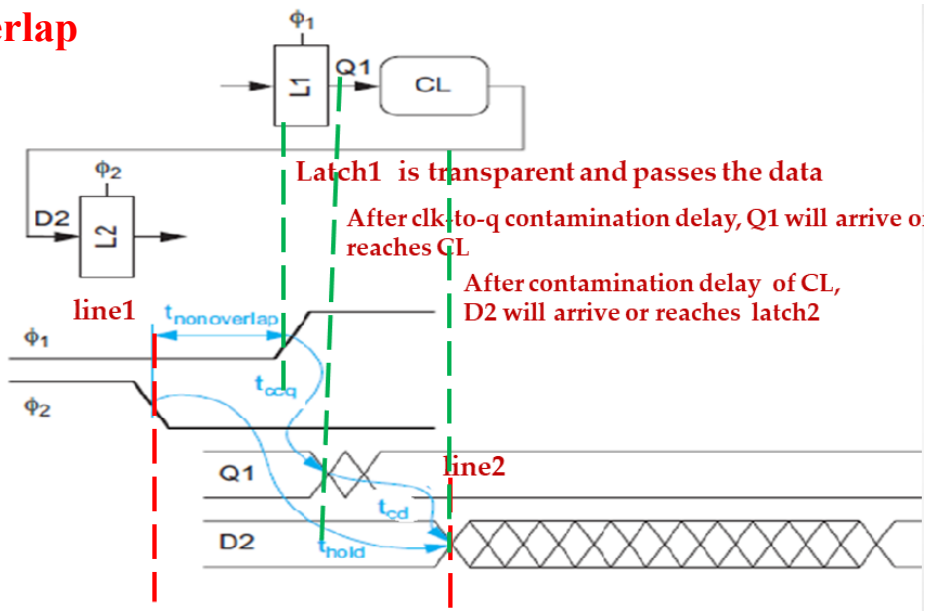


Min-Delay Constraint in Latch

- The path begins with data passing through L1 on the rising edge of Φ1.
- It (Q1) must not reach L2 until a hold time after the previous falling edge of Φ2 because L2 should have become safely opaque before L1 becomes transparent .
- As the edges (falling edge of Φ2 and rising edge of Φ1) separated by $t_{nonoverlap}$, the minimum logic contamination delay through each phase of logic is

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

$$t_{hold} \leq t_{ccq} + t_{cd} + t_{nonoverlap}$$



Min-Delay Constraint in Pulsed Latch

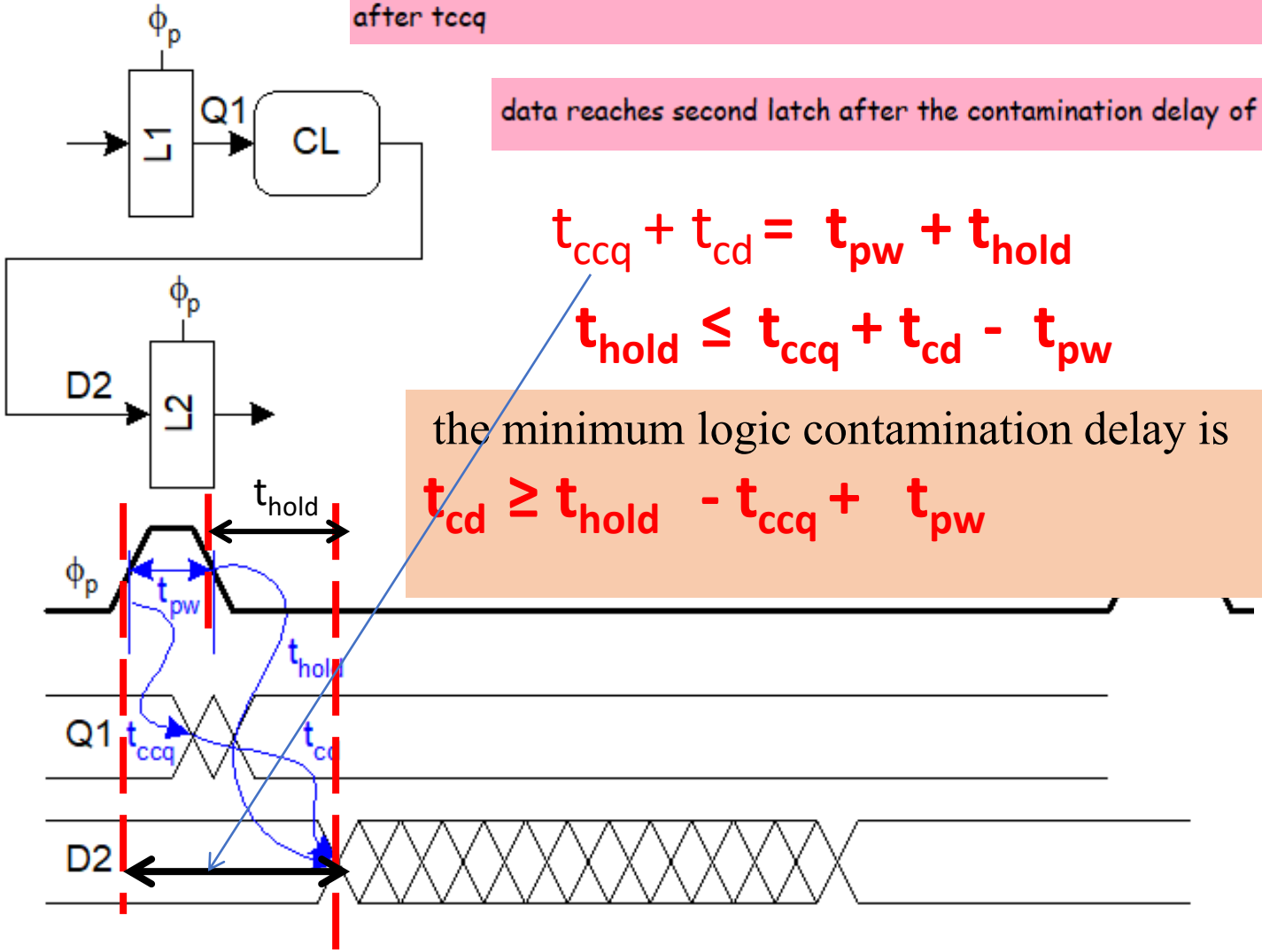
- Case 1: $t_{pw} > t_{setup}$ is similar to latch operation
- Case 2: $t_{pw} < t_{setup}$

Min-Delay Constraint in

- Case 2: $t_{pw} < t_{setup}$

during the rising edge of the pulse latch becomes transparent and data reach CL after tccq

data reaches second latch after the contamination delay of CL (tcd)



$$t_{ccq} + t_{cd} = t_{pw} + t_{hold}$$

$$t_{hold} \leq t_{ccq} + t_{cd} - t_{pw}$$

the minimum logic contamination delay is

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

Min-Delay constraints

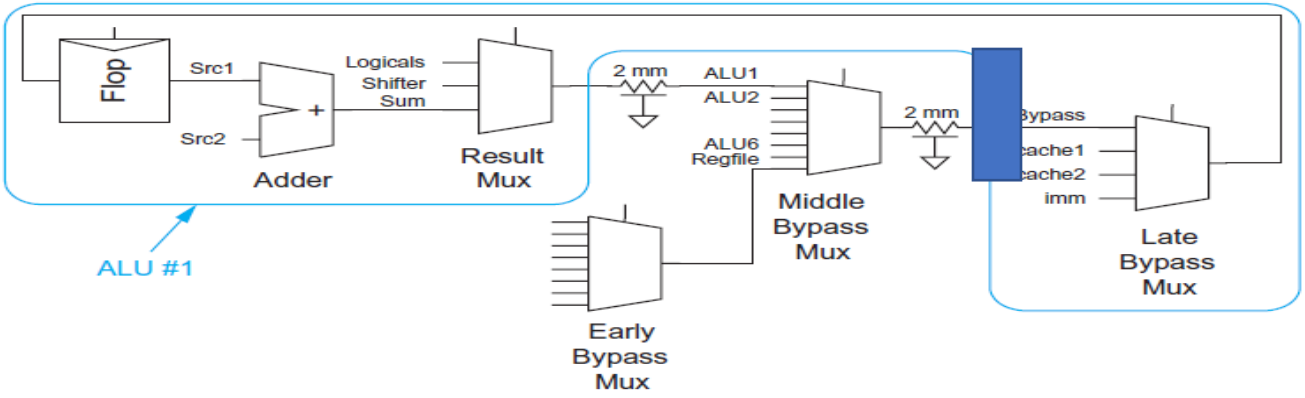
- To avoid Min delay constraints or hold time failure in the design

➢**Increase the contamination delay of the combinational logic.**

➢**That is hold time must be less than the contamination delay**

Problems on Min-Delay

- **Example 10.3**: In the ALU self-bypass example with flip-flops from Figure 10.6, the earliest input to the late bypass multiplexer is the *imm* value coming from another flip-flop. Will this path experience any hold-time failures?



flip-flops with a setup time of 62ps, hold time of –10 ps, delay of 90ps, and contamination delay of 75ps

TABLE 10.2 Combinational logic delays

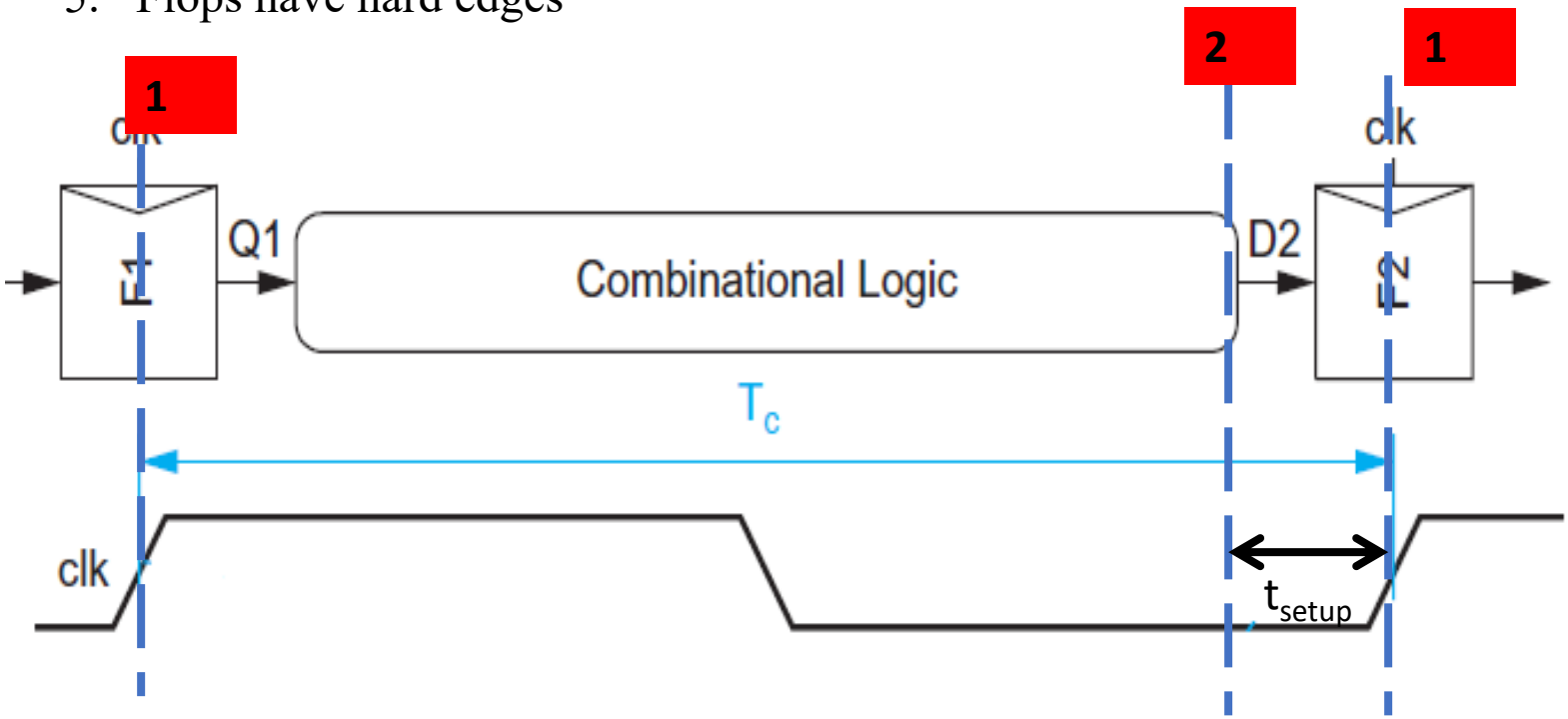
Element	Propagation Delay	Contamination Delay
Adder	590 ps	100 ps
Result Mux	60 ps	35 ps
Early Bypass Mux	110 ps	95 ps
Middle Bypass Mux	80 ps	55 ps
Late Bypass Mux	70 ps	45 ps
2-mm Wire	100 ps	65 ps

The late bypass mux has the contamination delay, *tcd* = 45 ps.
The flip-flops have *t*hold = –10 ps and *tccq* = 75 ps
Hence, equation $t_{cd} \geq t_{\text{hold}} - t_{ccq}$

is easily satisfied

Time Borrowing

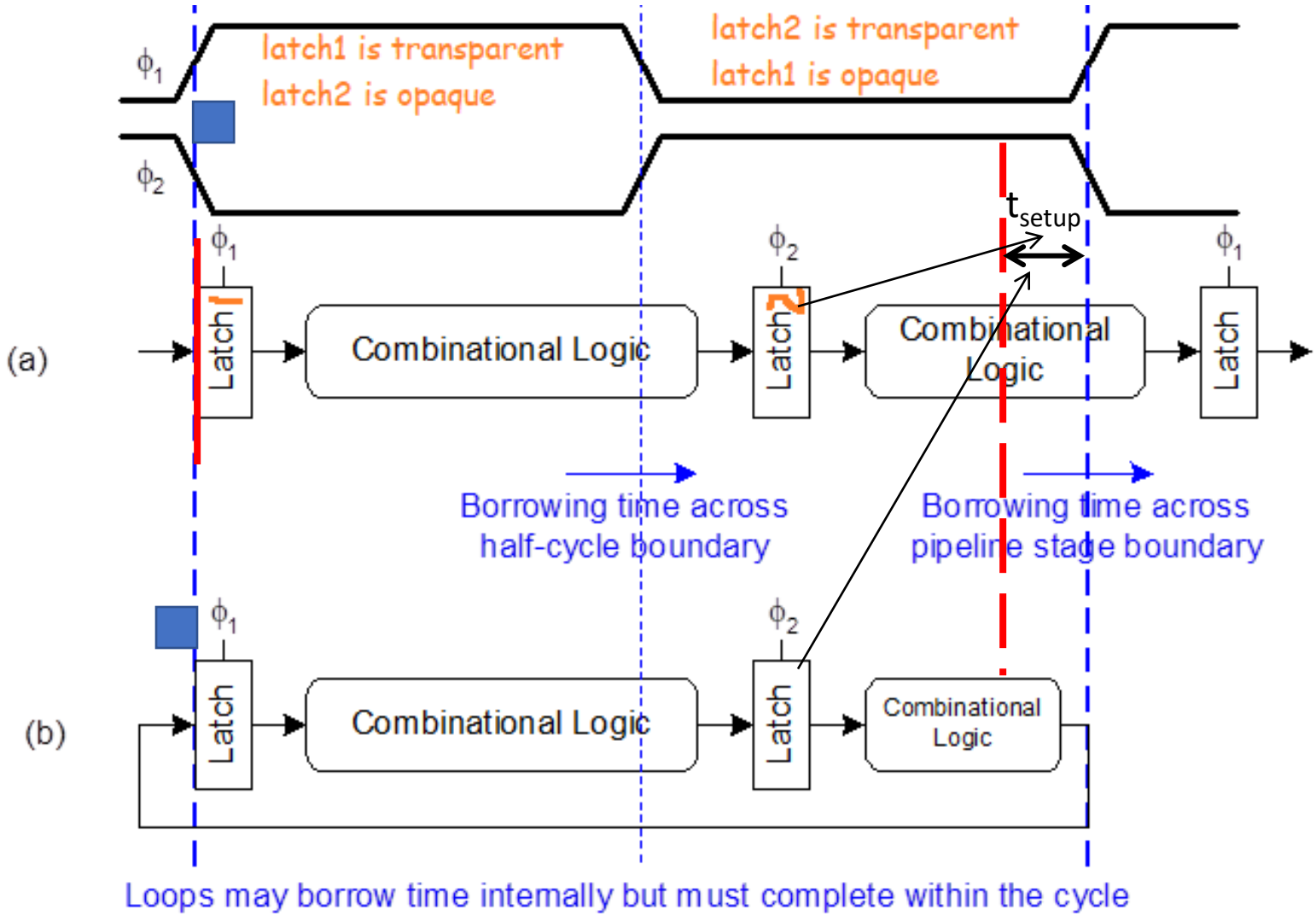
- In a flip flop-based system:
  1. Data launches on one rising edge
  2. Must setup before next rising edge
  3. If it arrives late, system fails
  4. If it arrives early, time is wasted
  5. Flops have hard edges



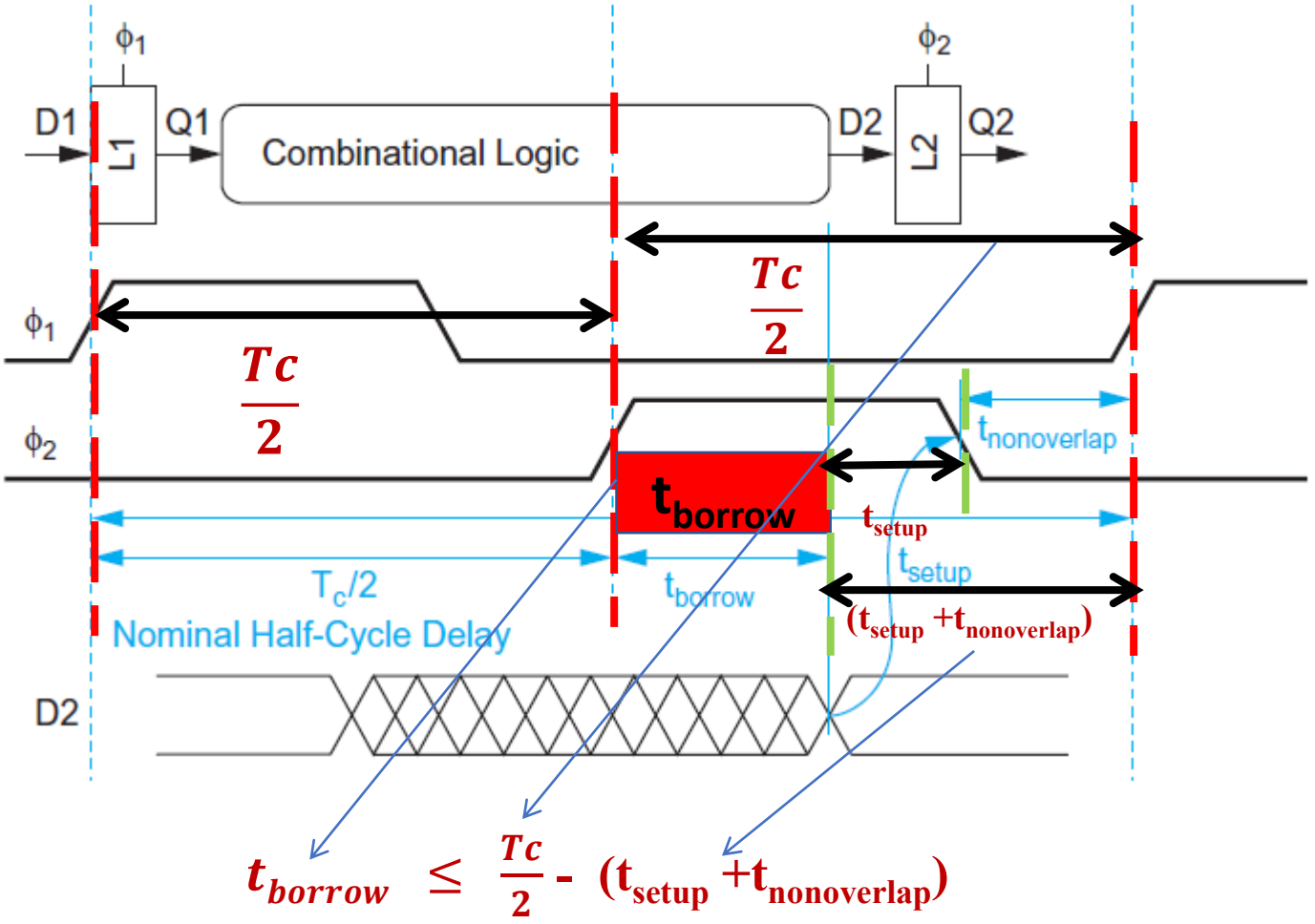
Time Borrowing

- 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



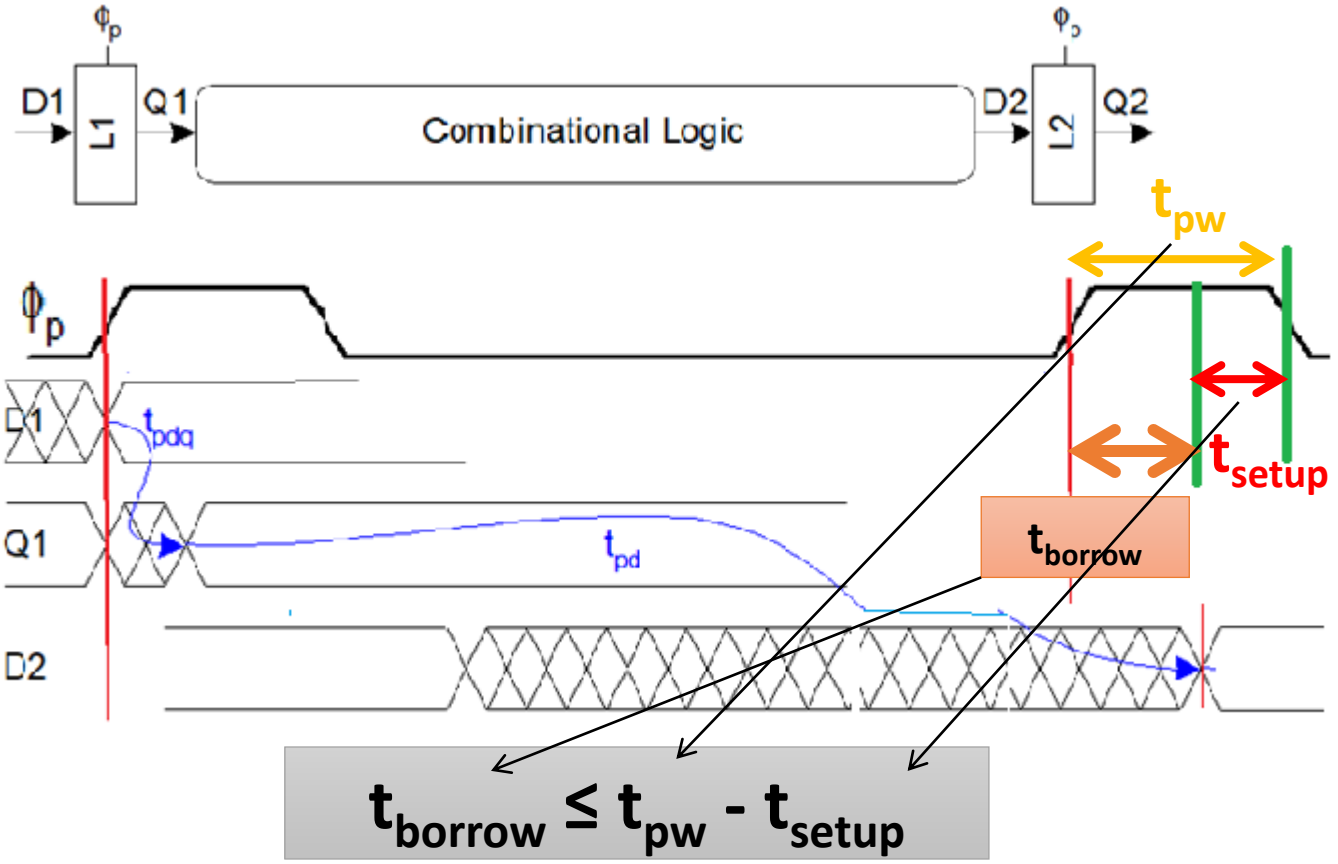
How Much time can be borrowed in two phase latch method ?



How Much time can be borrowed in Pulsed Latch method ?

- **Time borrowing is applicable only for the case where**

$t_{pw} > t_{setup}$



Problems on Time Borrowing

- **Example 10.5**: Suppose the ALU self-bypass path is modified to use two-phase transparent latches. A mid-cycle Φ2 latch is placed after the adder, as shown in Figure 10.14. The latches have a setup time of 40 ps, a hold time of 5 ps, a *clk*-to-*Q* propagation delay of 82 ps and contamination delay of 52 ps, and a *D*-to-*Q* propagation delay of 82 ps.
- Compute the minimum cycle time for the path. How much time is borrowed through the mid-cycle latch at this cycle time? If the cycle time is increased to 2000 ps, how much time is borrowed?

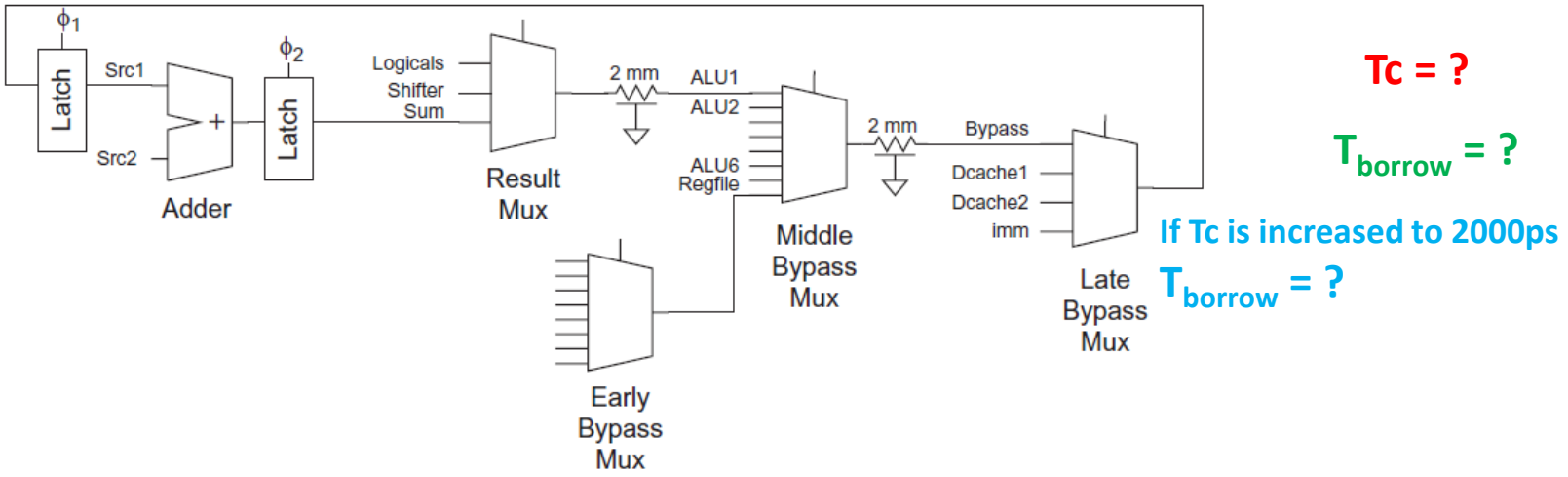
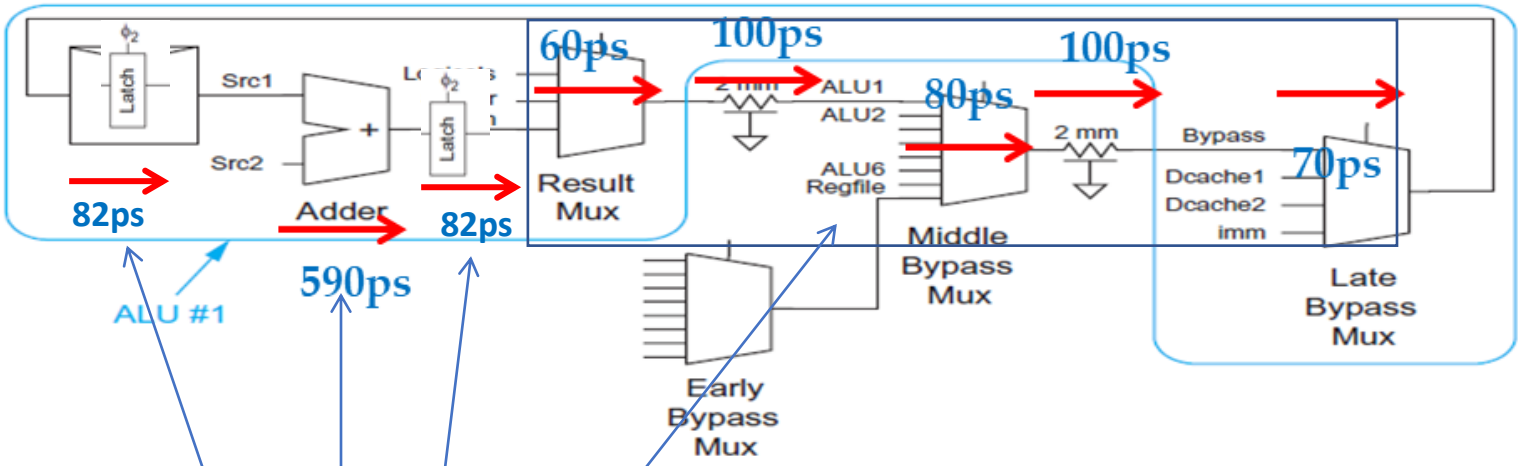


FIGURE 10.14 ALU self-bypass path with two-phase latches

Problems on Time Borrowing

- **Solution:** According to $T_c \geq t_{pdq1} + t_{pd1} + t_{pdq2} + t_{pd2}$



The latches have a setup time of 40 ps, a hold time of 5 ps, a *clk*-to-*Q* propagation delay of 82 ps and contamination delay of 52 ps, and a *D*-to-*Q* propagation delay of 82 ps.

- The cycle time is *Tc* = 82 + 590 + 82 + 410 = 1164 ps.
- The first half of the cycle involves the latch and adder delays and consumes 82 + 590 = 672ps.
- The nominal half-cycle time is *Tc* /2 = 1164 / 2 = 582 ps.

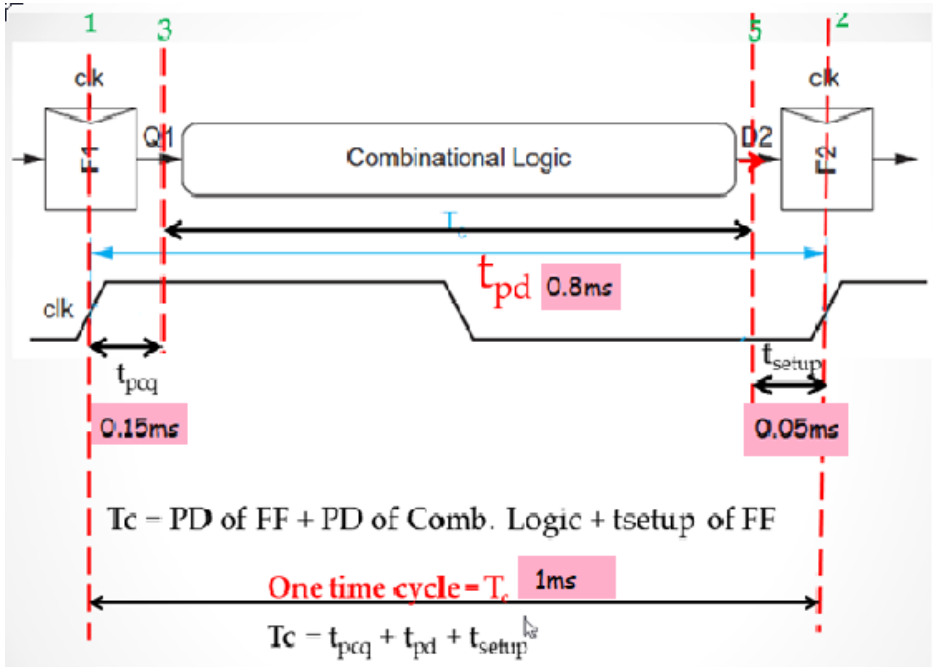
  Hence, the path borrows 672 – 582 = 90 ps from the second half-cycle.
- If the cycle time increases to 2000 ps and the nominal halfcycle time becomes 1000 ps, time borrowing no longer occurs.

Clock Skew

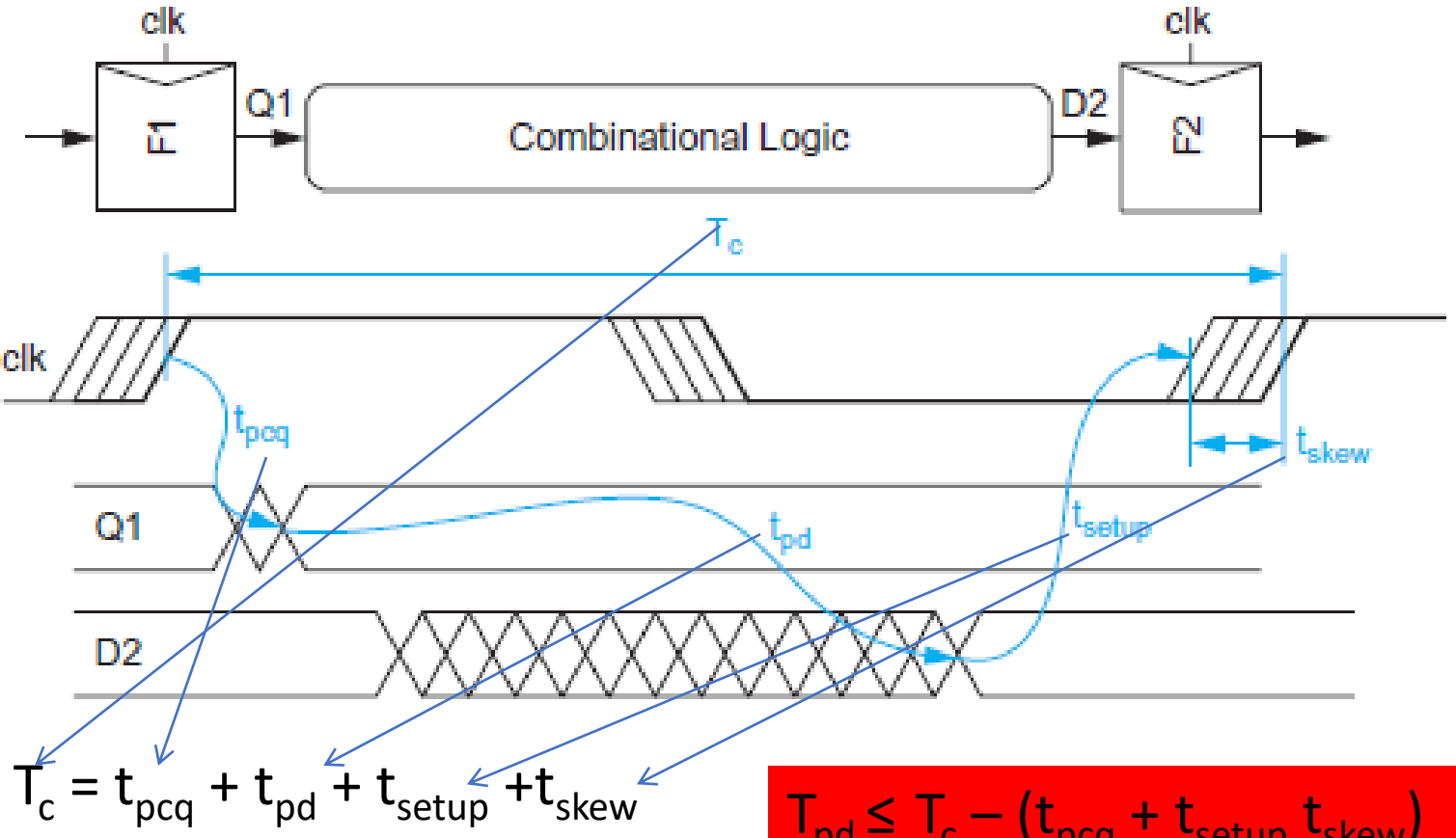
- We have assumed zero clock skew till now
- But Clocks really have uncertainty in arrival time, which will
  - Decreases maximum propagation delay (means decreases the propagation delay of the combinational logic)
  - Increases minimum contamination delay (means increases the contamination delay of the combinational logic)
  - Decreases time borrowing

Clock skew in FF based system

- Assume clk arrives at 0.9ms instead of 1ms
- But the sequencing overheads (i.e., tpcq and tsetup) will not change
- So if clk arrives at 0.9ms, then data from the combinational circuit must be ready at 0.85ms ($T_c - t_{setup}$)
- But according to the example data from CL arrives at 0.95ms.
- The only solution is to decrease the propagation delay of the CL to 0.7ms instead of 0.8ms



Clock skew in FF based system

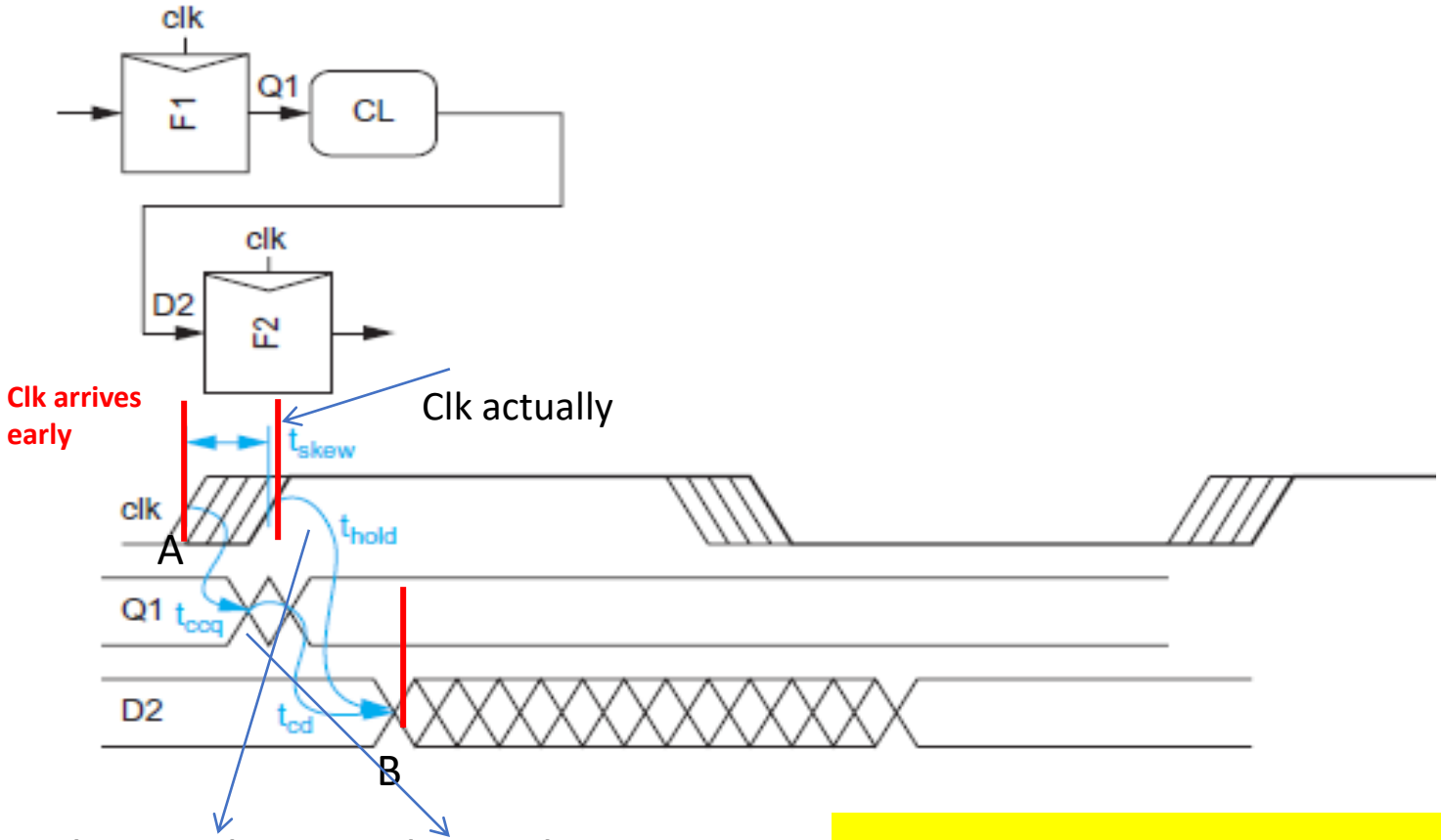


$T_c = t_{pcq} + t_{pd} + t_{setup} + t_{skew}$

$T_{pd} \leq T_c - (t_{pcq} + t_{setup}\ t_{skew})$

- **Sequencing overheads(FF delay components) cannot be changed hence in order to avoid setup time failure the propagation delay of the logic block has to be reduced**

Clock skew in FF based system



$t_{hold} + t_{skew} = t_{ccq} + t_{cd}$

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

- **Sequencing overheads(FF delay components) cannot be changed hence in order to avoid hold time failure the contamination delay of the logic block has to be increased**

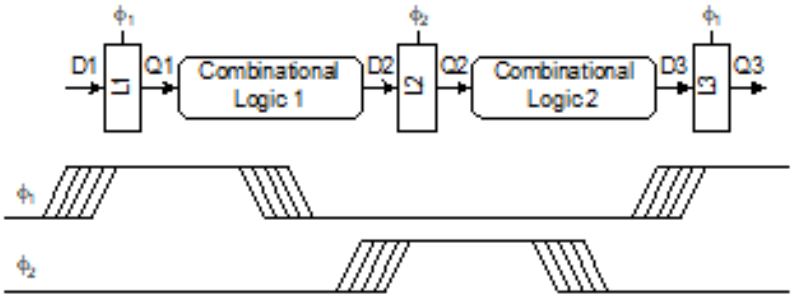
Clock skew in latches

2-Phase Latches

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

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

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



Pulsed Latches

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

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

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

THANK YOU

Rekha S S
Department of Electronics and Communication
rekha.ss@pes.edu