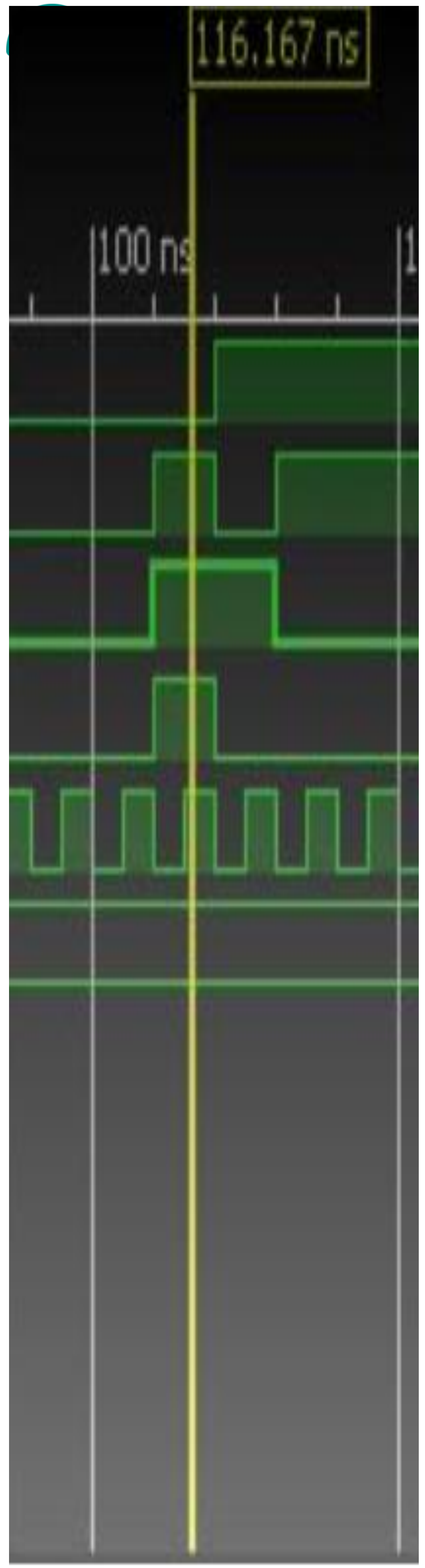
Sequential logic I

MC. Martin González Pérez

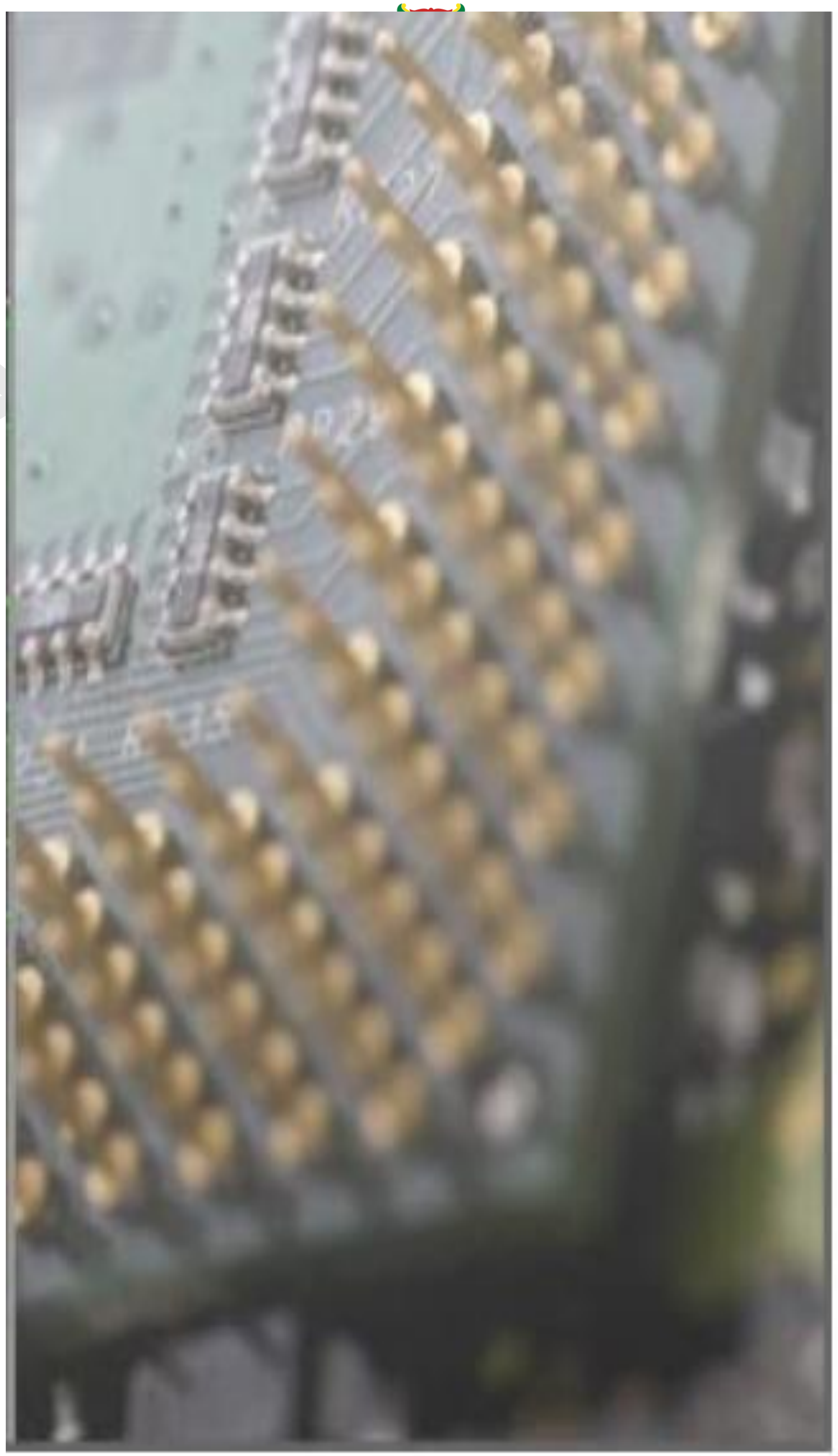
DIGITAL DESIGN

Agenda

- Sequential logic
- Latch
- Flip-Flop
- Synchronous circuits
- Timing



Sequential logic

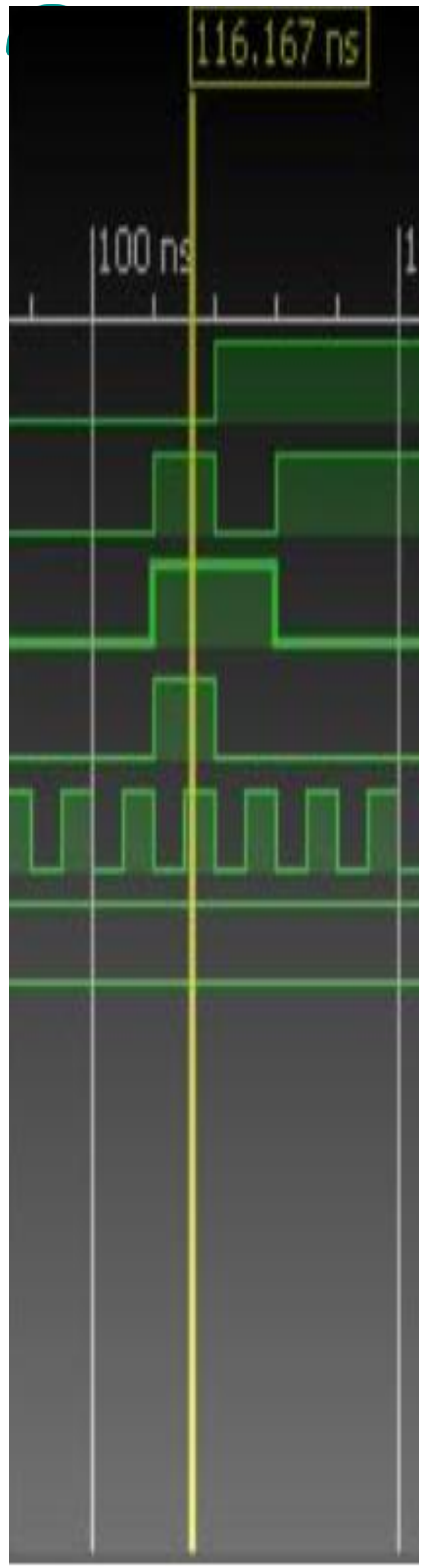


Sequential logic circuits

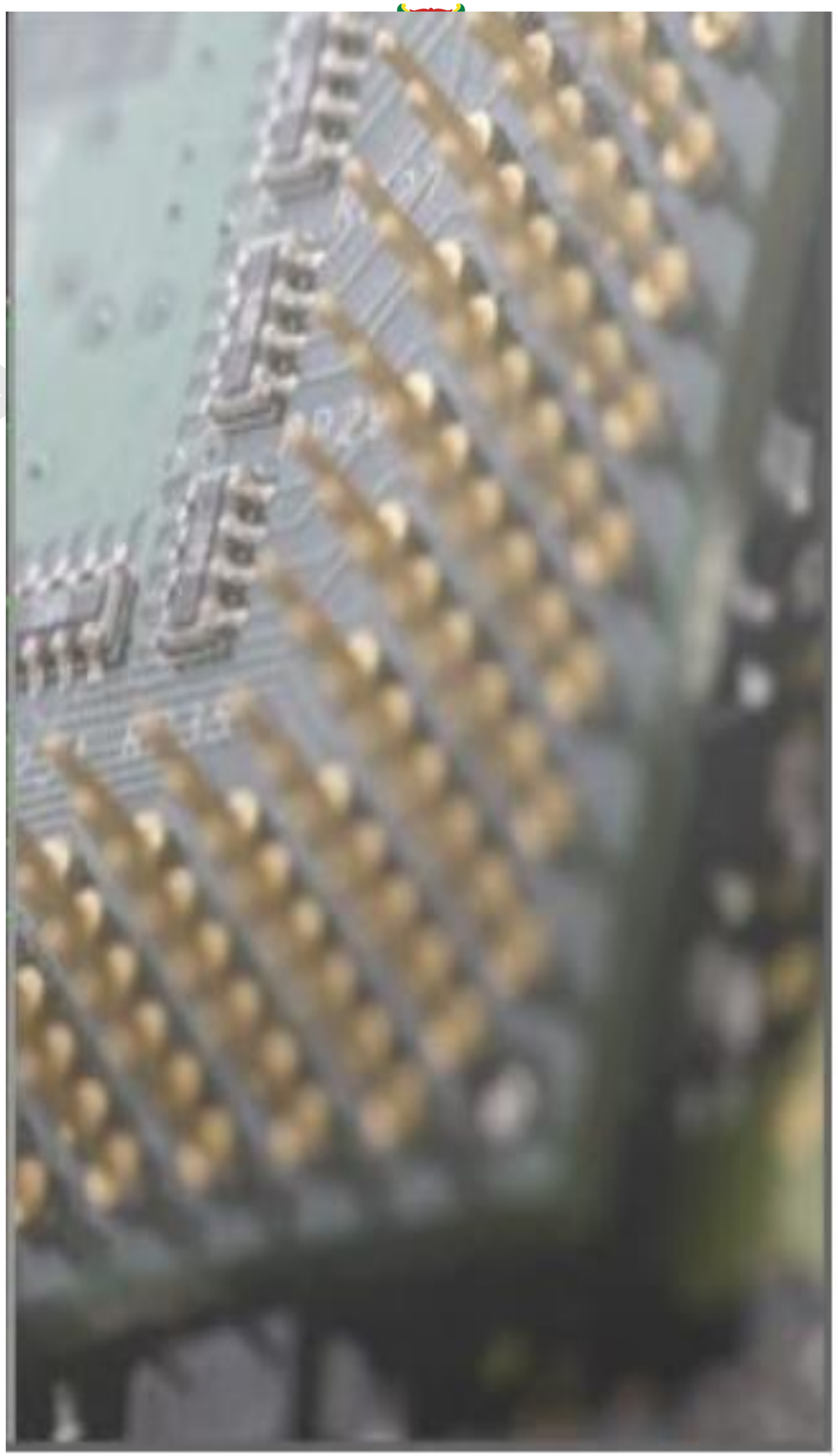
Outputs of sequential logic depend on current and prior input values, **it has memory**.

The **state** of a digital sequential circuit is a set of bits called state variables that contain all the information about the past necessary to explain the future behavior of the circuit, therefore, the behavior of the circuit depends on the current inputs and its current state.

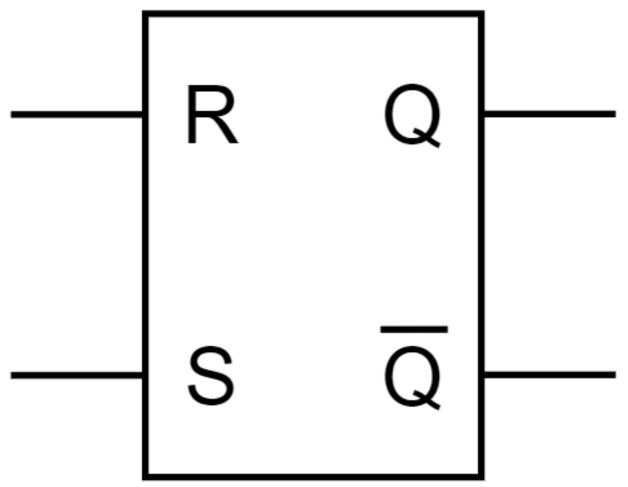
Latches and **flip-flops**, are simple sequential circuits that store one bit of the state.



Latch



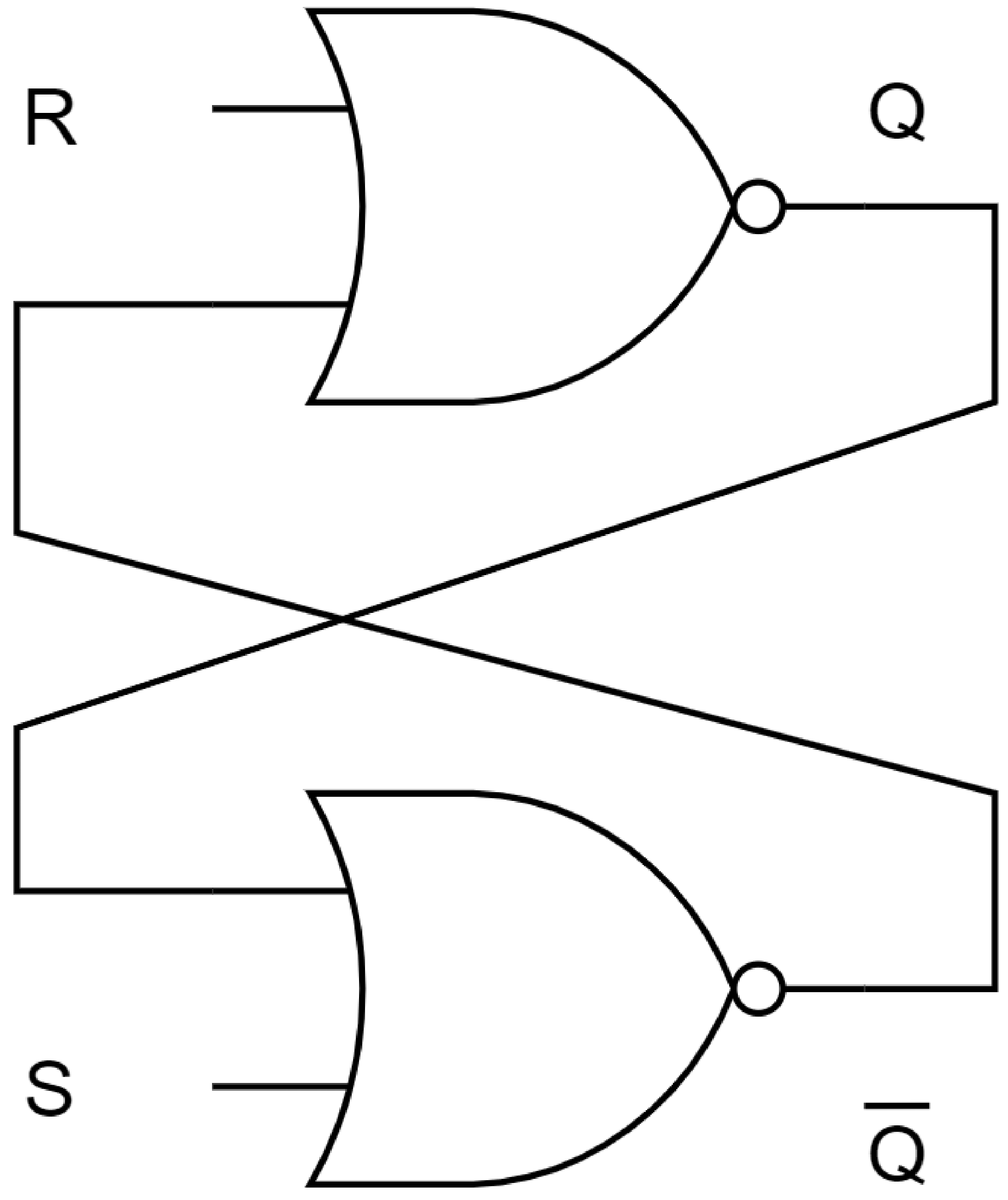
SR Latch



SR latch symbol

One of the simplest sequential circuits is the SR latch.

S	R	Q	$\bar{Q}$
0	0	Q	$\bar{Q}$
0	1	0	1
1	0	1	0
1	1	-	-

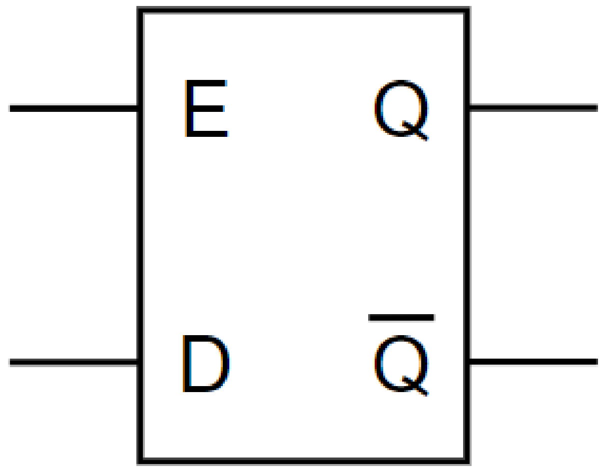


S=1, R=0 Set S=0, R=1 Reset

S=0, R=0 retains the previous value (it has memory!!).

S=1, R=1 is invalid and should be avoided.

D Latch



D latch symbol

SR latch behaves is undefined when S and R are asserted. **D latch** solves this problem.

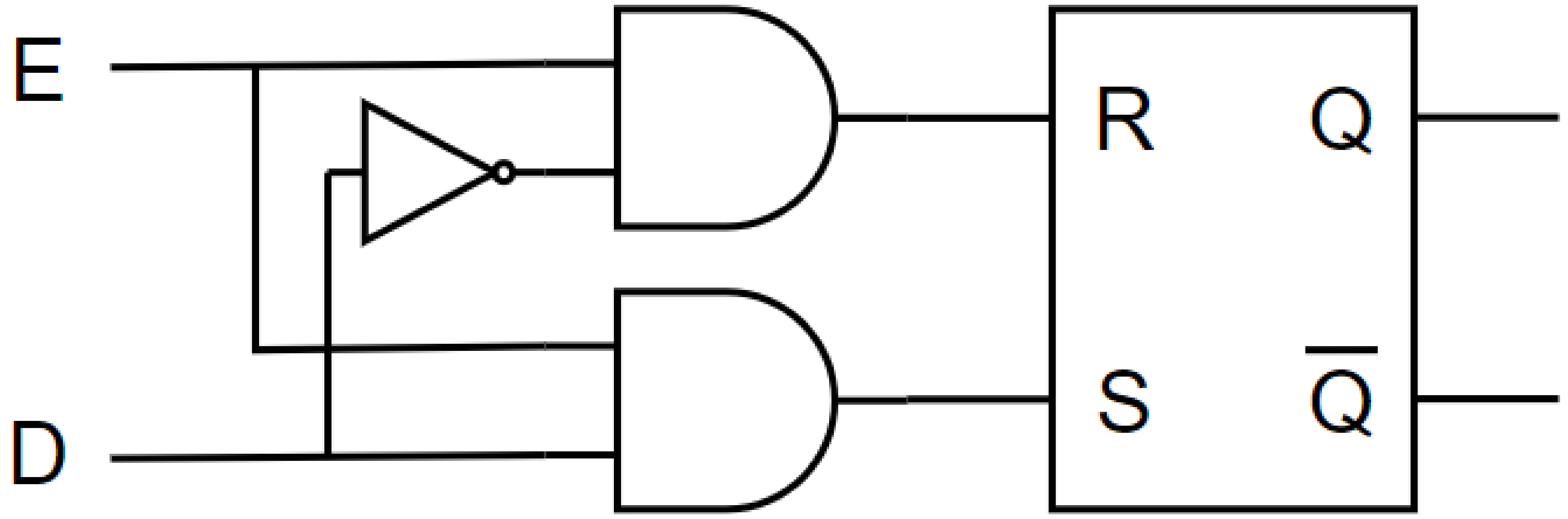
E: control when the output changes

D: data input.

When **E = 1**, D passes through to Q.

When **E = 0**, holds its previous value.

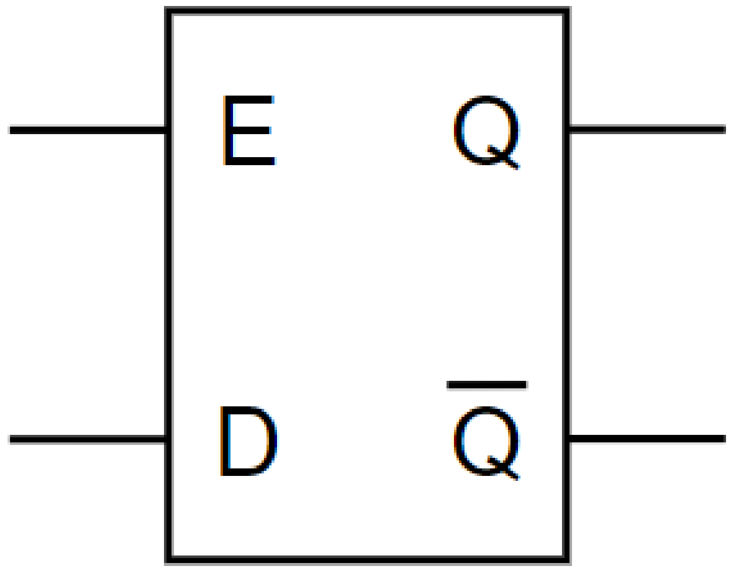
E	D	S	R	Q	$\overline{Q}$
0	X	0	0	Q	$\overline{Q}$
1	0	0	1	0	1
1	1	1	0	1	0

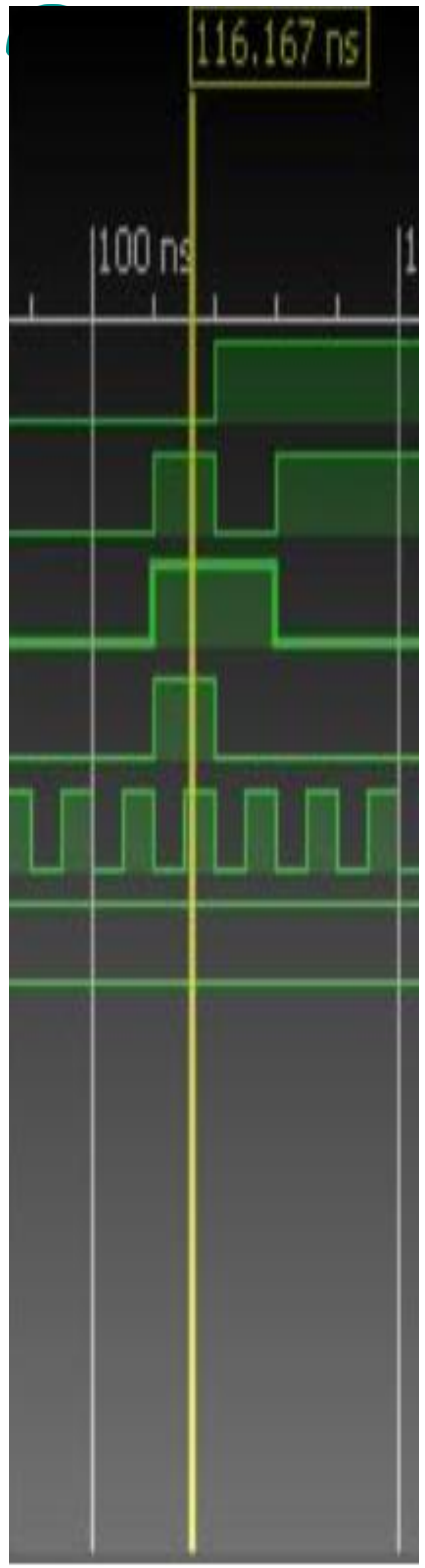


Martin González Pérez
martin.perez@cinvestav.mx

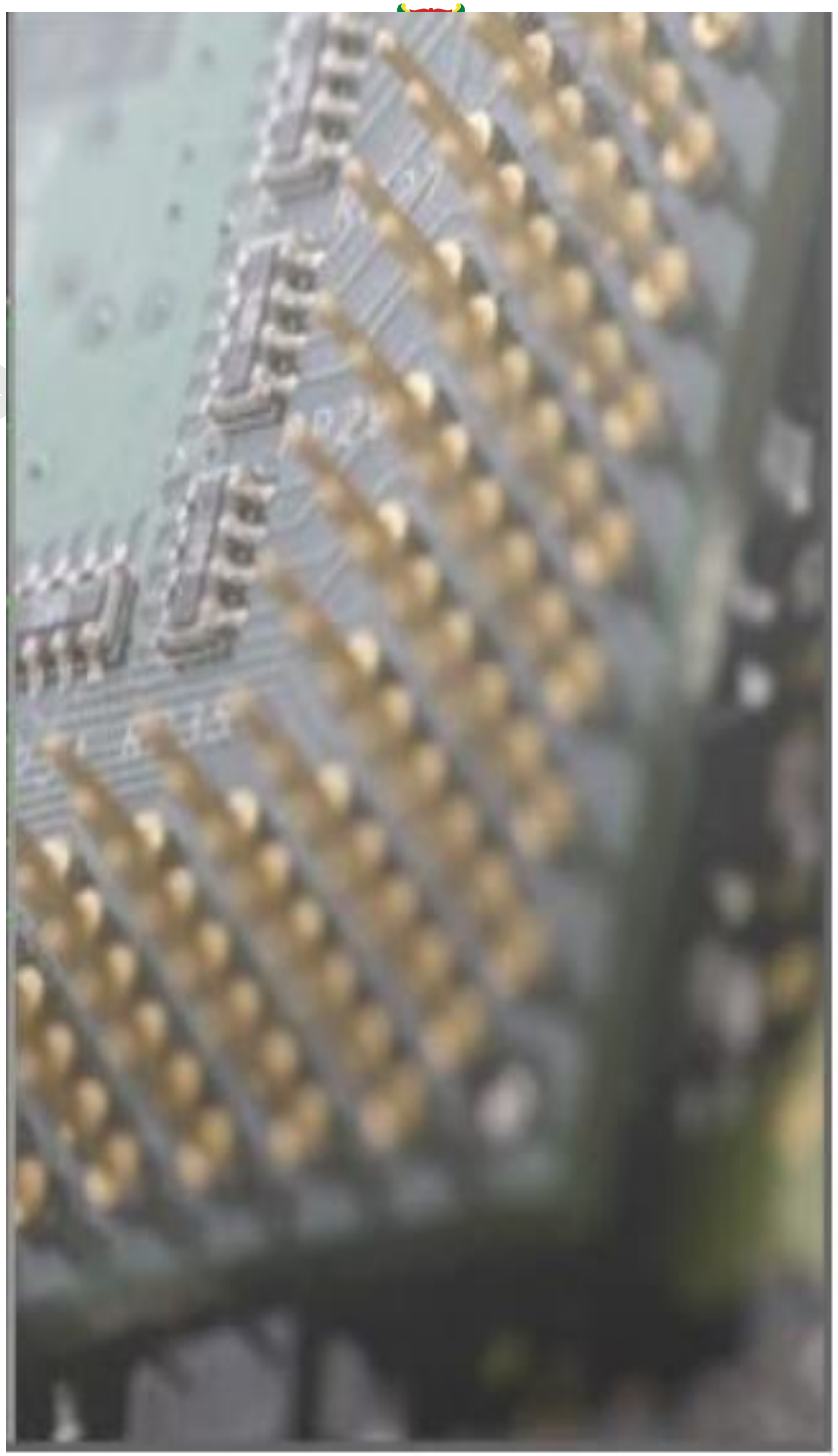
D Latch

D latch updates its state **continuously** while E = 1. We shall see later that is useful to update the state only at a specific **instant in time**. The **D Flip-Flop** does just that.





Flip-Flop



Cinvestav Guadalajara

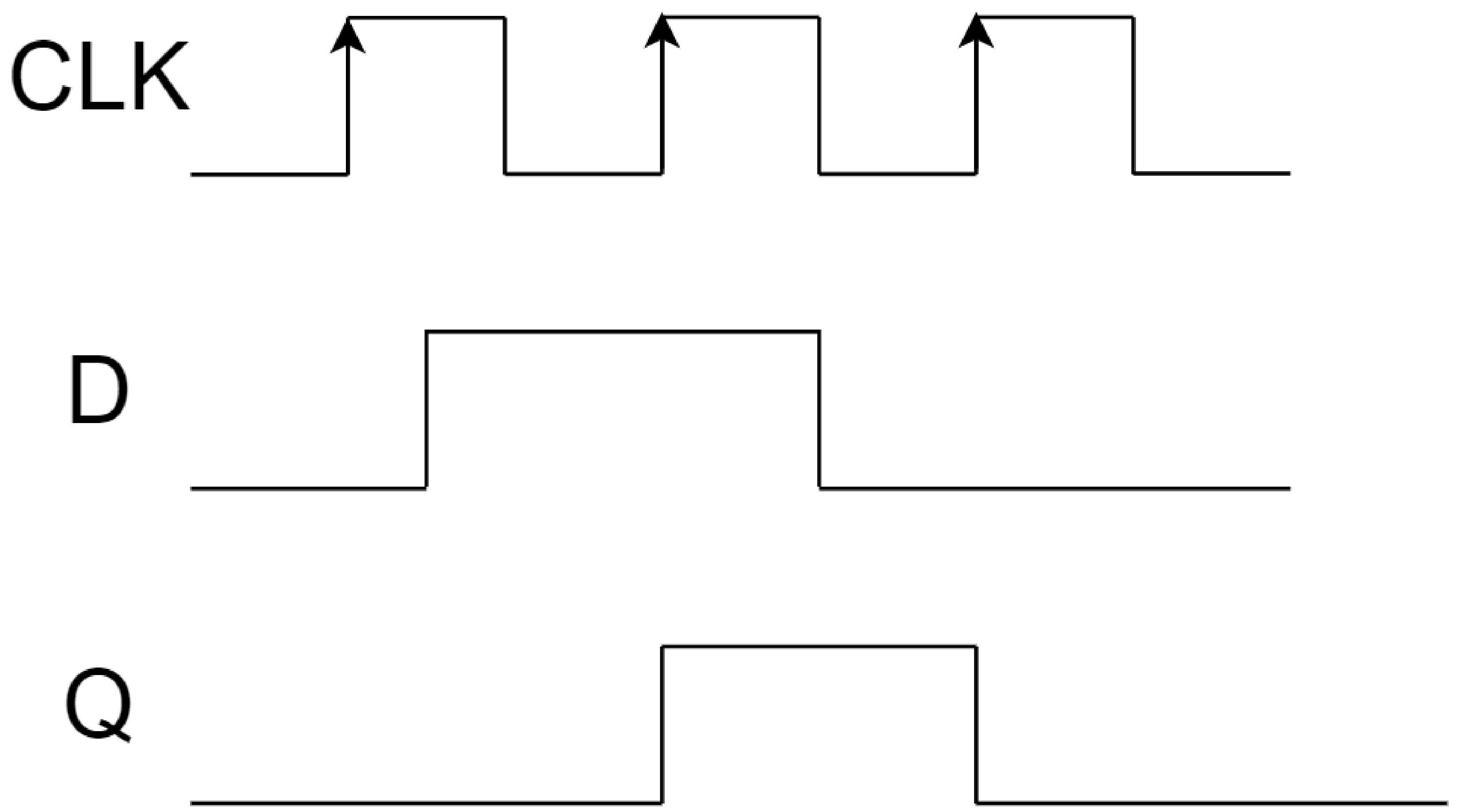
Nayarit NUESTRA LEALTAD Y COMPROMISO | COCYTEN

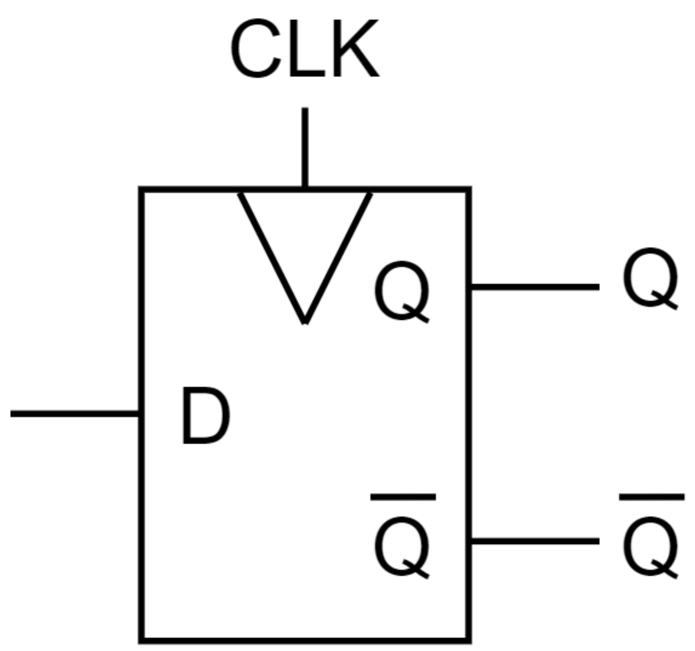
D Flip-Flop

D flip-flop has 2 inputs D and CLK.

Function:

- D is sampled on rising edge of CLK, when CLK changes from 0 to 1 D passes through to Q, otherwise, Q holds its previous value.

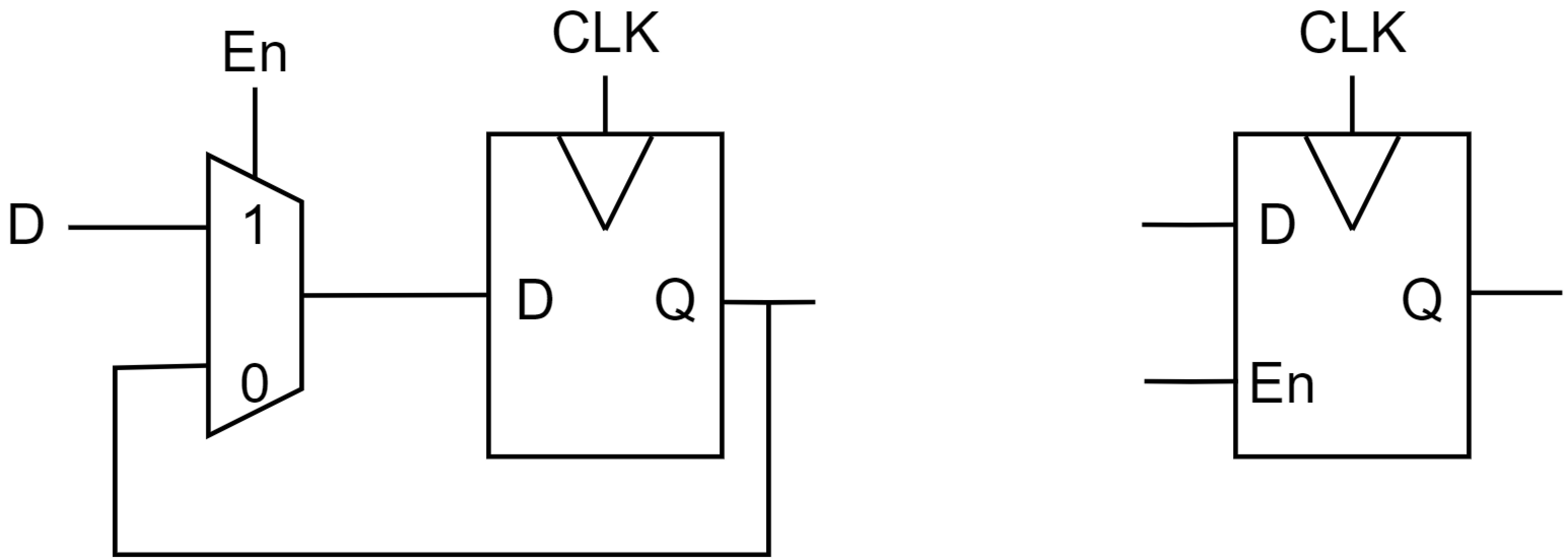




D Flip-Flop symbol

Flip-Flop with enable

An enabled flip-flop adds an enable input that determine whether the data (D) is loaded on the clock edge.



Flip-Flop with reset

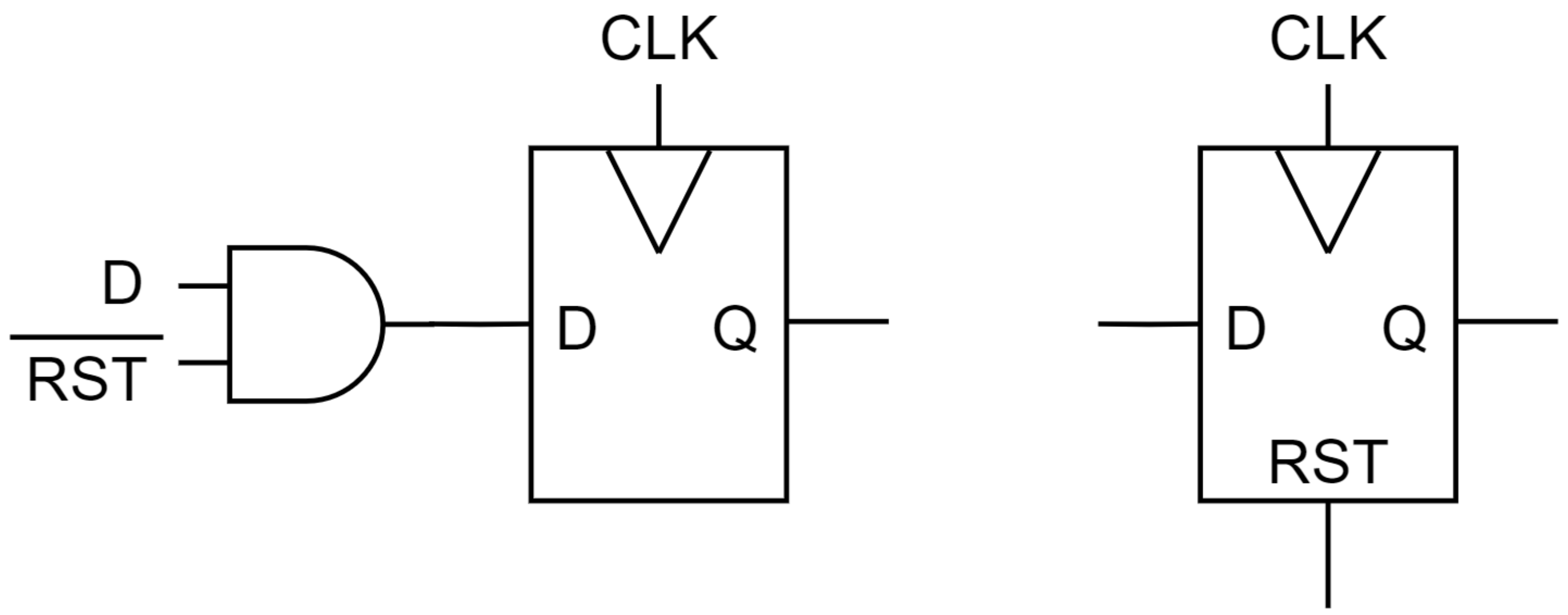
There are two types of resettable flip-flop:

- **Synchronous**: Reset at the clock edge only.
- **Asynchronous**: Reset immediately when reset is active.

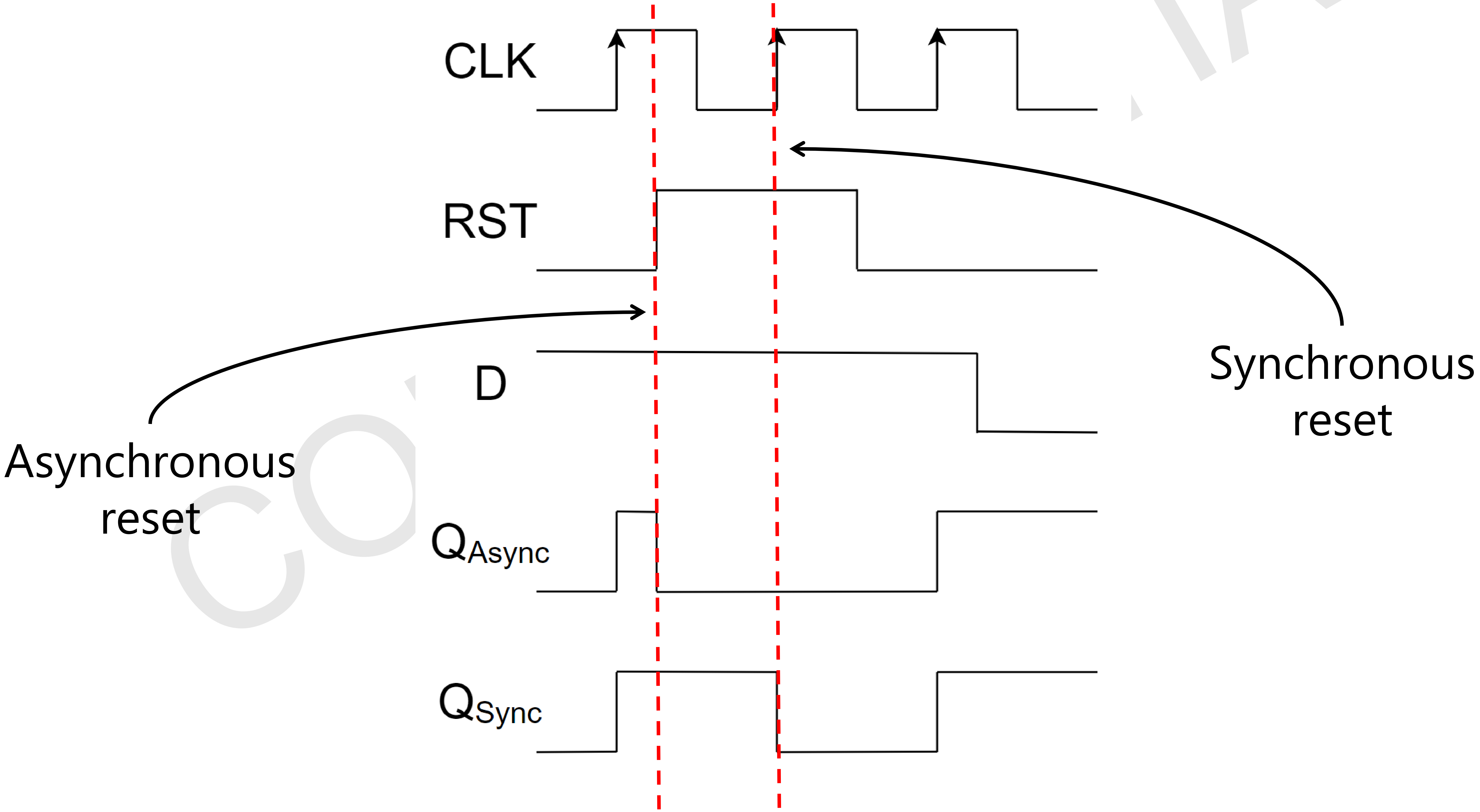
Asynchronous resettable flip-flop

Require changing internal circuit of the flip-flop.

Synchronous resettable flip-flop



Flip-Flop with reset

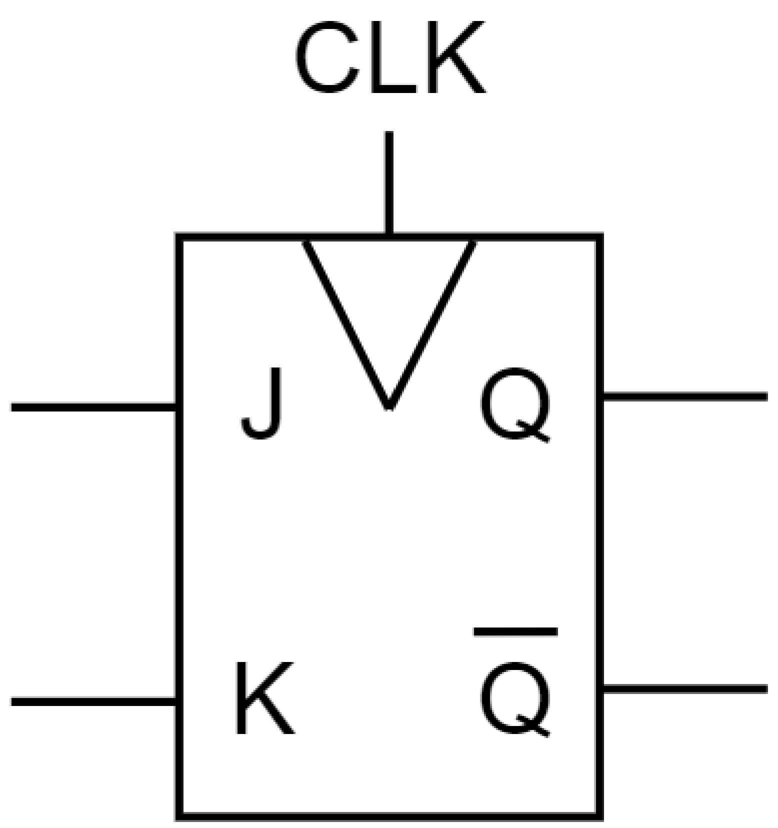


JK flip-flop

JK flip-flop has 2 inputs J(Set) and K(Reset).

J	K	Q	Q+
0	0	0	**0**
0	0	1	**1**
0	1	0	**0**
0	1	1	**0**
1	0	0	**1**
1	0	1	**1**
1	1	0	**1**
1	1	1	**0**

J	K	Q+
0	0	**Q**
0	1	**0**
1	0	**1**
1	1	$\overline{\mathbf{Q}}$



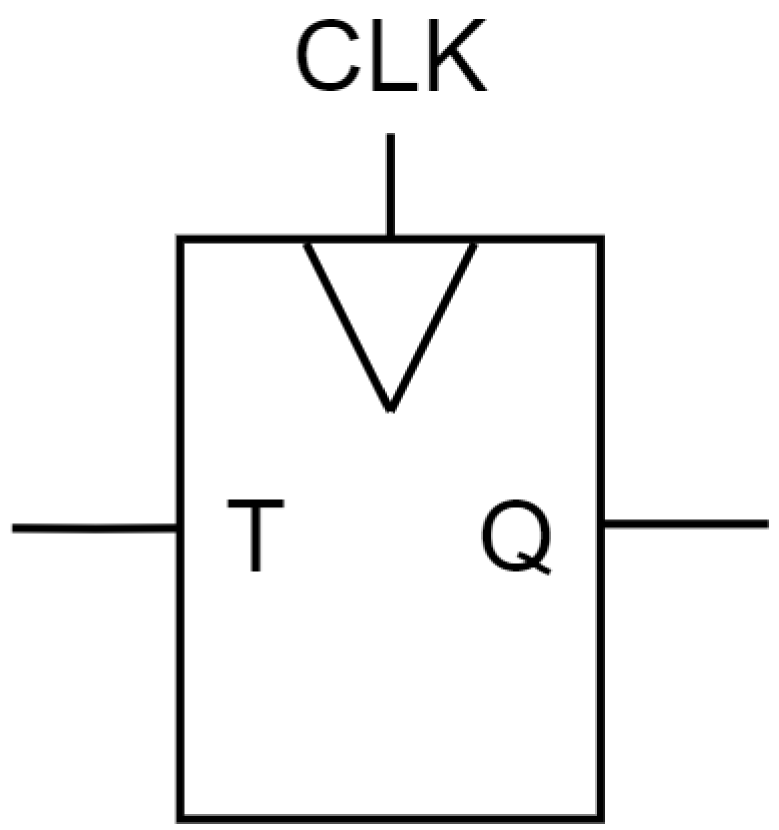
JK Flip-Flop symbol

T flip-flop

T flip-flop has one input (T), if T = 0 output doesn't toggle and if T=1 Q toggles.

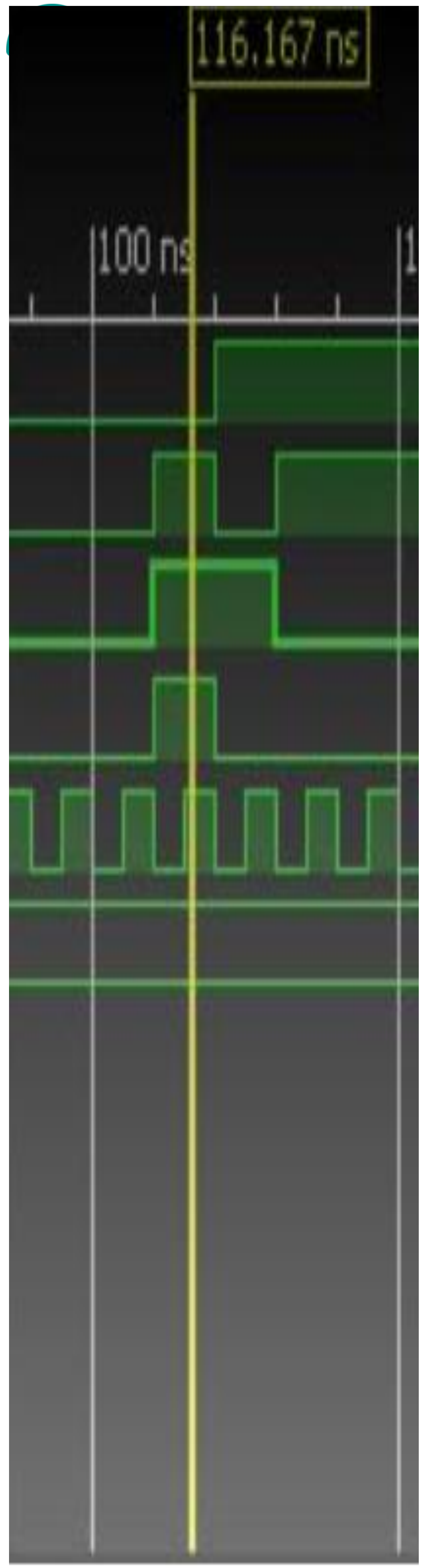
T	Q	Q+
0	0	**0**
0	1	**1**
1	0	**1**
1	1	**0**

T	Q+
0	**Q**
1	$\bar{Q}$

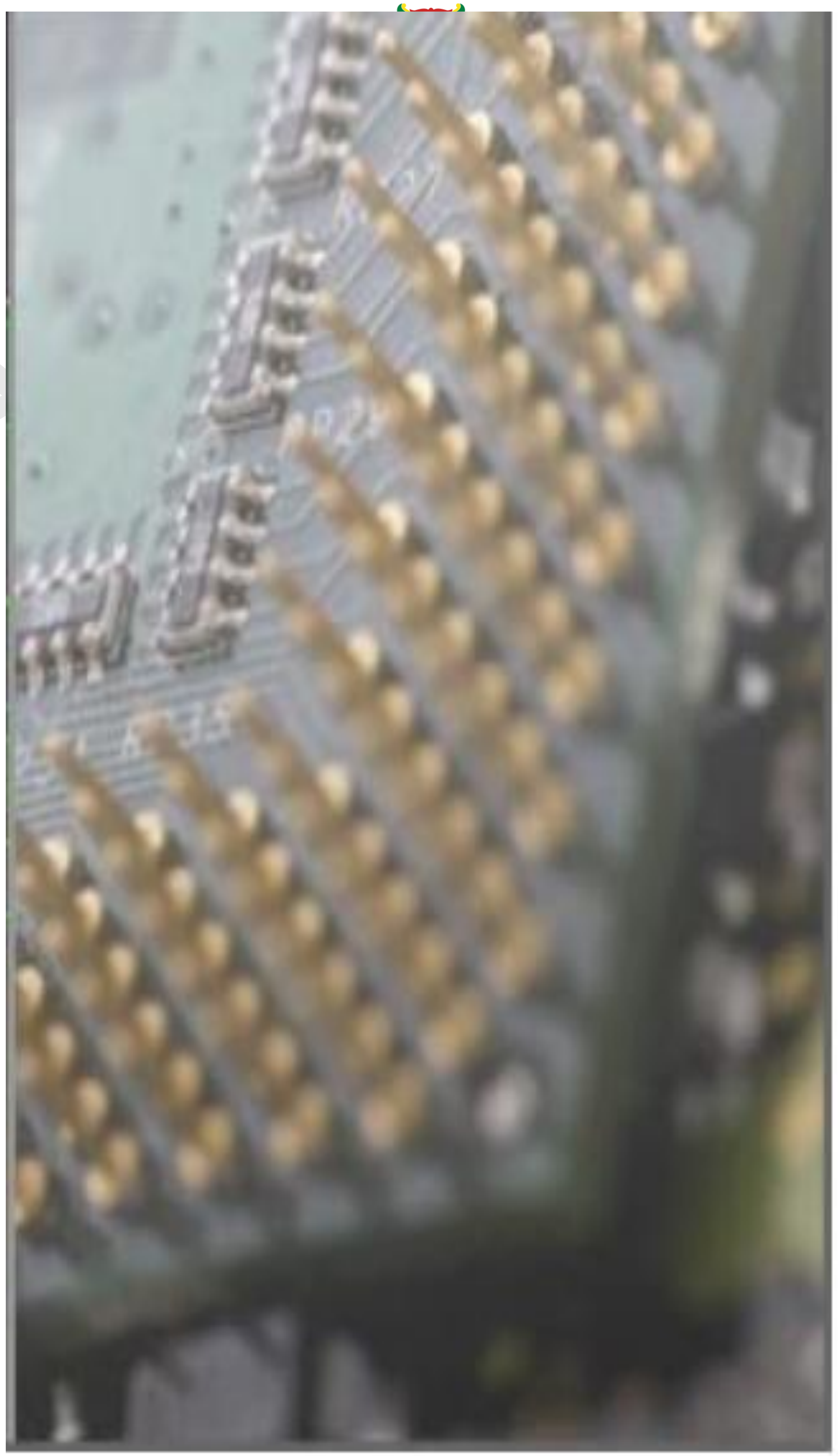


T Flip-Flop symbol

Martin González Pérez
martin.perez@cinvestav.mx



Synchronous circuits



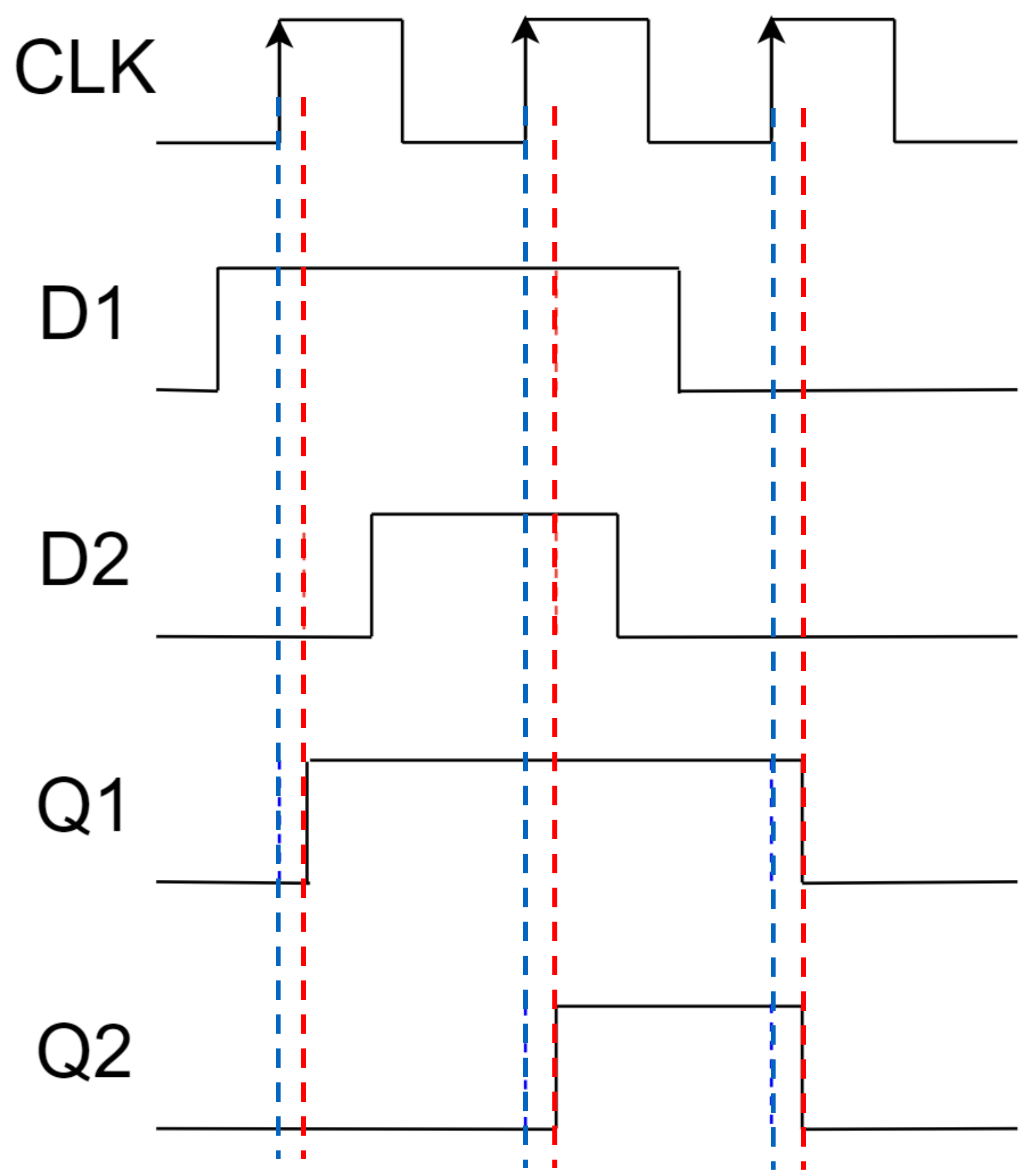
Synchronous circuit

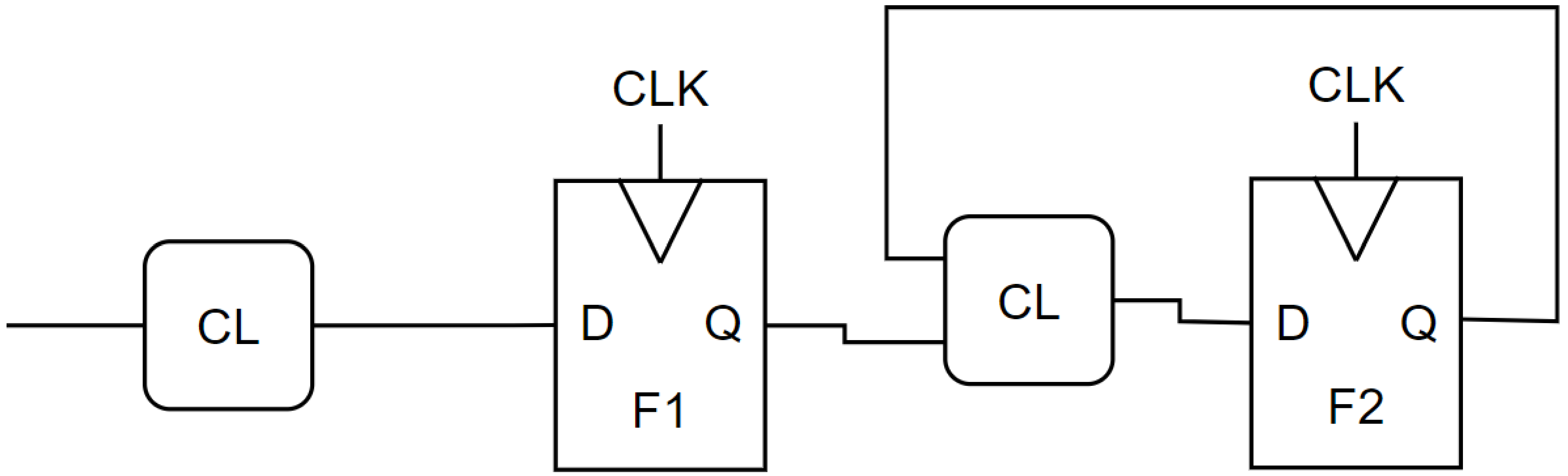
Sequential logic is not necessarily synchronous logic. Synchronous logic has some basic rules:

- Every element is either a register or a combinational circuit.
- At least one circuit is a register.
- All registers receive (are sensitive to) the same clock.
- Every cyclic path contains at least one register (remember, combinational circuits are no cyclic).

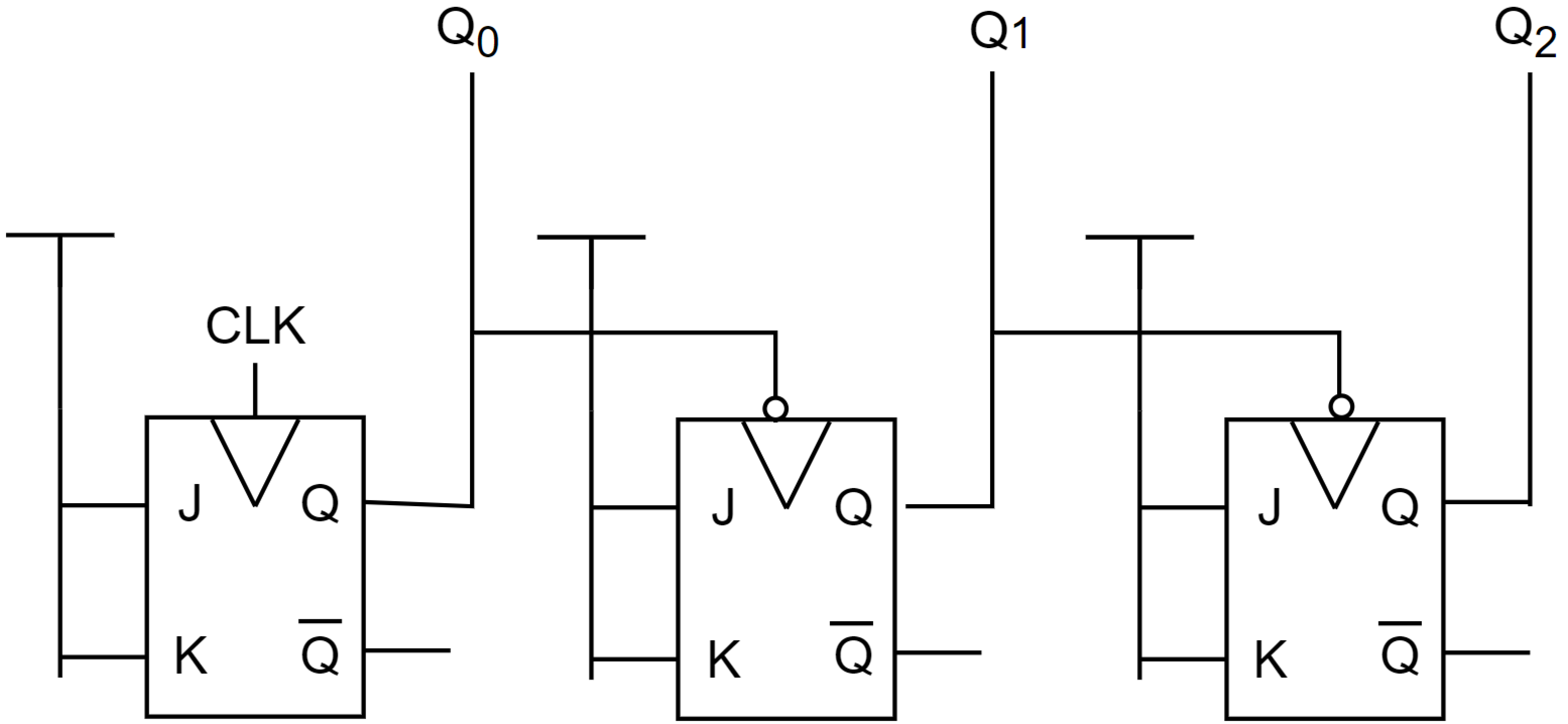
Synchronous circuit

In a synchronous circuit, all registers are updated with the same clock, which prevents the combinational logic delays from having a cumulative effect.





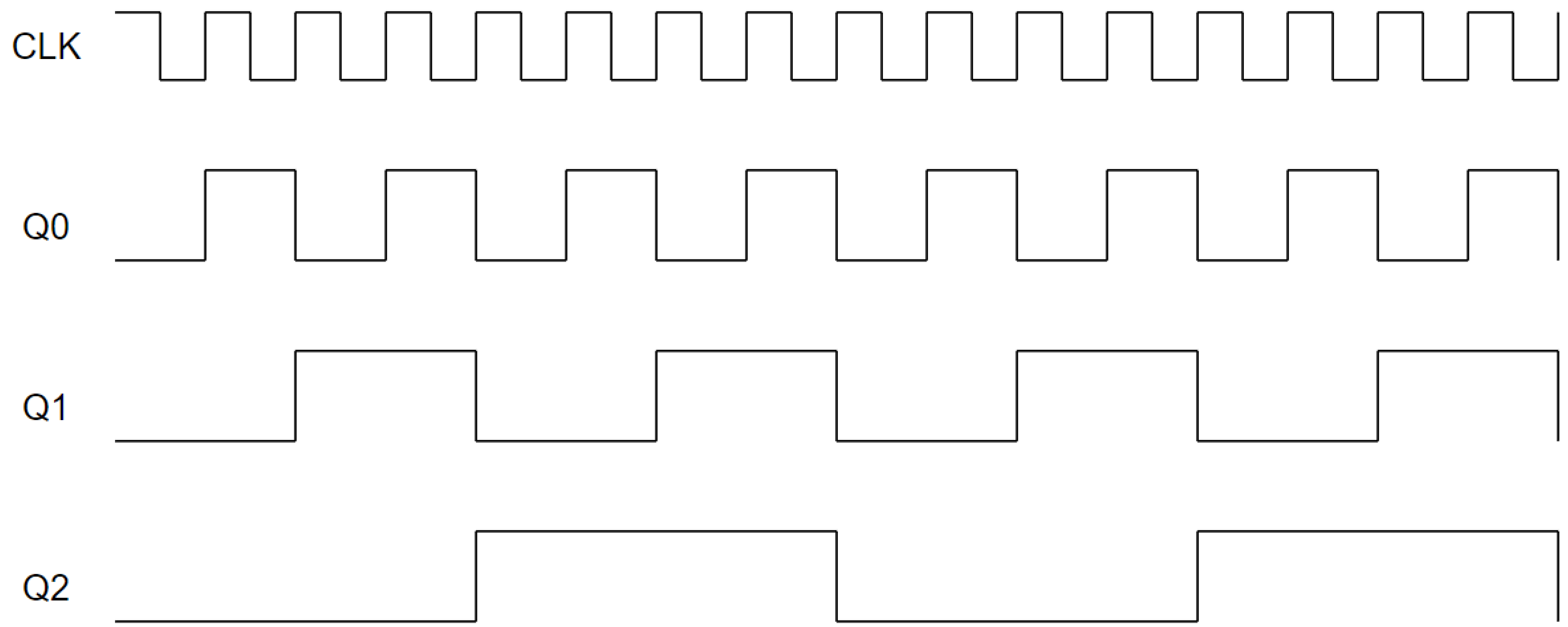
Is this circuit synchronous?

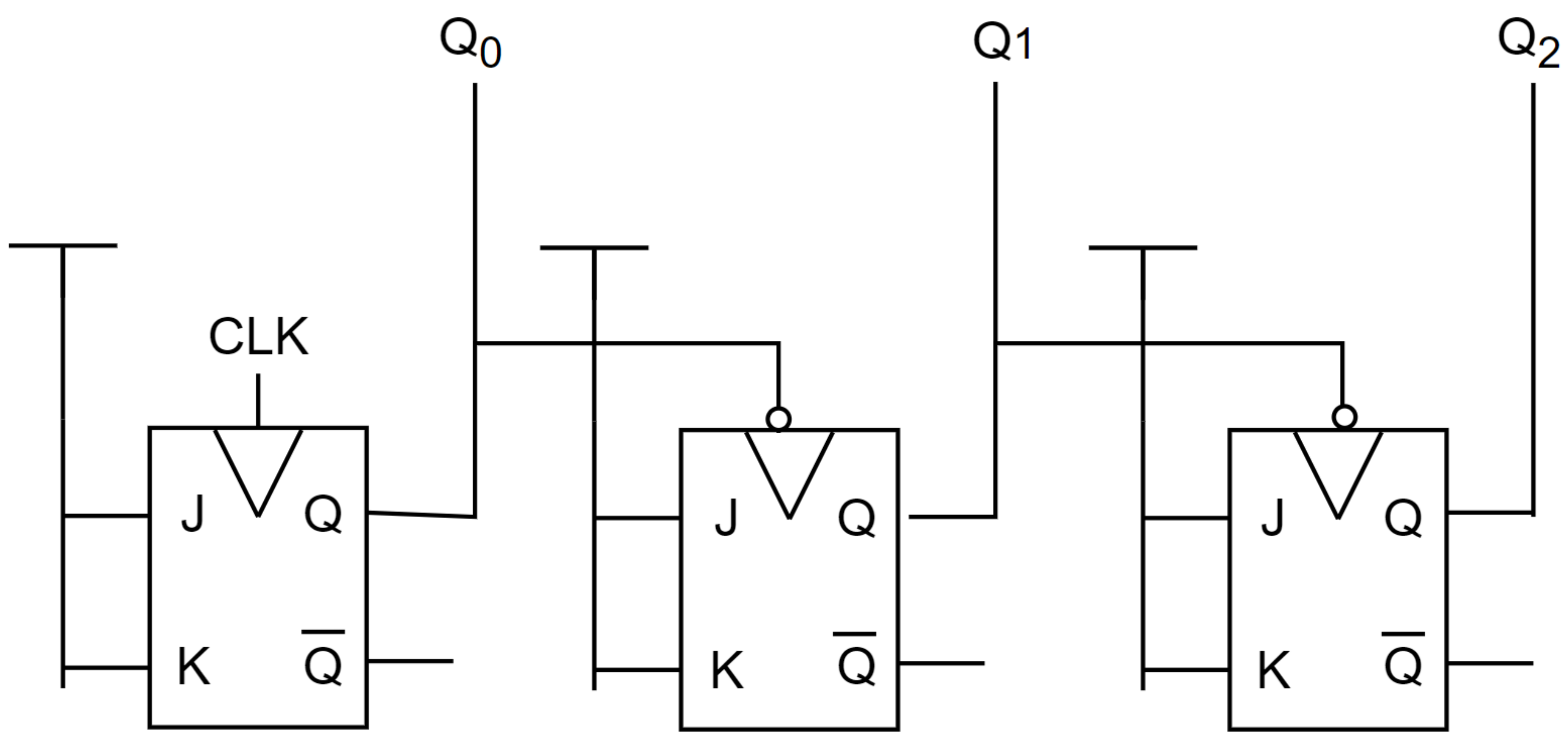


Martin González Pérez
martin.perez@cinvestav.mx

Asynchronous counter

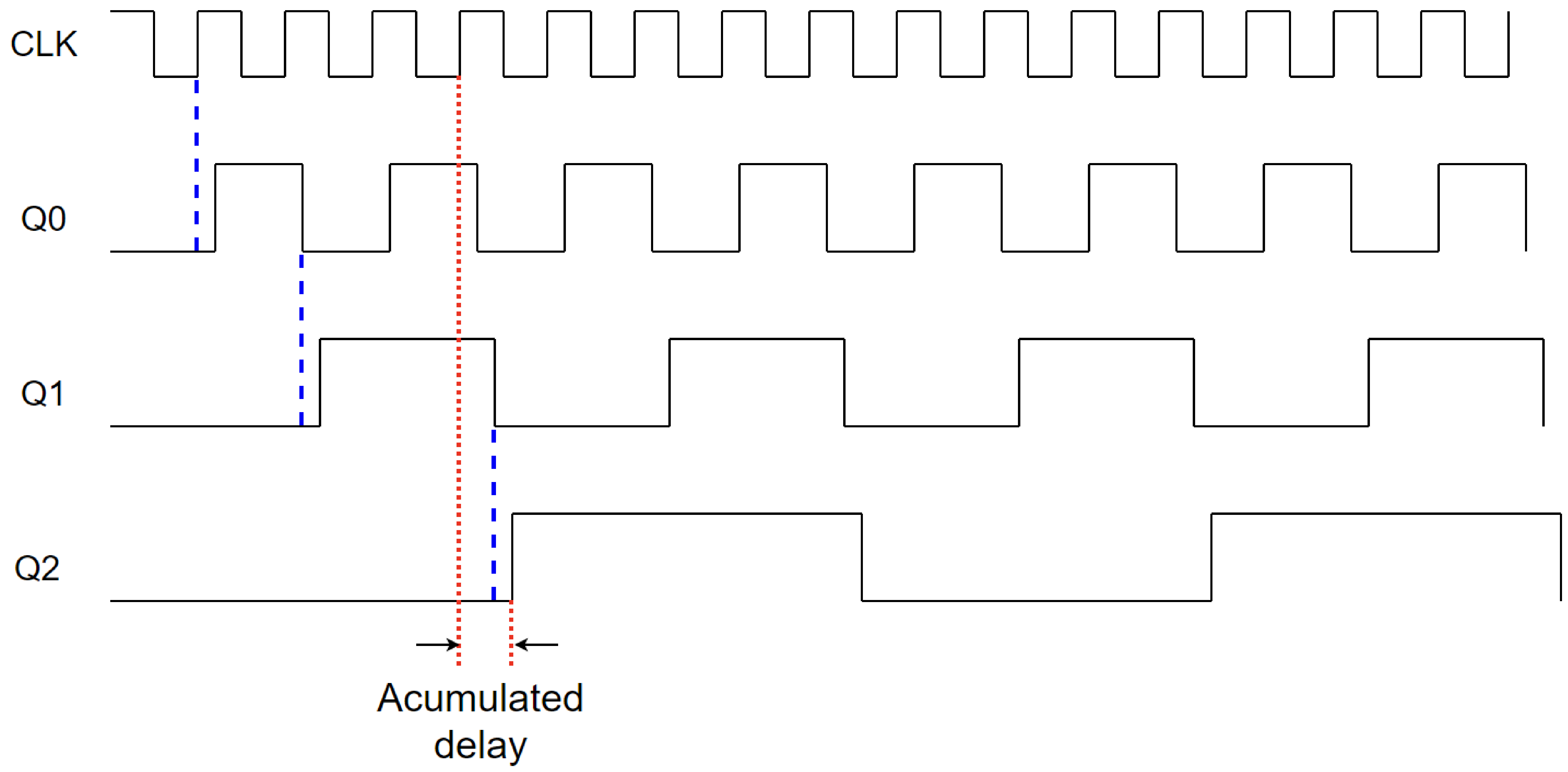
No delay

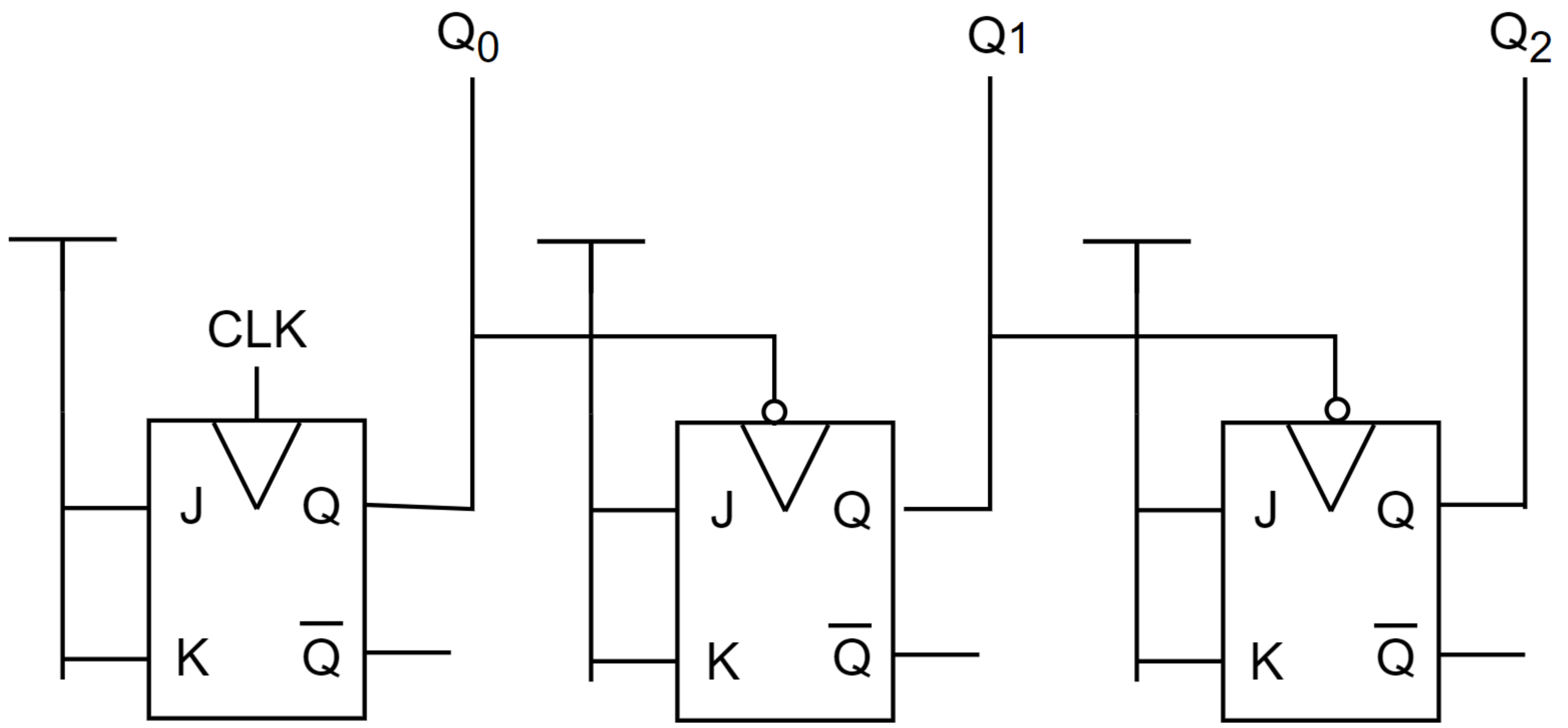


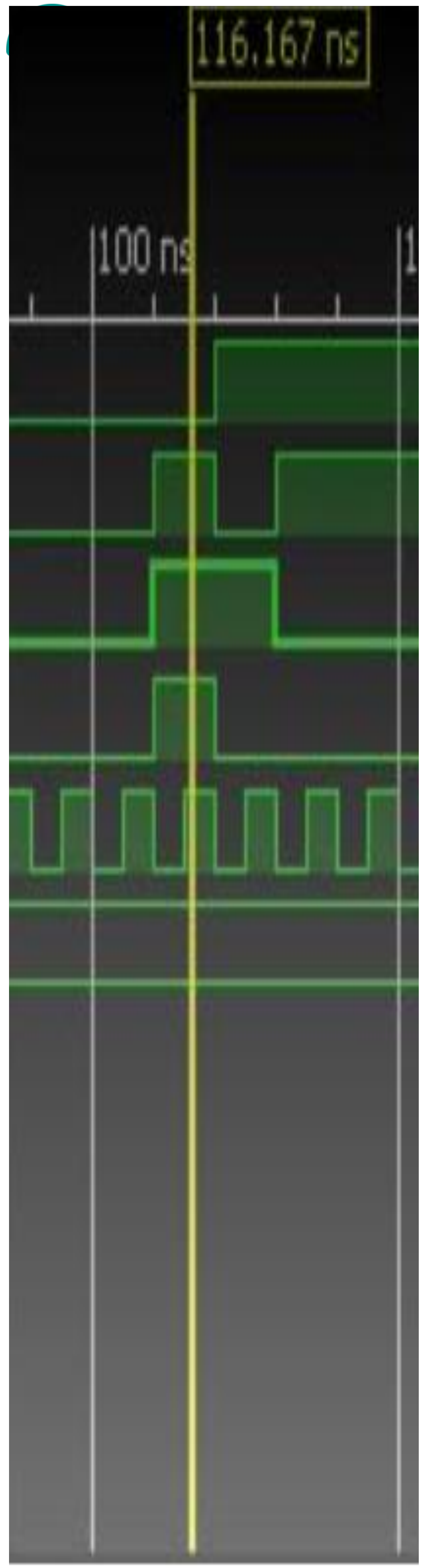


Asynchronous counter

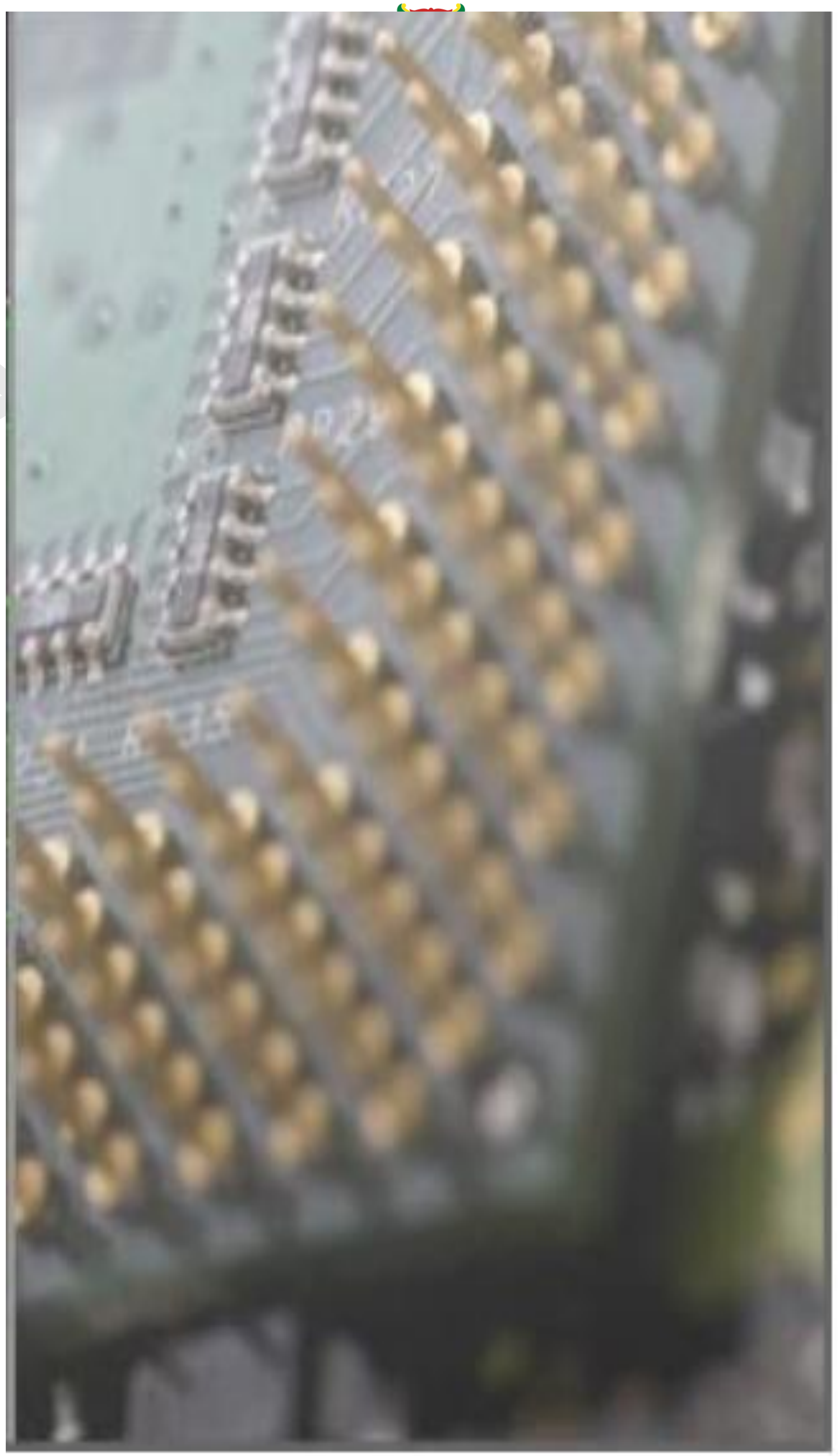
Delay considerations







Timing (Overview)



Sequential logic timing

Flip-Flops sample inputs at clock edge, but what if input change at the same time that clock transition happens? The sampled value could be incorrect.

That's why input must be stable while is sampled, the time that inputs should be stable is defined by **setup time** and **hold time** constraints.

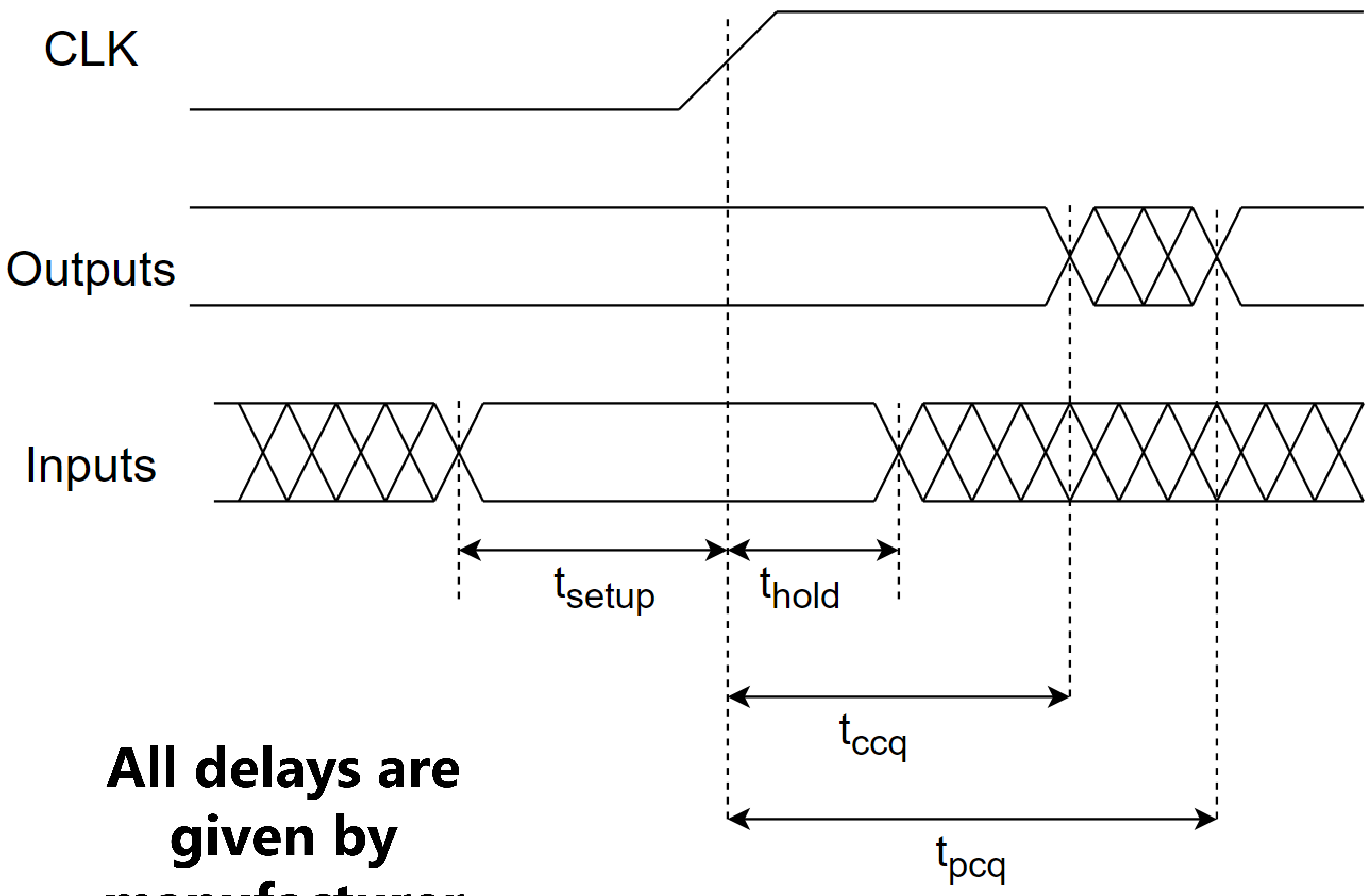
Flip-Flop timing

t_{setup}: time before clock edge data must be stable.

t_{hold}: time after clock edge data must be stable.

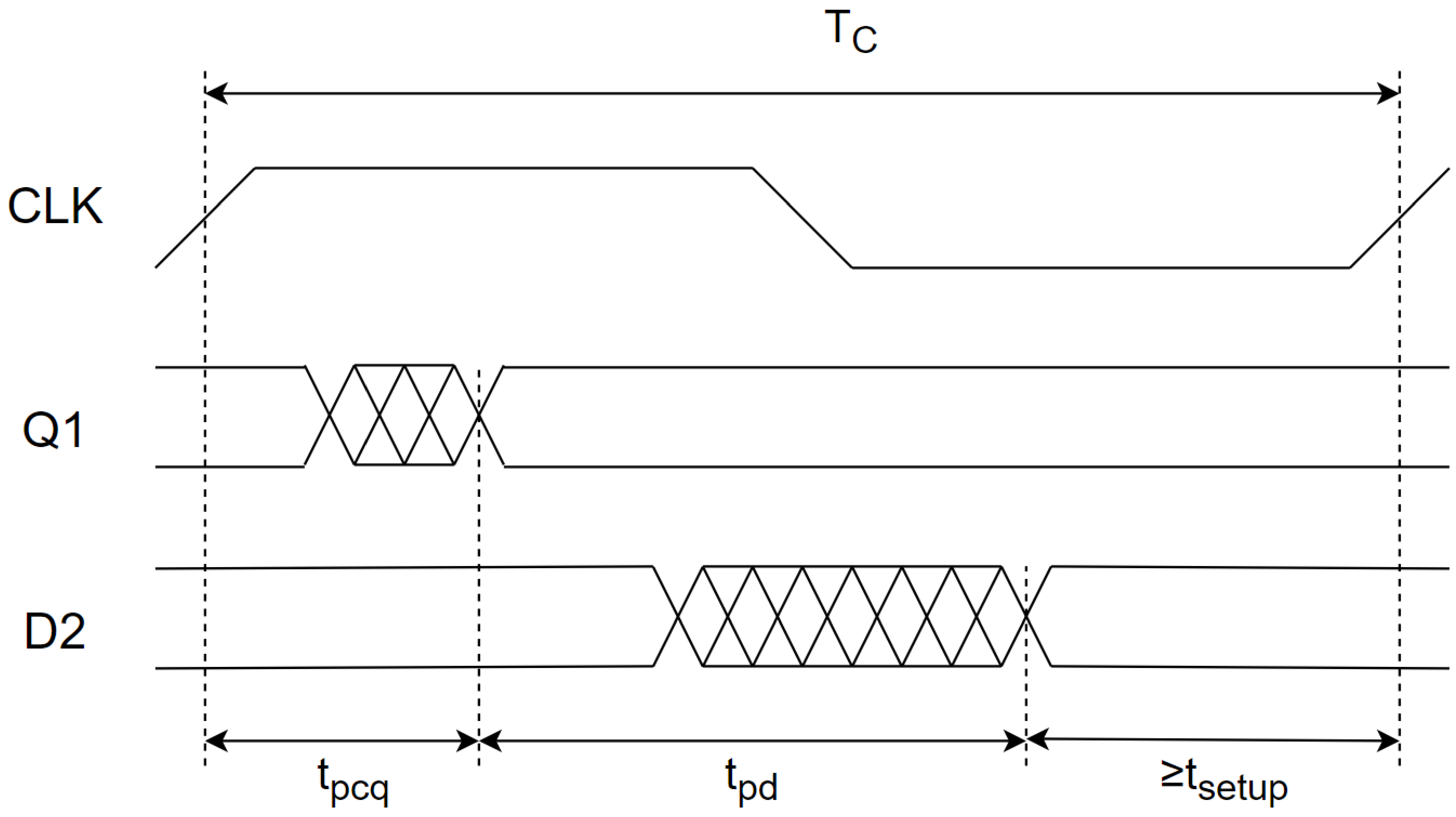
t_{ccq}: contamination delay clock to q (shortest path).

t_{pcq}: Propagation delay clock to q(longest path).



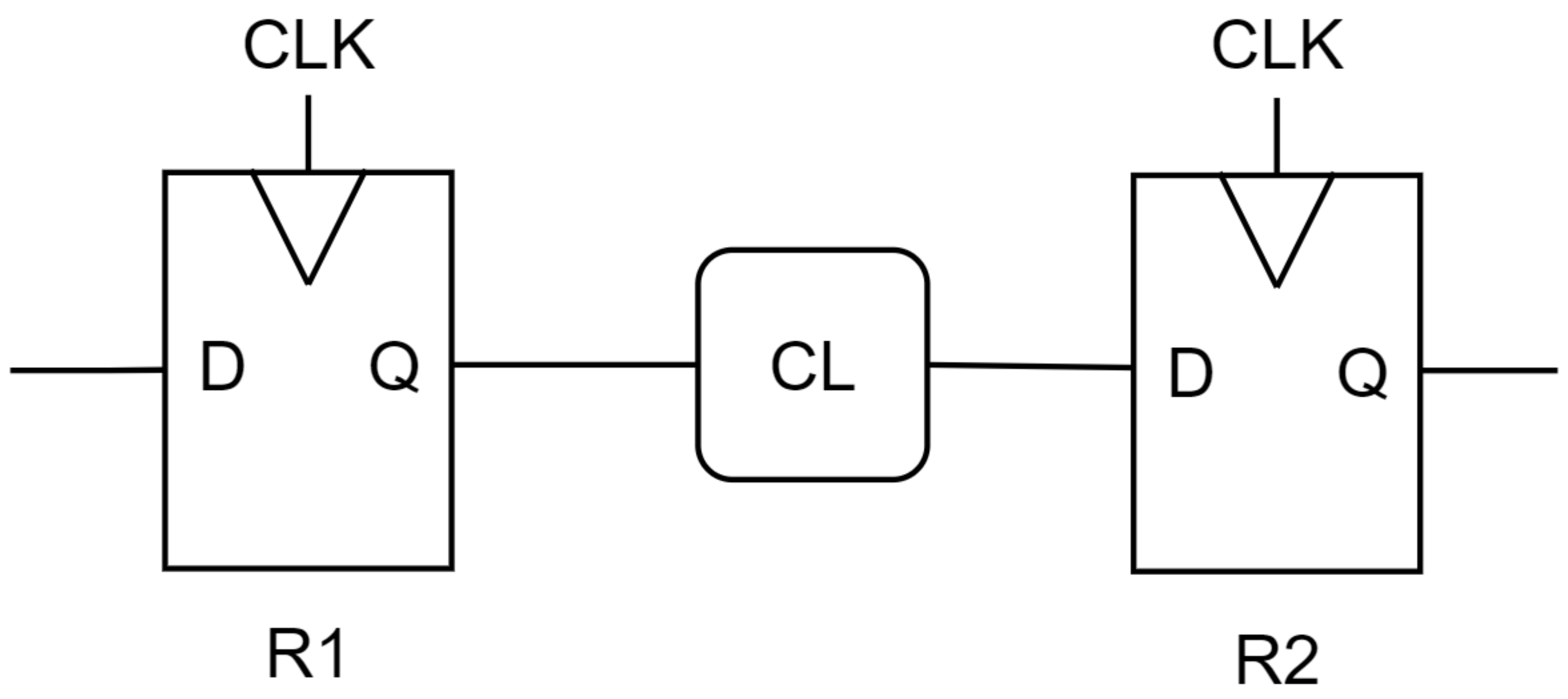
All delays are given by manufacturer

Set up time

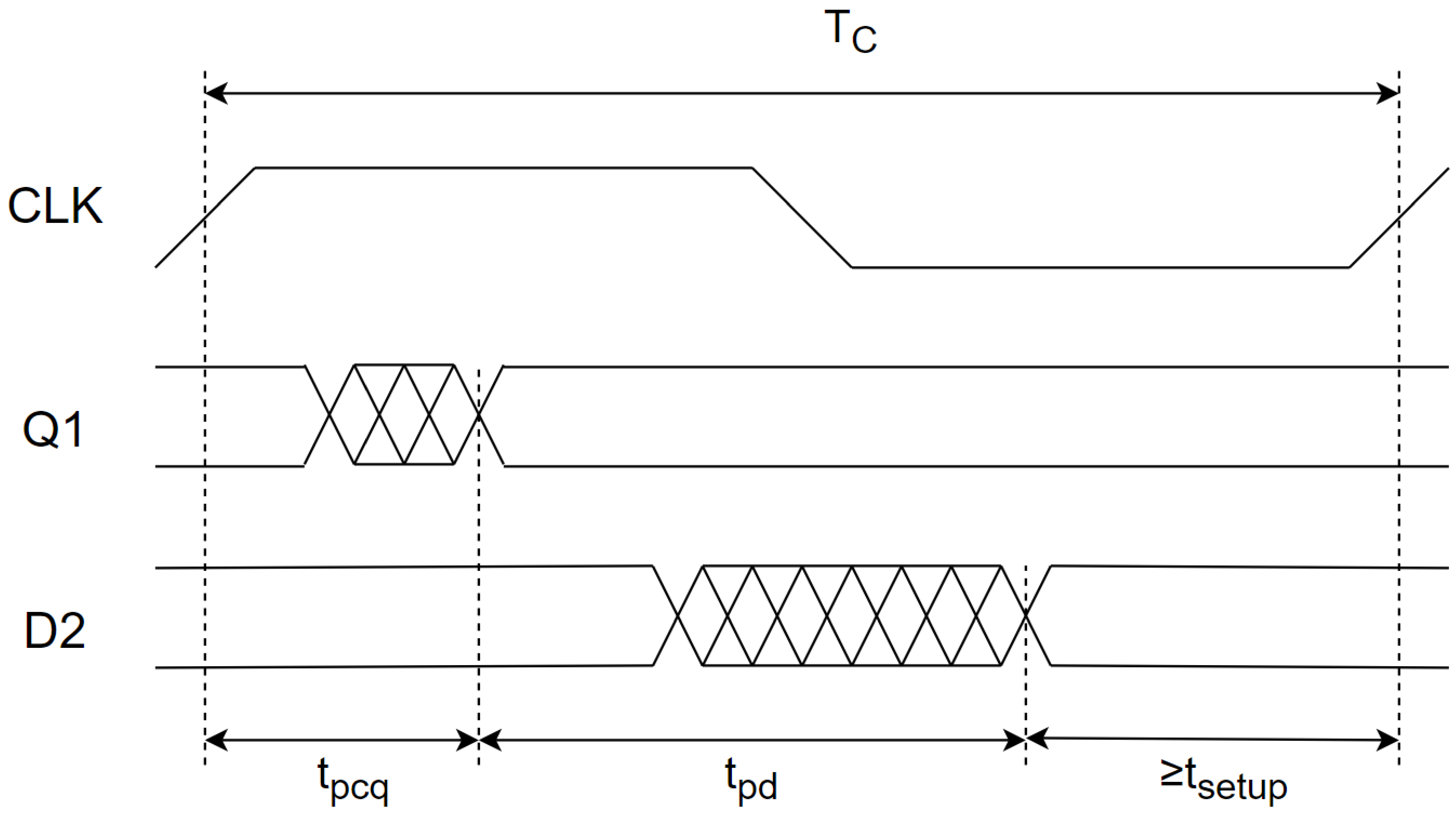


Minimum clock period can be determined by:

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

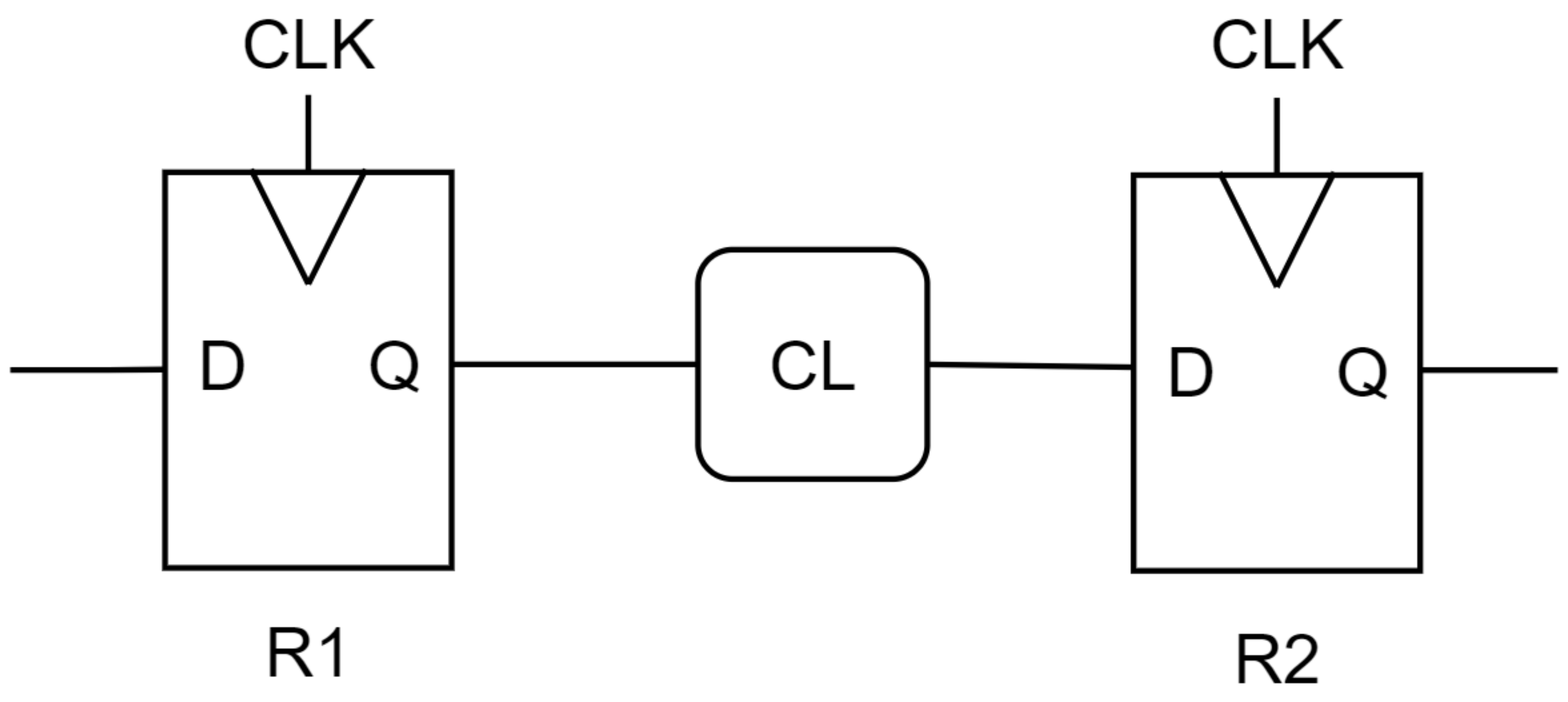


Set up time

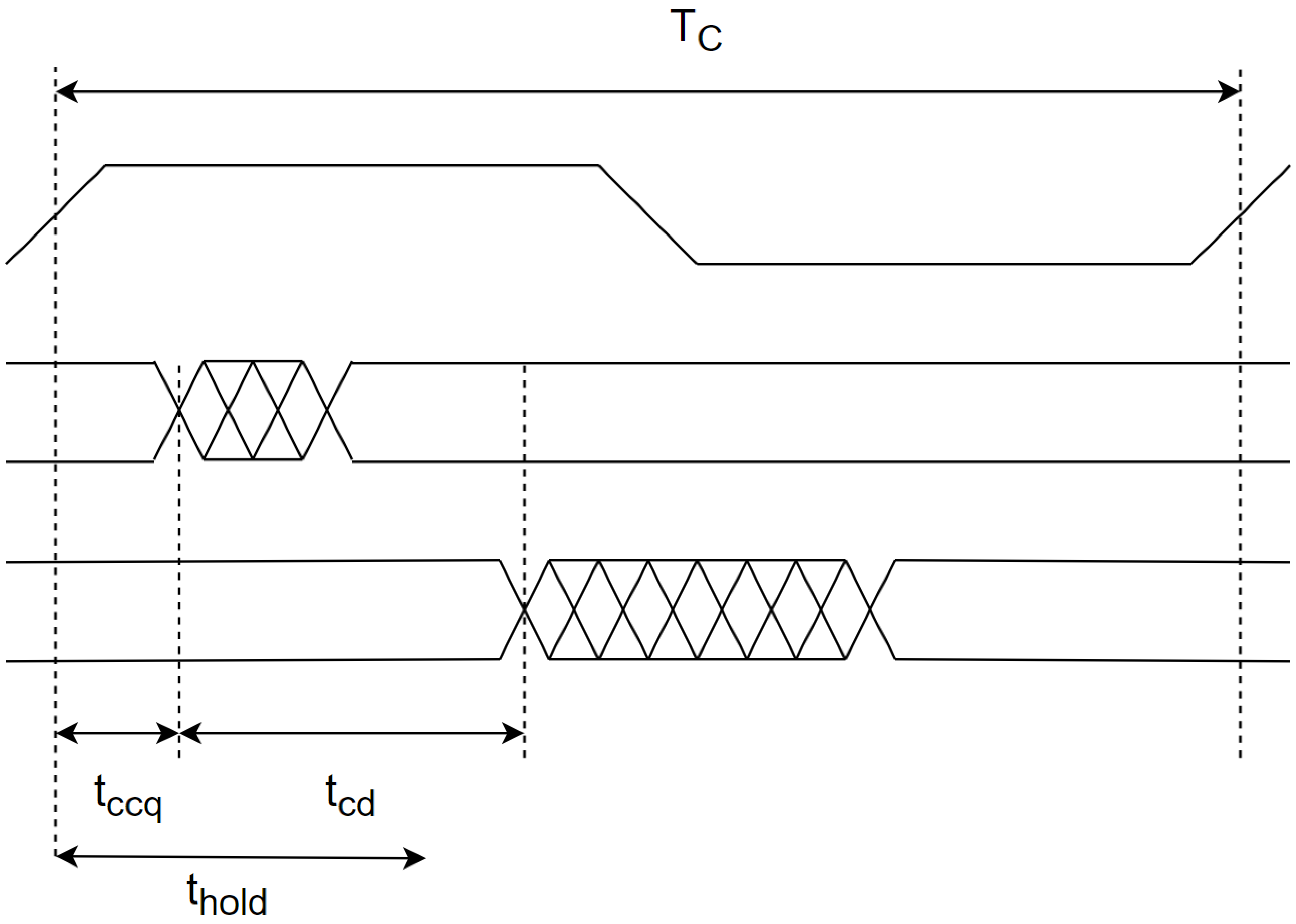


In a project, clock period is often defined for the specs, so the propagation delay through combinational logic is the only variable under control.

$$t_{pd} \leq T_C - (t_{pcq} + t_{setup})$$

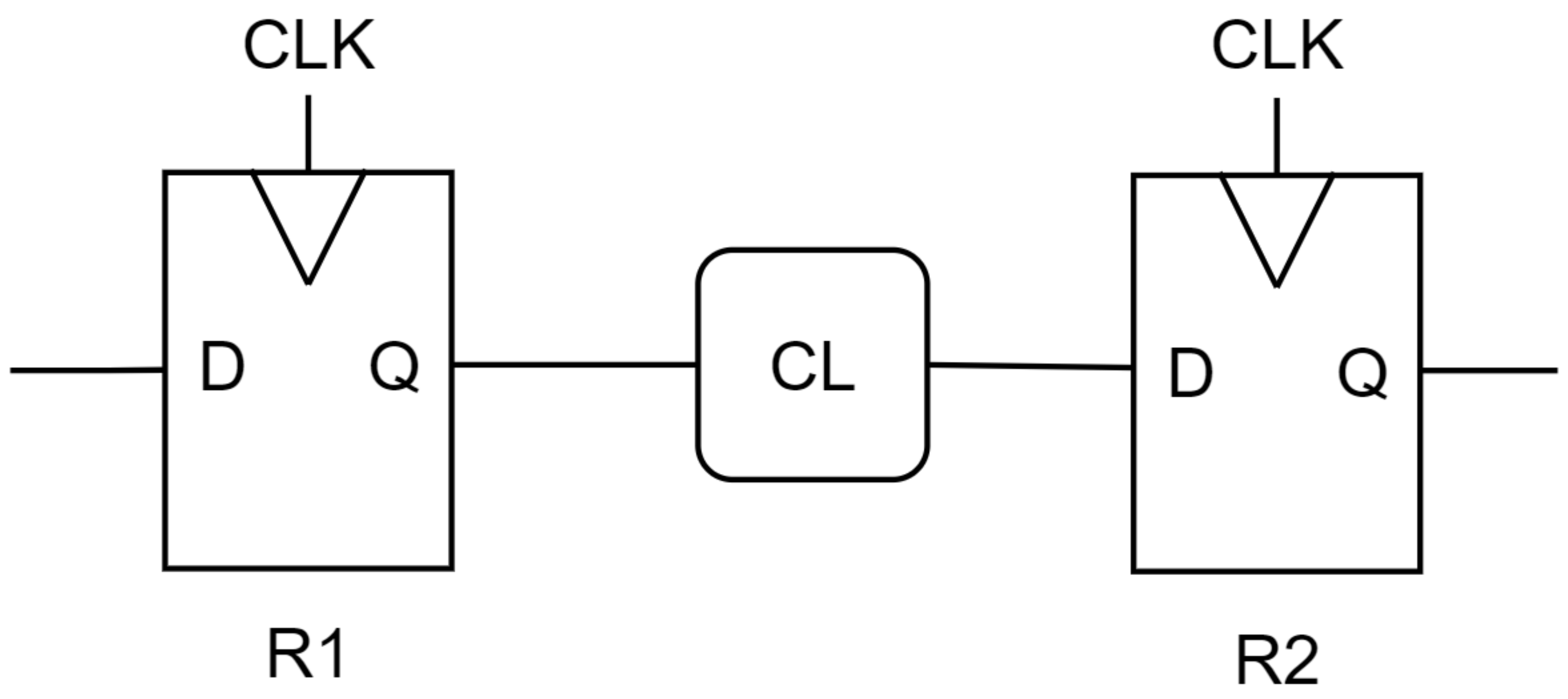


Hold time

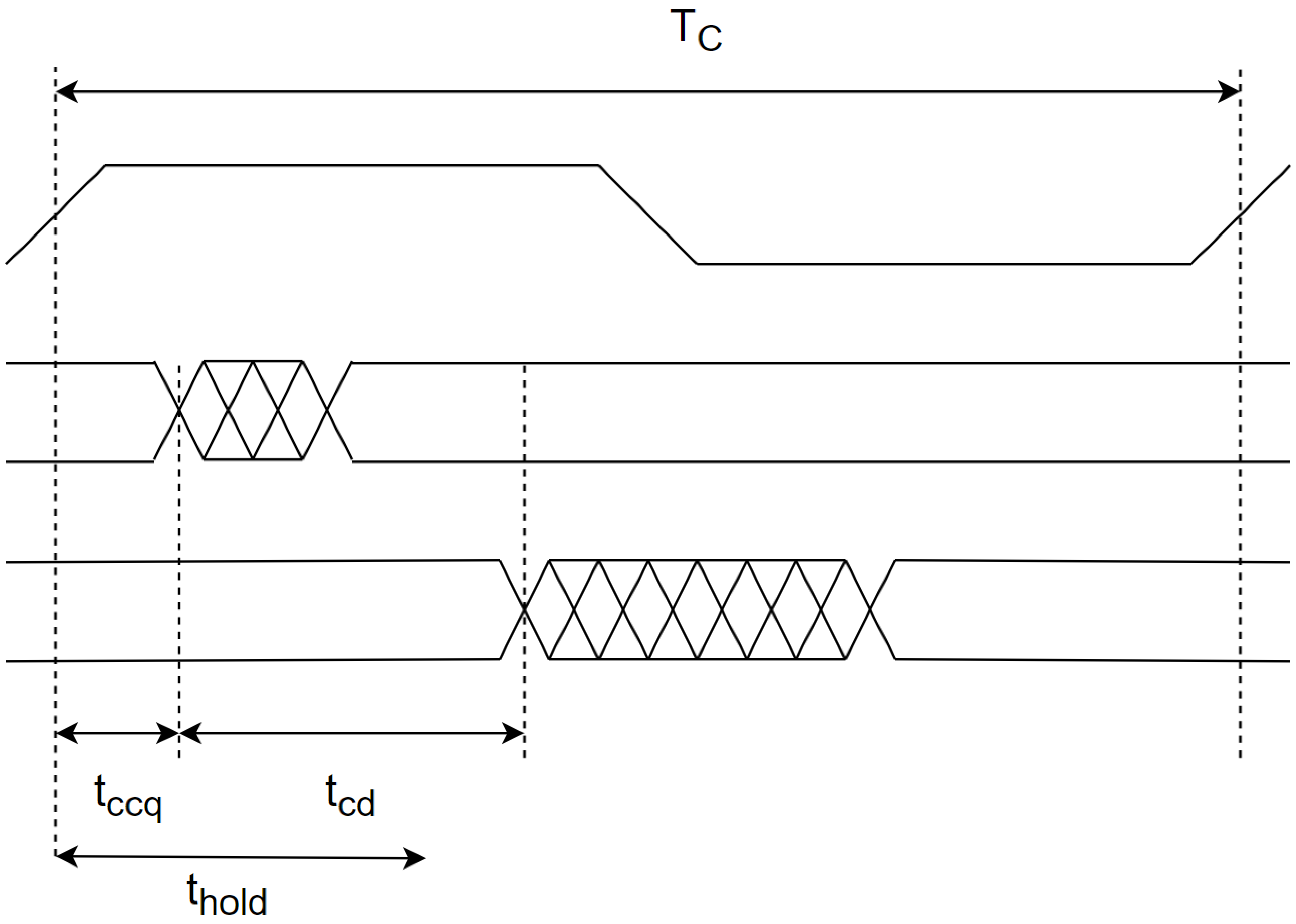


Input must be stable t_{hold} time after clock edge, we know that input can change as soon as $t_{ccq}+t_{cd}$ time.

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



Hold time



As designer, t_{cd} is the only variable under control, so we can solve for the minimum contamination delay (t_{cd}) through combinational logic.

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