MSE | MASTER OF SCIENCE IN ENGINEERING

Hes·so // FRIBOURG FREIBURG
Haute Ecole Spécialisée de Suisse occidentale
Fachhochschule Westschweiz
University of Applied Sciences and Arts Western Switzerland

Chapitre 1

Digital Timing Analysis MA-AdvElDes

Pietro Buccella

A. A. 2021/22

Objectives

Objectives of the lesson:

- Know the different timing parameters of logic circuits
- Understand the concepts and arithmetic behind digital logic circuitry timing analysis
- Calculate the maximum operating clock frequency of a digital circuit
- Use LTspice® software to perform timing analysis of simple logic gates

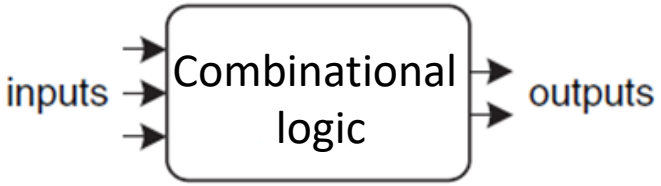
Contents

- Logic Circuits: Combinational and Sequential
- Transient Behaviour of Logic Circuits
- Timing Parameters for Combinational Logic
- Timing Parameters for Synchronous Sequential Logic
- Timing Analysis of Synchronous Sequential Logic: Setup and Hold Time Constraints
- Timing Analysis – Inside a Component (FPGA/ASIC)
- Delays – where do they come from
- Exercises

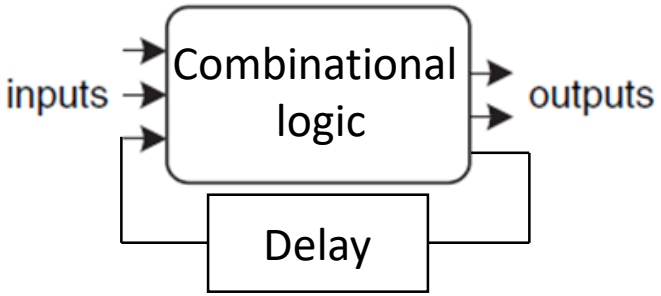
Logic Circuits

- **Combinational logic**: The outputs depend only on current input values
- **Sequential logic:** The outputs depend on both current and prior input values→ Combinational logic + Memory elements
- **Sequential logic:** Synchronous Sequential and Asynchronous Sequential Circuits
- Asynchronous circuits: An un-clocked flip flop or time delay element acts as memory element
- Synchronous circuits: A clocked flip flop acts as memory element; the status of memory element is affected only at the active edge of clock, if input is changed

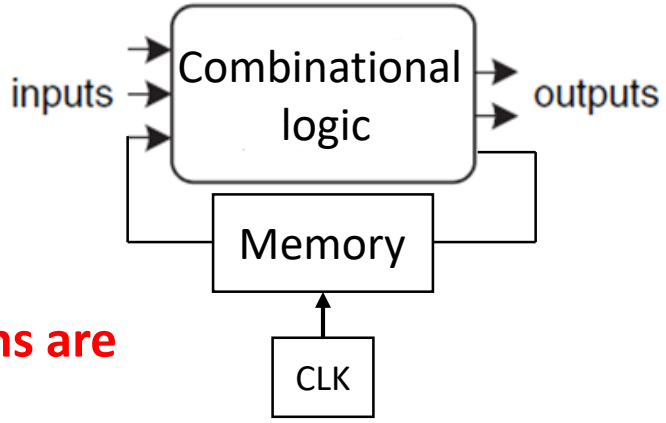
Combinational logic



Asynchronous logic



Synchronous logic

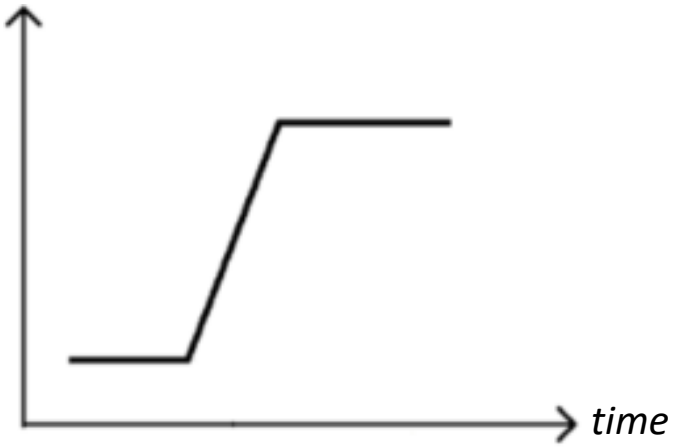


Virtually all digital sequential systems are synchronous

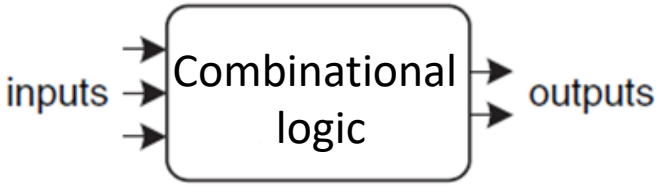
Transient Behaviour of Logic Circuits

How do signal of digital circuits evolve from one value to the next?

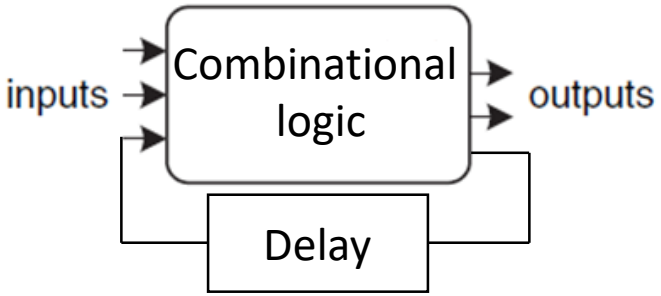
- *Ideal case: a logic signal* varies between a low and a high signal level in monotonic ramp



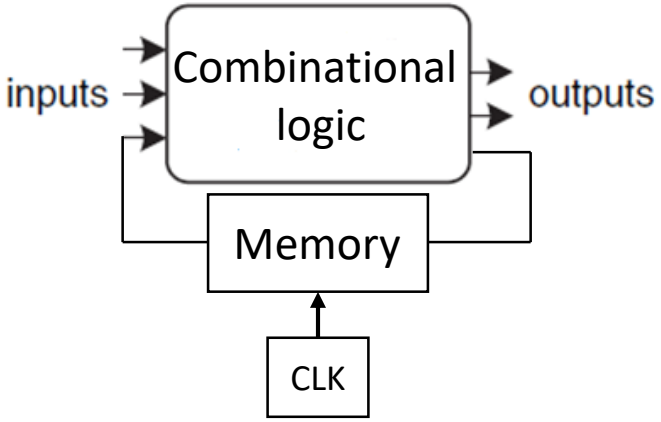
Combinational logic



Asynchronous logic



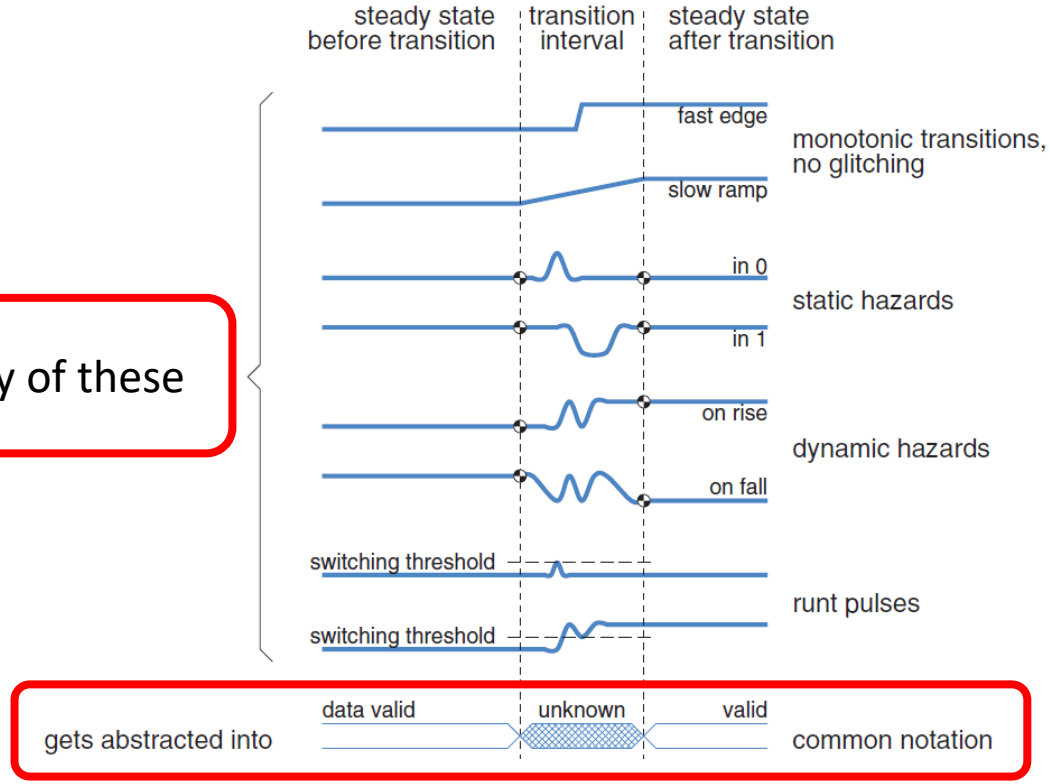
Synchronous logic



Transient Behaviour of Logic Circuits

How do the outputs of digital circuits evolve from one value to the next?

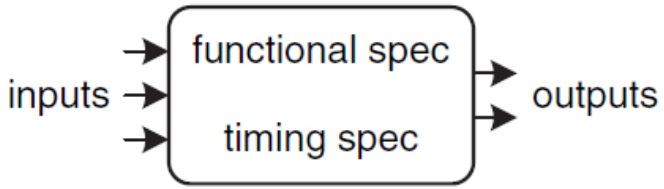
- *Real case* → Any of these

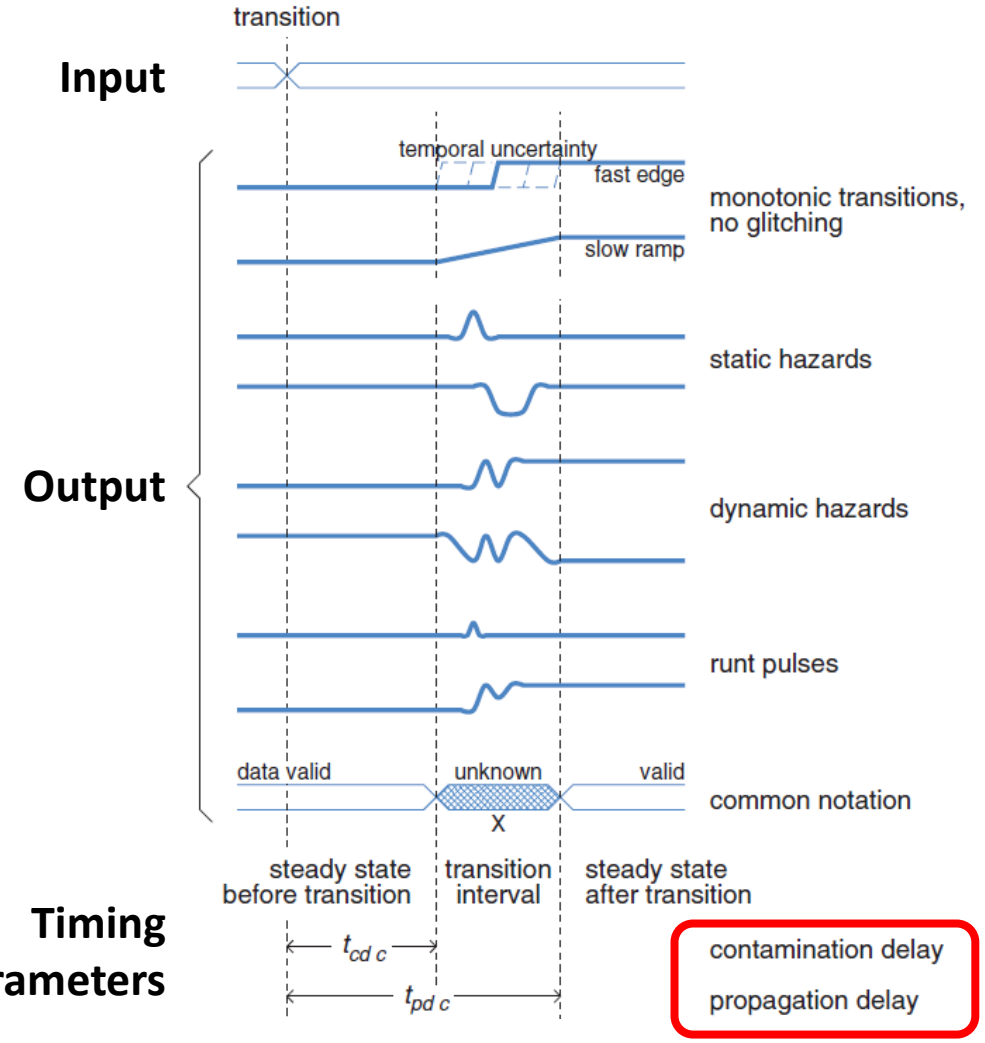


Transient Behaviour of Logic Circuits

How long transient phenomena persist at the output of a digital circuit in response to a change at one of the circuit's inputs?

- it takes a pair of **Timing Parameters** :
  - Contamination delay: t_{cd}
  - Propagation delay: t_{pd}

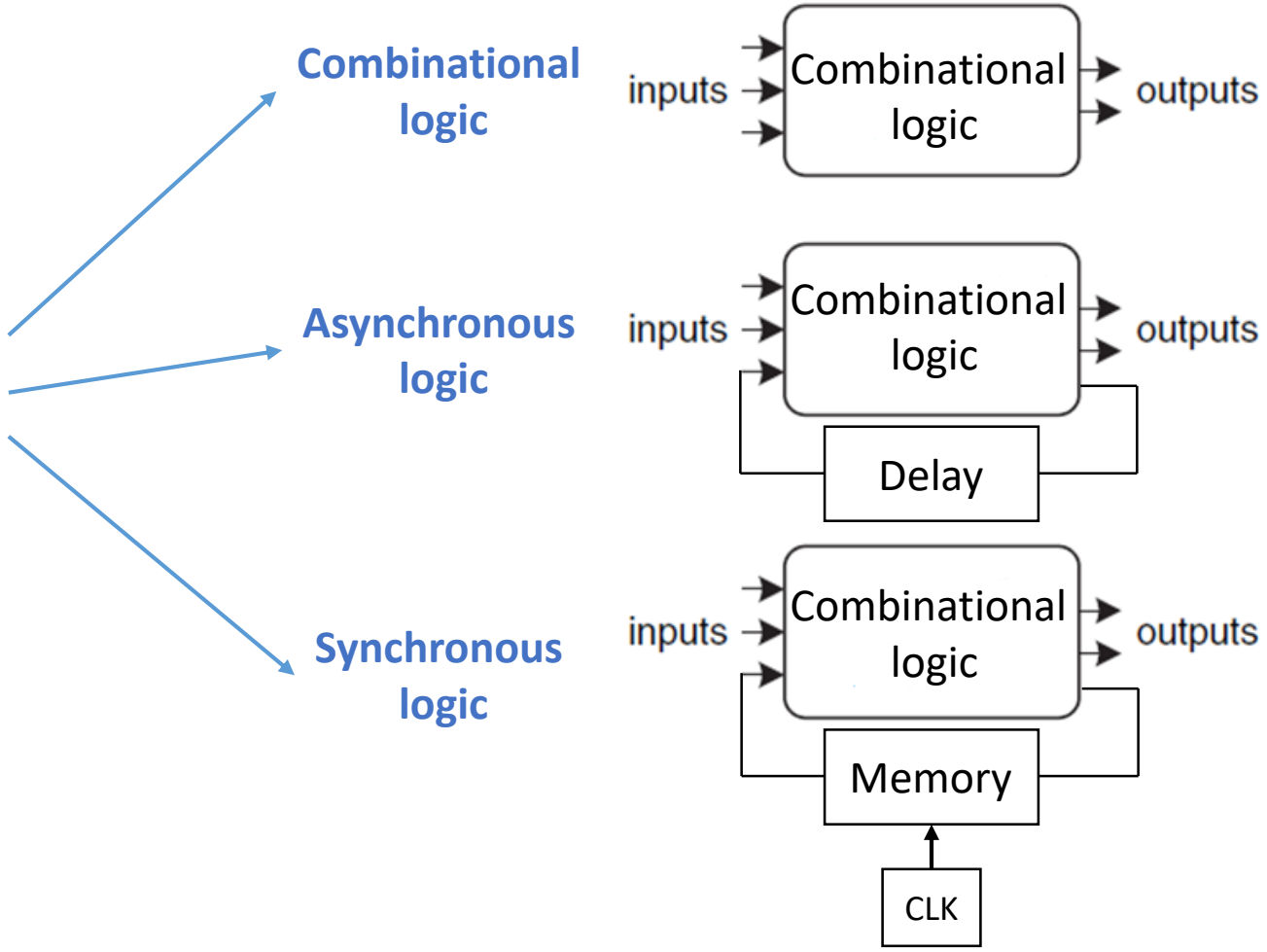




Transient Behaviour of Logic Circuits

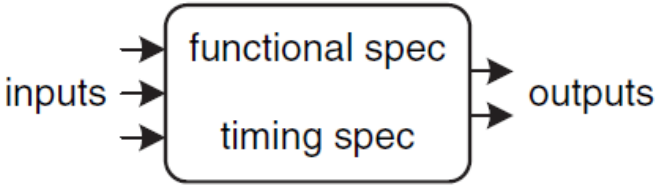
Timing Parameters :

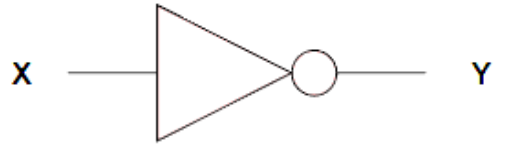
- Contamination delay: t_{cd}
- Propagation delay: t_{pd}



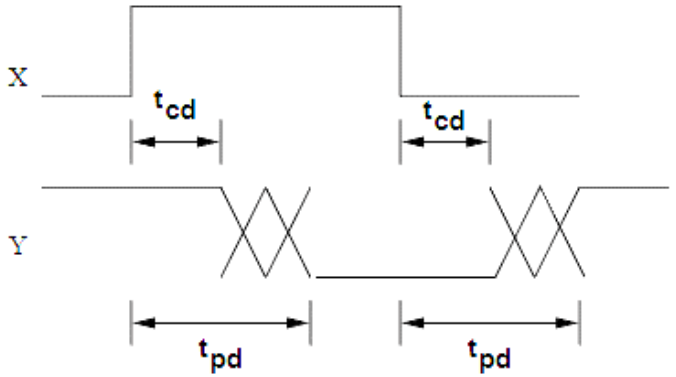
Timing Parameters for Combinational Logic

- When a binary value is applied at the input to a combinational circuit, the change at the circuit output is not instantaneous





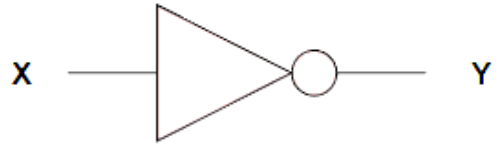
- Input-to-output delay in combinational circuits is expressed with two parameters, t_{pd} and t_{cd}



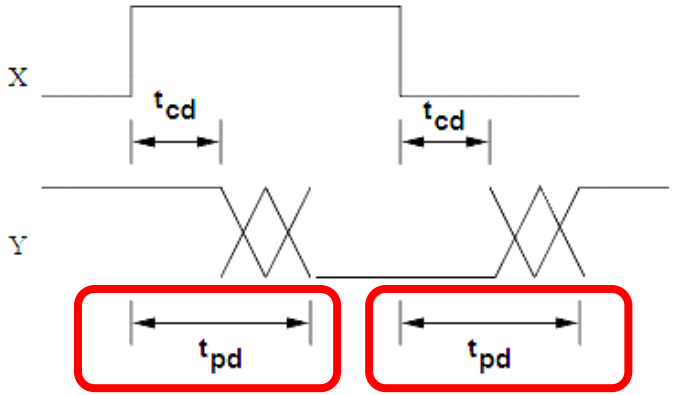
Timing Parameters for Combinational Logic

Propagation Delay t_{pd}

- amount of time needed for a change in a logic input to drive a permanent change at an output



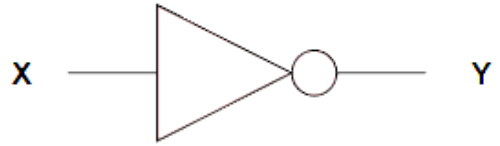
For combinational logic no output changes in response to an input change after t_{pd} time passed



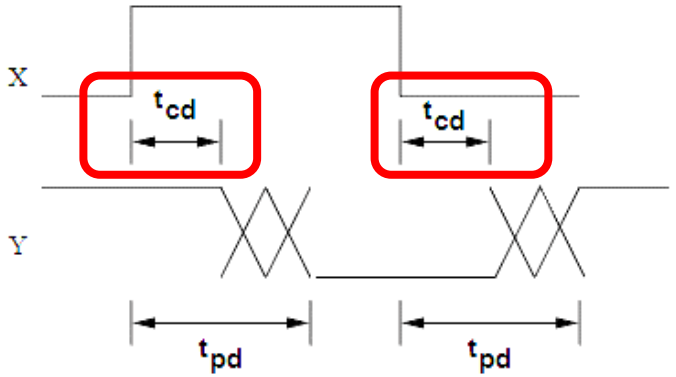
Timing Parameters for Combinational Logic

Contamination Delay t_{cd} :

- amount of time required for a change in a logic input to drive an initial change in an output

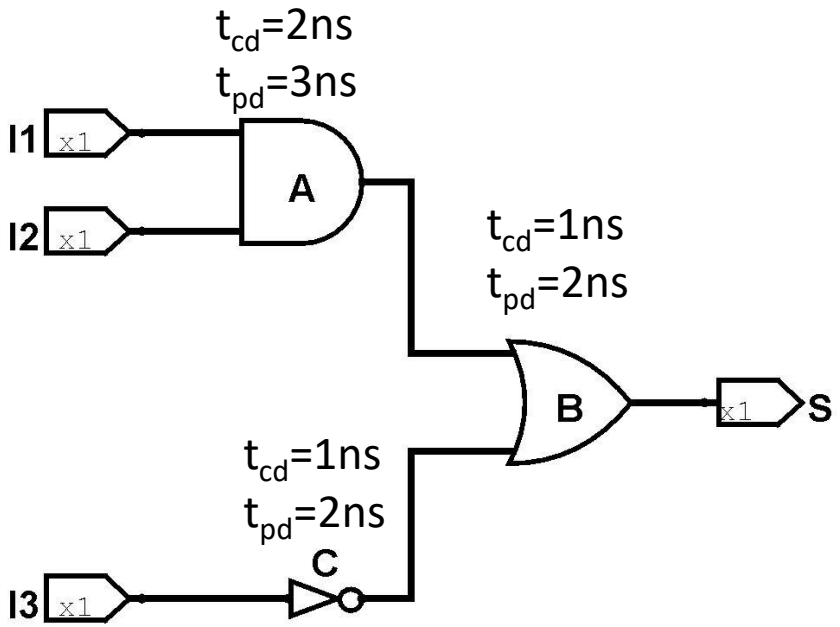


Combinational logic is guaranteed not to show any output change in response to an input change before t_{cd} time units have passed



Timing of Combinational Logic : exercise 1

1. Determine the total t_{pd} of such combinational circuit:
   - identify the longest path
   - **t_{pd} =**
2. Determine the total t_{cd} of such combinational circuit:
   - identify the shortest path
   - **t_{cd} =**



Timing Parameters for Synchronous Sequential Logic: Clock Signal

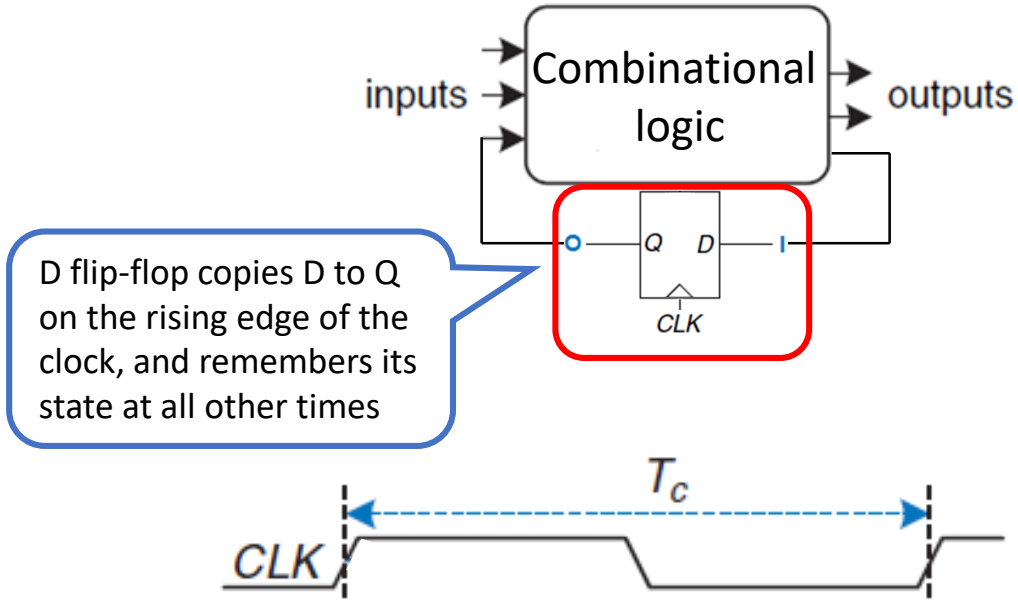
Synchronous circuits: A flip flop acts as memory element, such as the edge-trigged D flip flop controlled by the **clock signal**, CLK

Clock signal is a periodic square wave between logic high ("1") and logic low ("0")

- The amount of time between rising clock edges: **clock period**, $\mathbf{T_c}$
- The inverse of the clock period ($1/T_c$) is the **clock frequency**, $\mathbf{f_c}$.

$\mathbf{f_c}$ measures how often the data is transferred into edge-triggered flip flops:

↑clock frequency → data stored more quickly→ sequential circuit generates results at a faster rate



Timing Parameters for Synchronous Sequential Logic: D Flip-Flop

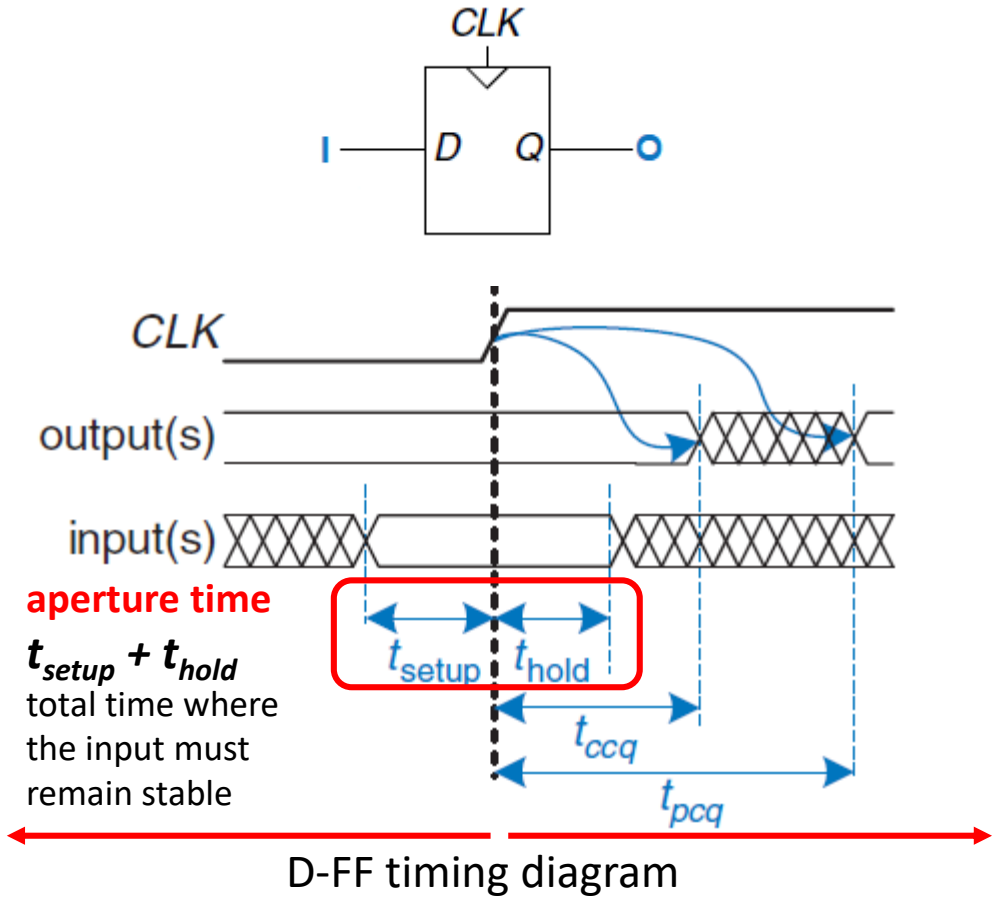
Requirement, flip-flop signals must be sampled while they are not changing:

Before clock rising edge:

- the input must have stabilized at least some **setup time**, t_{setup}

After clock rising edge:

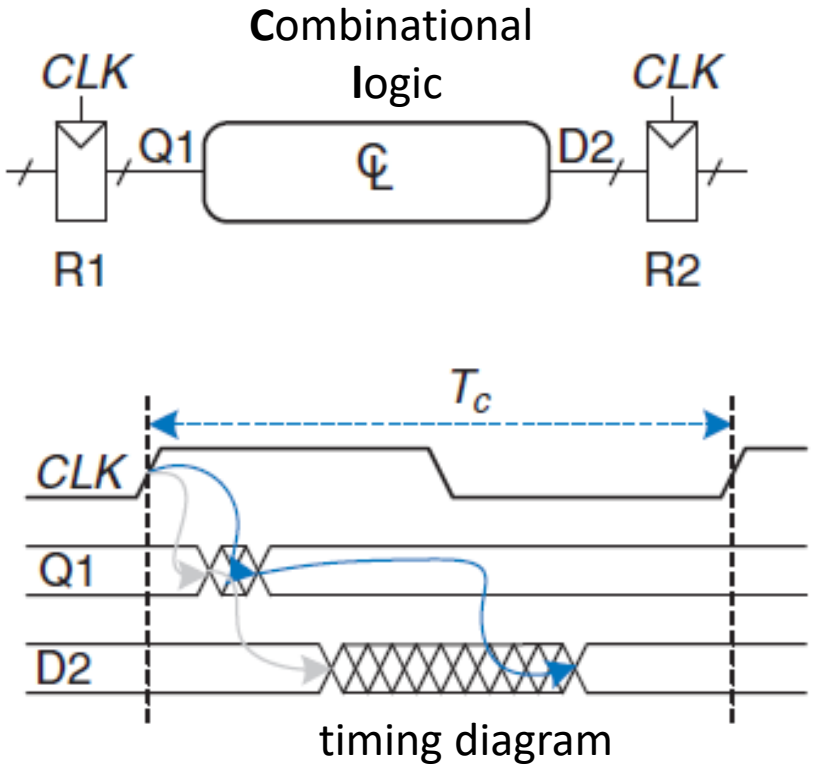
- the input must remain stable for at least some **hold time**, t_{hold}
- the output may start to change after the **clock-to-Q contamination delay**, t_{ccq}
- the output must definitely settle to the final value within the **clock to-Q propagation delay**, t_{pcq}



D-FF timing diagram

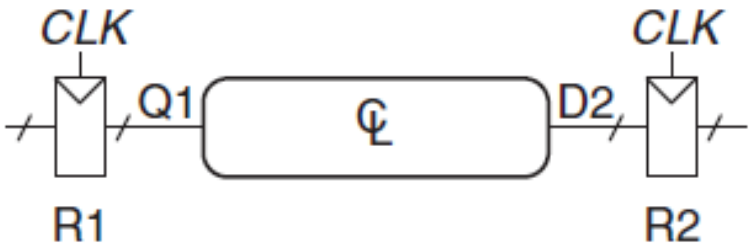
Timing of Synchronous Sequential Logic

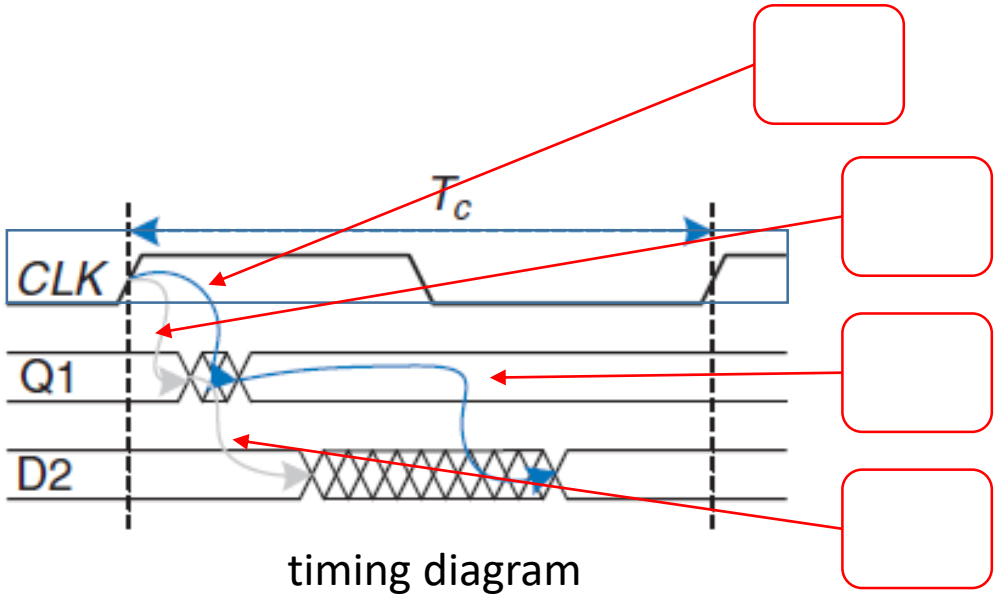
- Each calculation cycle starts immediately after an active clock edge and ends with the next one
- On the rising edge of the clock, register R1 generates the output Q1
- Q1 enters the combinational logic generating D2, the input to register R2



timing diagram

Timing of Synchronous Sequential Logic





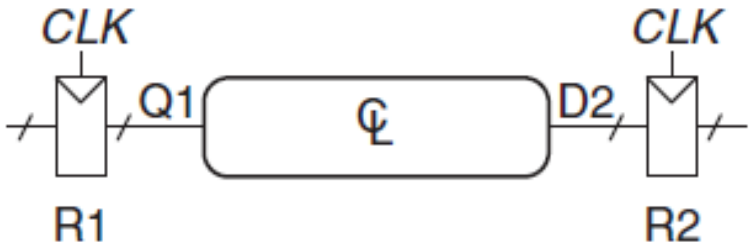
timing diagram

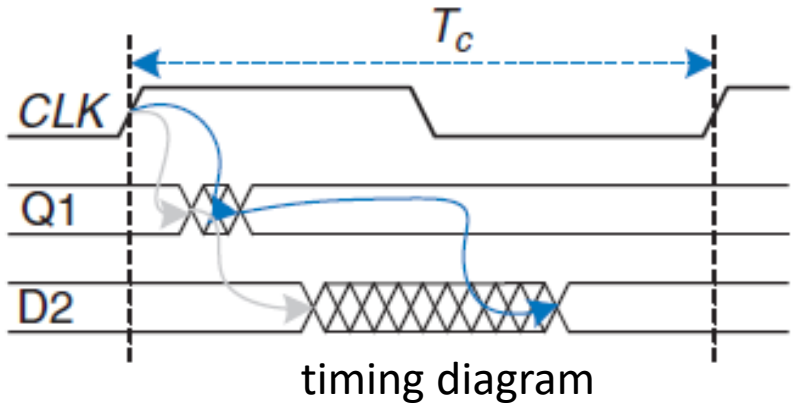
Term	Name
t_{pd}	Logic Propagation Delay
t_{cd}	Logic Contamination Delay
t_{pcq}	Latch/Flop Clock-to-Q Propagation Delay
t_{ccq}	Latch/Flop Clock-to-Q Contamination Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

Timing of Synchronous Sequential Logic

Gray arrows represent the **contamination delay** through R1 and the combinational logic

Blue arrows represent the **propagation delay** through R1 and the combinational logic



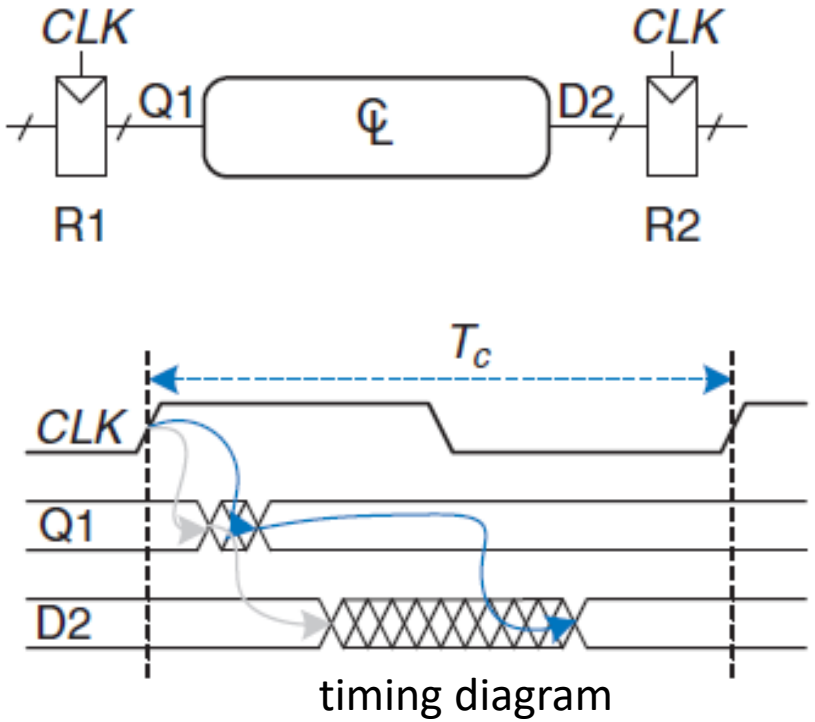


timing diagram

Timing Analysis of Synchronous Sequential Logic

Problem: given a generic path in a synchronous sequential circuit, calculate the **maximum clock frequency** that can be applied

Procedure: analyse the timing constraints with respect to the setup and hold time of the second register R2



timing diagram

Timing Analysis of Synchronous Sequential Logic: Setup Time Constraint

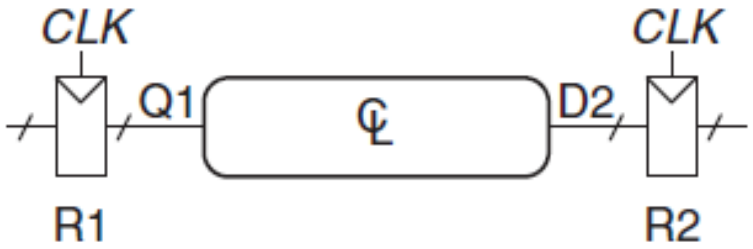
Analysis of the timing constraints with **respect to the setup time** of the second register R2

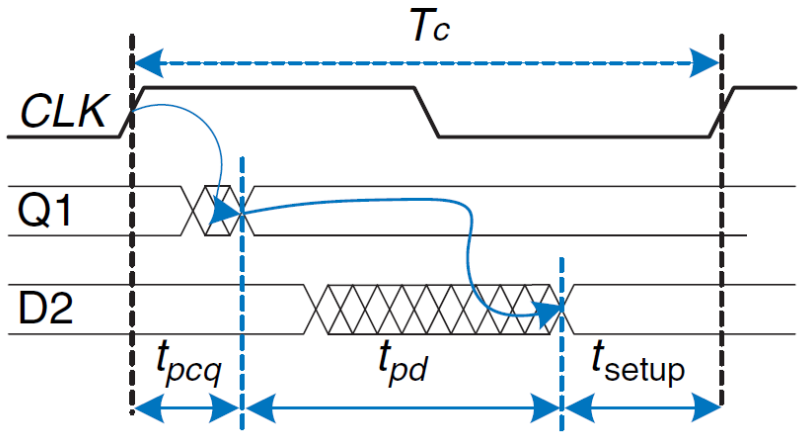
- use of timing diagram with only the maximum delay through the path
- to satisfy the setup time of R2, D2 must settle no later than the setup time before the next CLK edge:

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

If the equation is solved for the maximum propagation delay through the combinational logic:

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

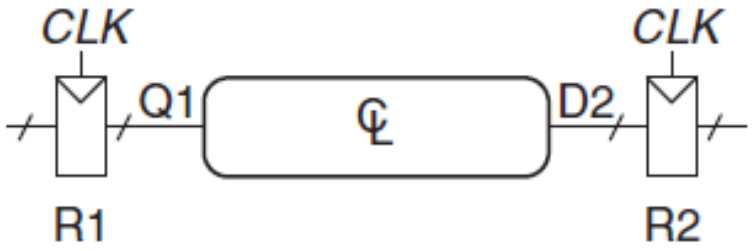


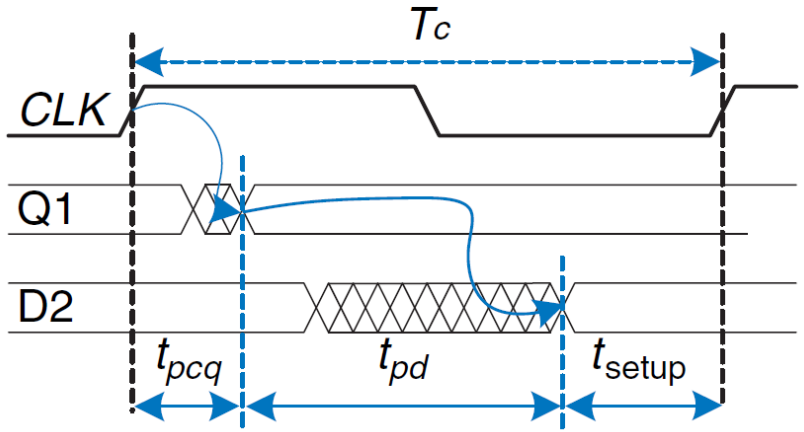


Timing Analysis of Synchronous Sequential Logic: Setup Time Constraint

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

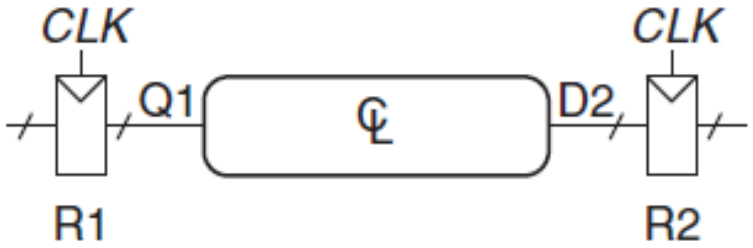
- **setup time constraint** or max-delay constraint, because it depends on the setup time
- it limits the maximum delay through combinational logic

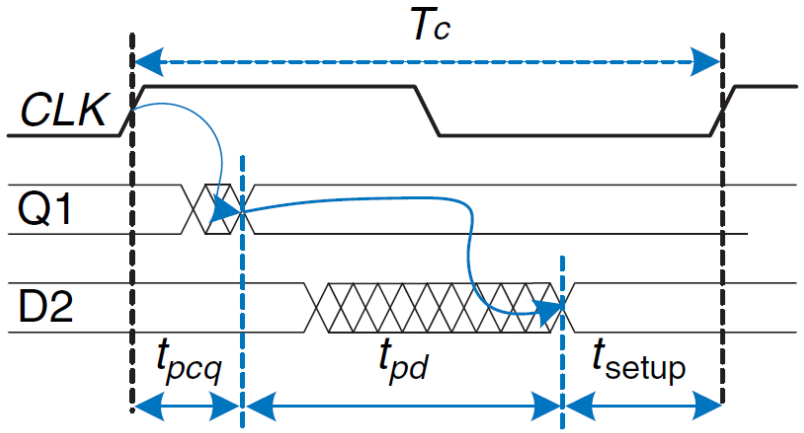




Timing Analysis of Synchronous Sequential Logic: Setup Time Constraint

- If the propagation delay $\mathbf{t_{pd}}$ through the combinational logic is too high, R2 may sample an incorrect result and the circuit will malfunction
- Solution: increase the clock period or redesign the combinational logic for a shorter propagation delay $\mathbf{t_{pd}}$





Timing Analysis of Synchronous Sequential Logic: Hold Time Constraint

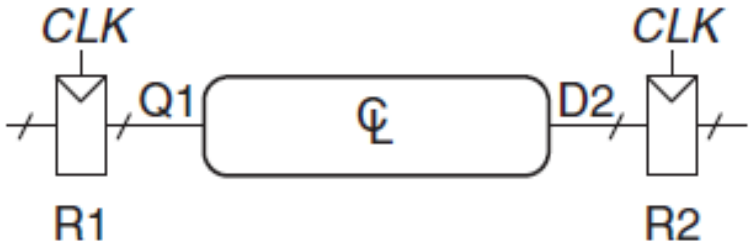
Analysis of the timing constraints with **respect to the hold time** of the second register R2

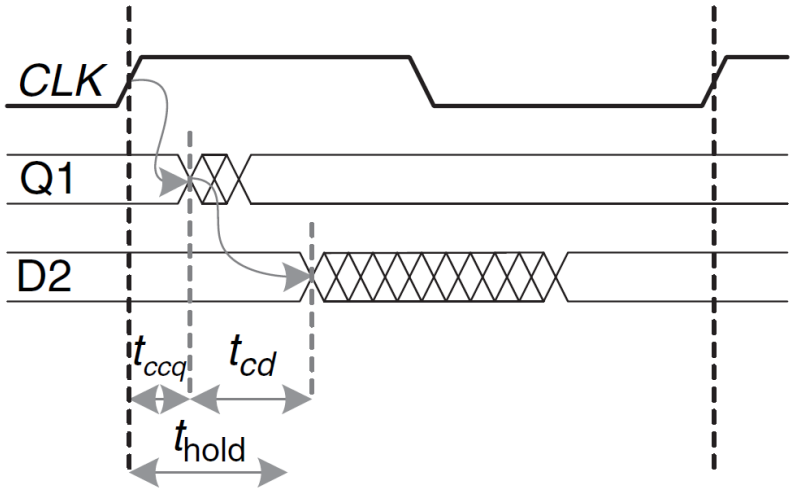
- use of timing diagram with only the minimum delay through the path
- to satisfy the hold time of R2, its input, D2, must not change until some time, t_{hold}, after the rising edge of CLK

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

- if the equation is solved for the minimum contamination delay through the combinational logic:

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

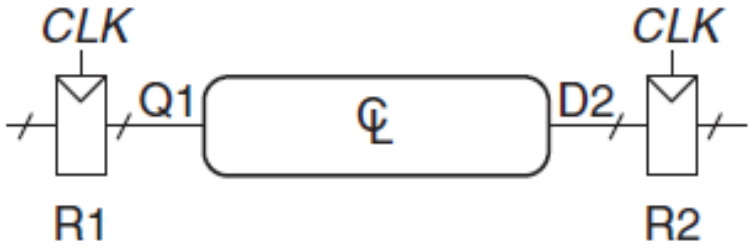


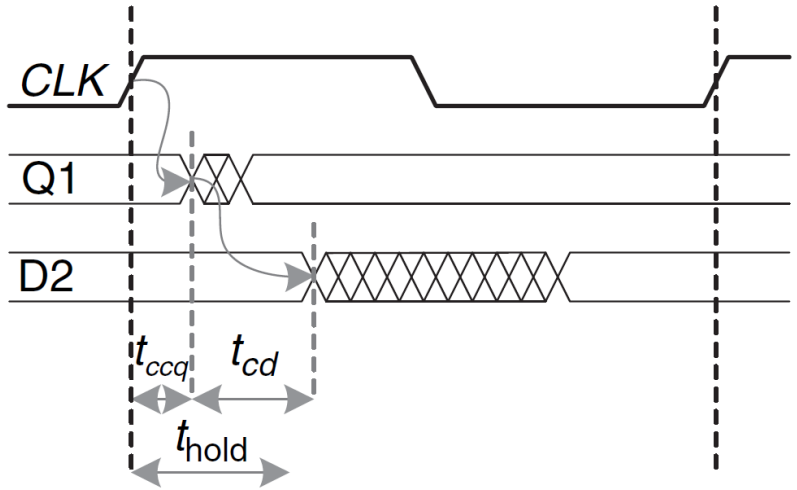


Timing Analysis of Synchronous Sequential Logic: Hold Time Constraint

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

- **hold time constraint** or min-delay constraint, because it depends on the hold time
- it limits the minimum delay through combinational logic



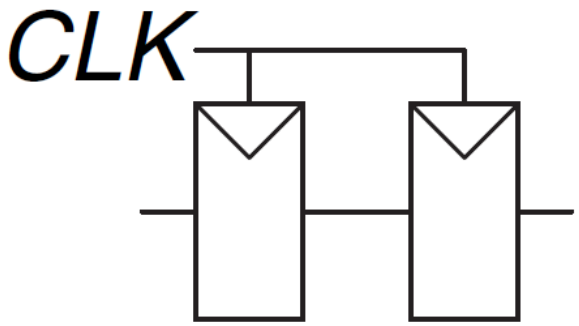


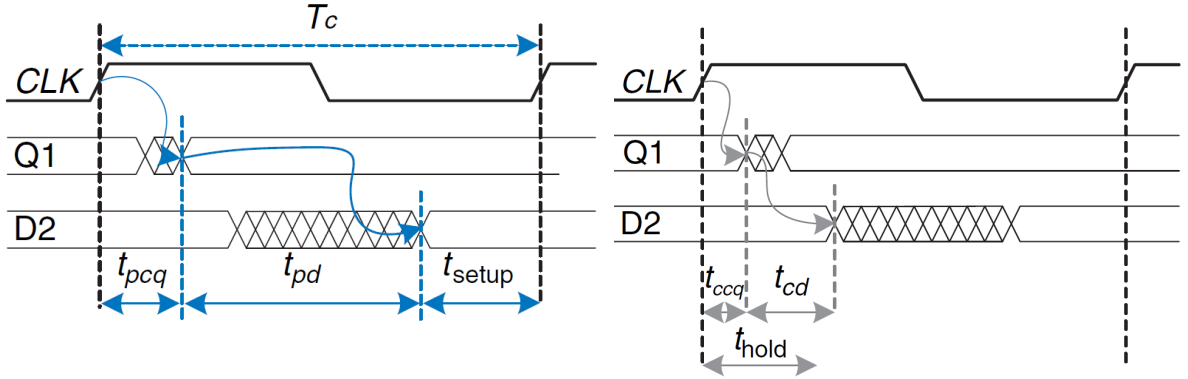
Timing Analysis of Synchronous Sequential Logic: Cascading flip-flops

Analysis of a *cascaded* arrangement of two D-FF, are there hold time problems?

In such a case there is no combinational logic between flip-flops

- hold time constraint:
  $t_{cd}=0 \rightarrow t_{hold} \leq t_{ccq}$
  $$t_{cd} \geq t_{\text{hold}} - t_{ccq}$$
- a reliable flip-flop must have a t_{hold} shorter than its t_{ccq}
- often, flip-flops are designed with $t_{hold} = 0$





Timing Analysis of Synchronous Sequential Logic: exercise 2

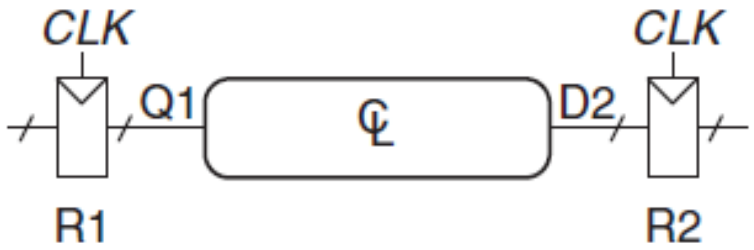
Timing analysis: determine the maximum clock frequency and whether any hold time violations could occur

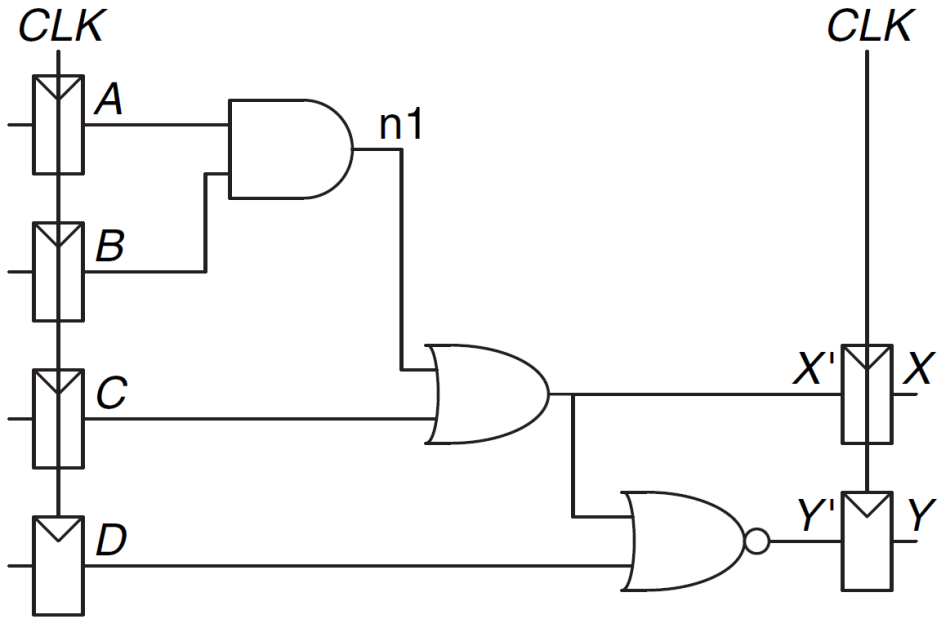
DFF:

- clock-to-Q contamination delay of 30 ps
- propagation delay of 80 ps
- setup time of 50 ps
- hold time of 60 ps

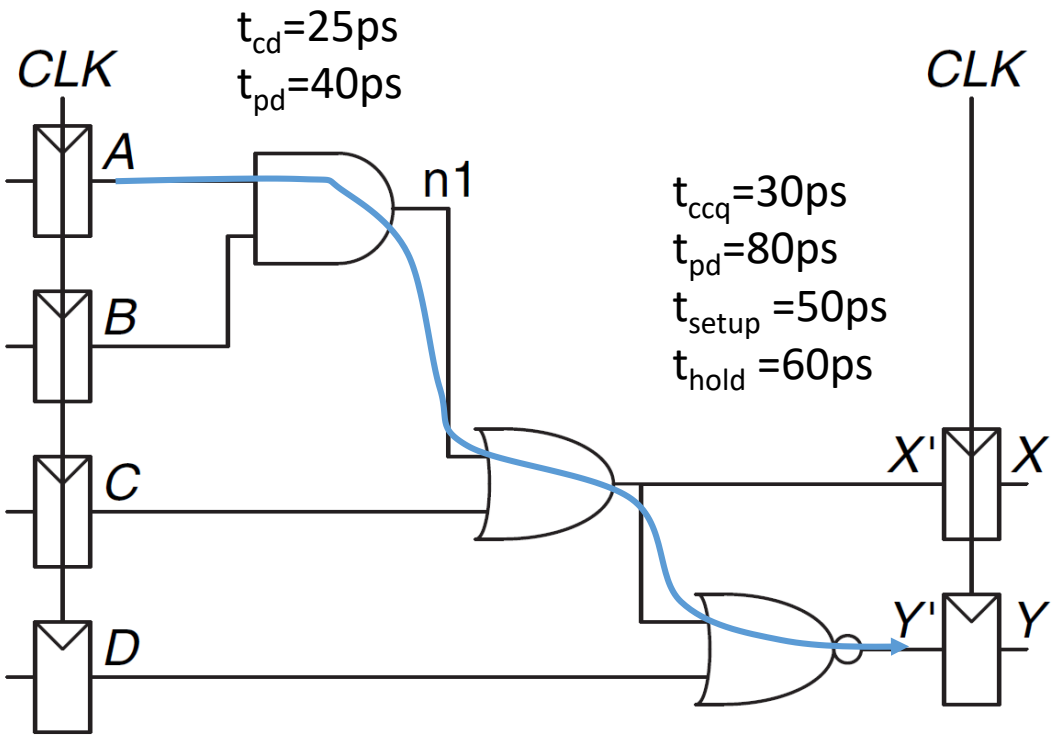
Logic gates:

- propagation delay of 40 ps
- contamination delay of 25 ps

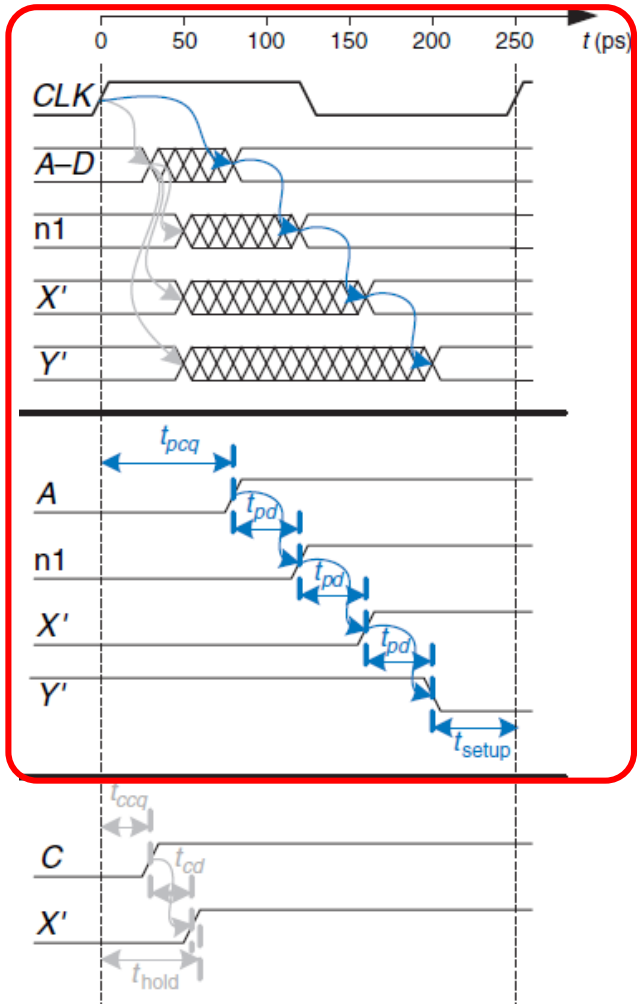




Timing Analysis of Synchronous Sequential Logic: exercise 2

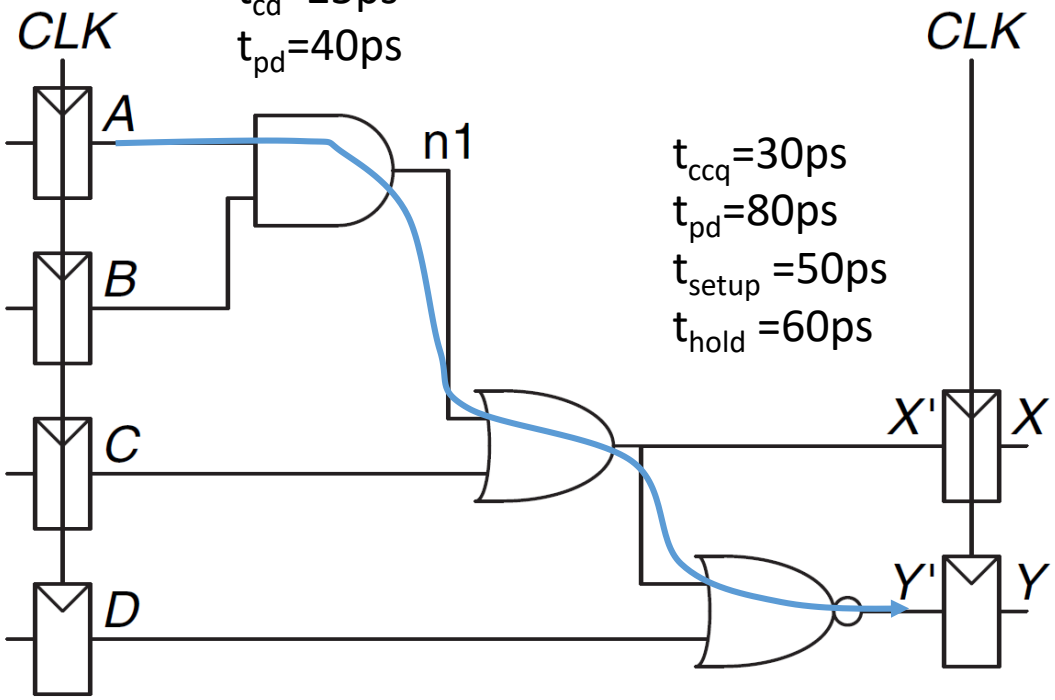


Setup Time Constraint



Hold Time Constraint

Timing Analysis of Synchronous Sequential Logic: exercise 2

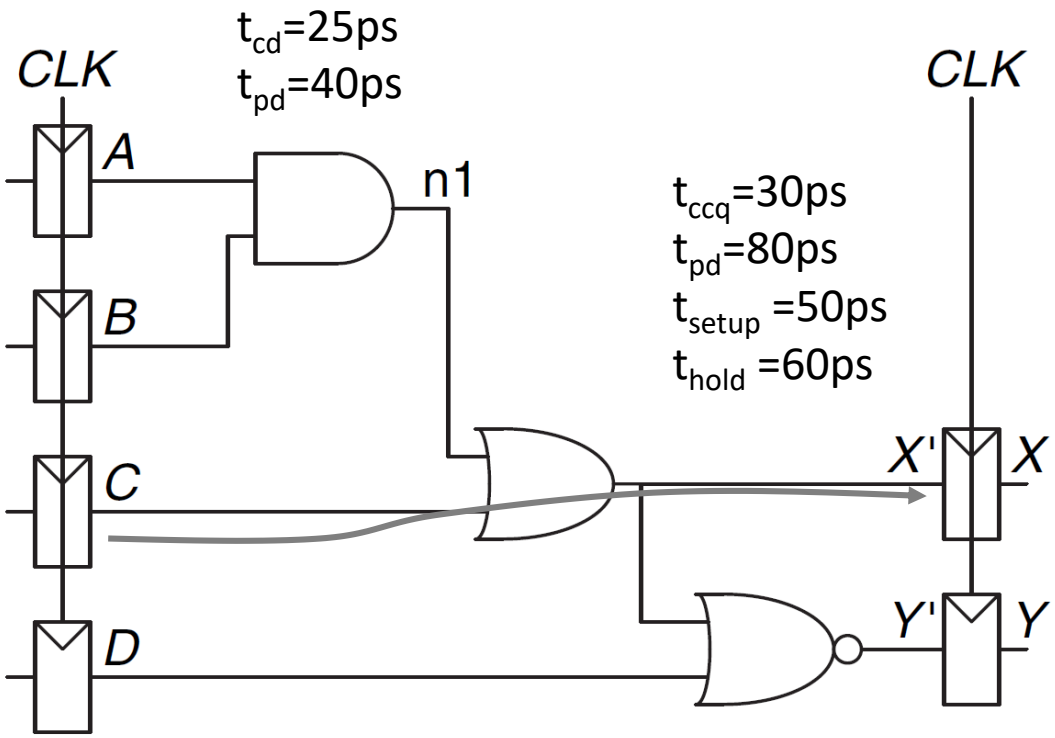


$$T_c \geq t_{pcq} + t_{pd} + t_{\text{setup}}$$

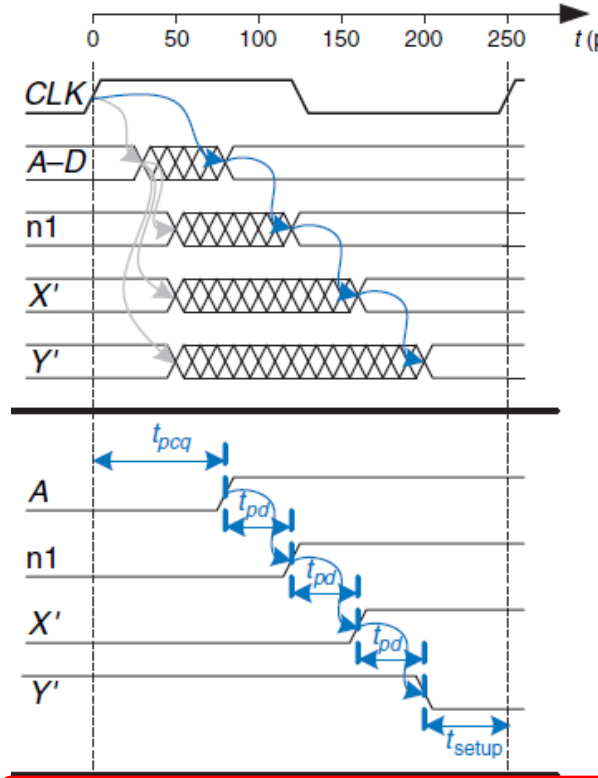
Maximum clock frequency

Setup Time Constraint: find the longest path

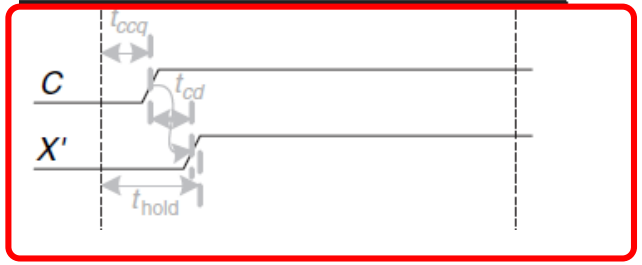
Timing Analysis of Synchronous Sequential Logic: exercise 2



Setup Time Constraint



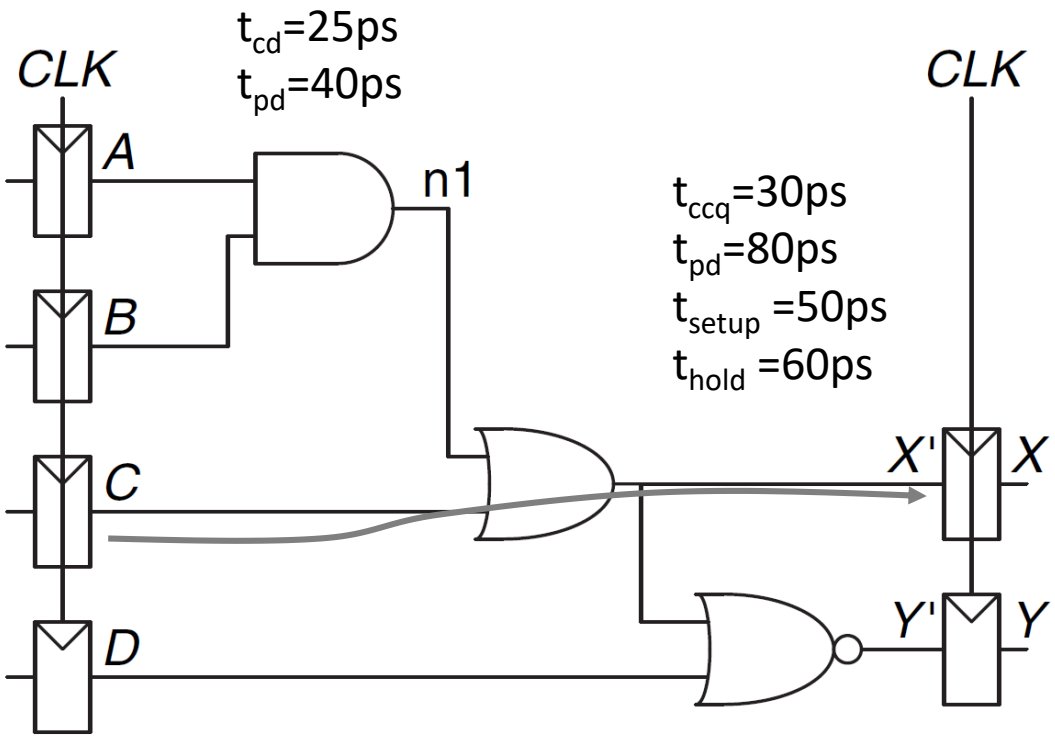
Hold Time Constraint



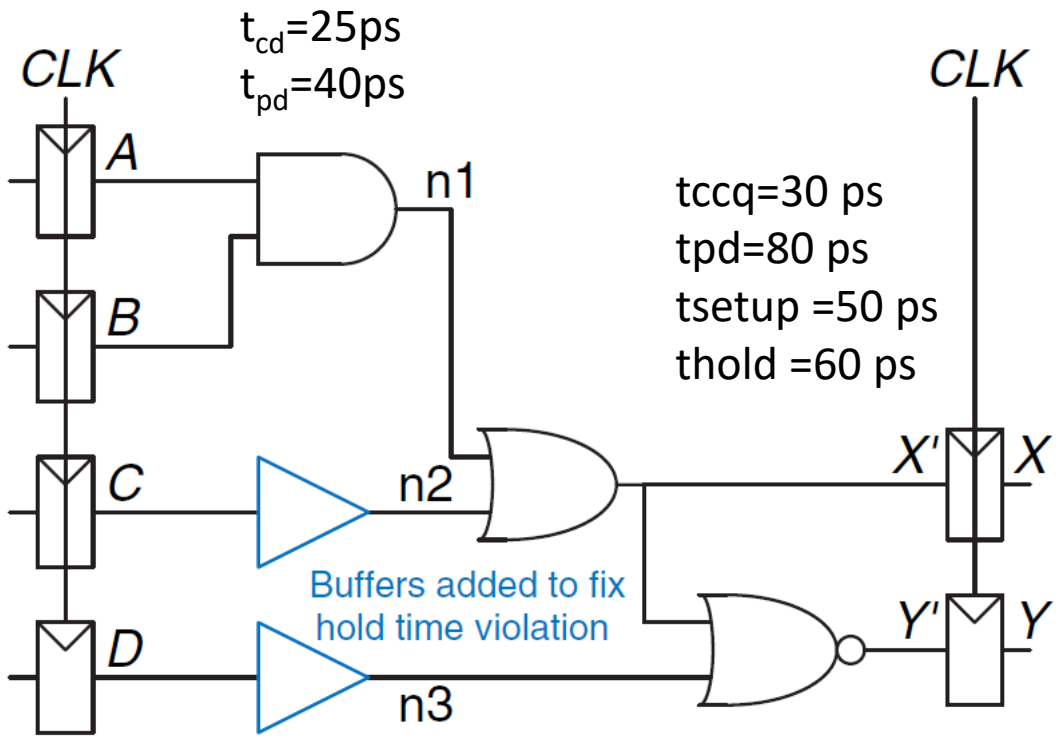
Timing Analysis of Synchronous Sequential Logic: exercise 2

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

Hold Time Constraint: find the shorter path



Timing Analysis of Synchronous Sequential Logic: exercise 2



Hold time violations fix: add buffers to slow down short paths

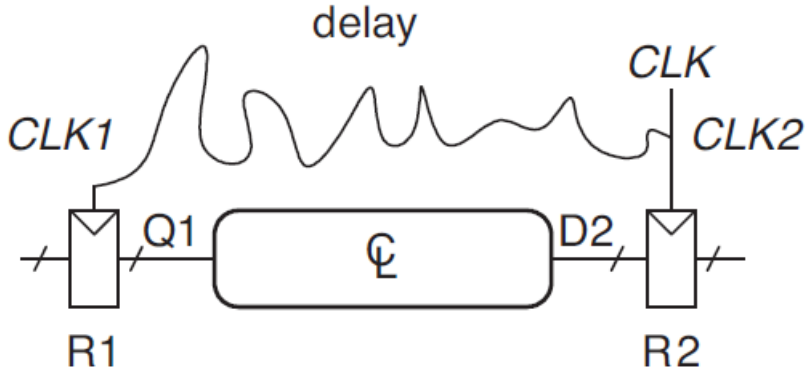
If the buffers have the same delays as other gates, **determine:**

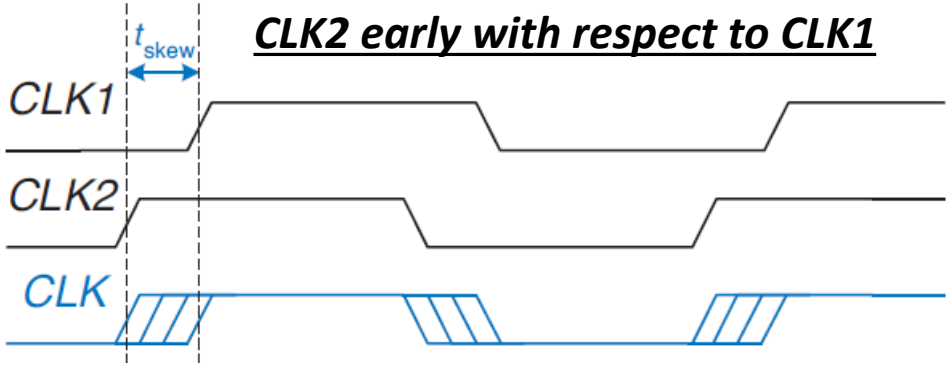
- the maximum clock frequency
- whether any hold time problems could occur

Clock Skew

What happens if the clock does not reach all registers at exactly the same time?

- Clock skew, t_{skew} : variation in clock edges
- When doing the timing analysis, consider the worst-case scenario to ensure the circuit works under all circumstances

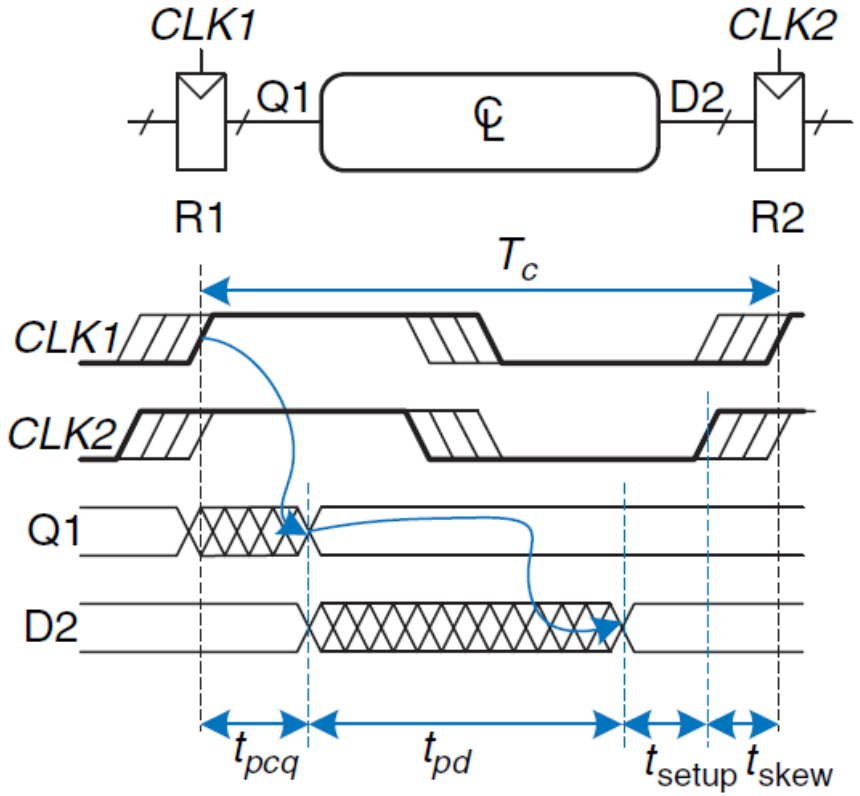




Timing Analysis with Clock Skew

Skew is added to the timing diagram:

- heavy clock line indicates the latest time at which the clock signal might reach any register
- hashed lines show that the clock might arrive upto t_{skew} earlier



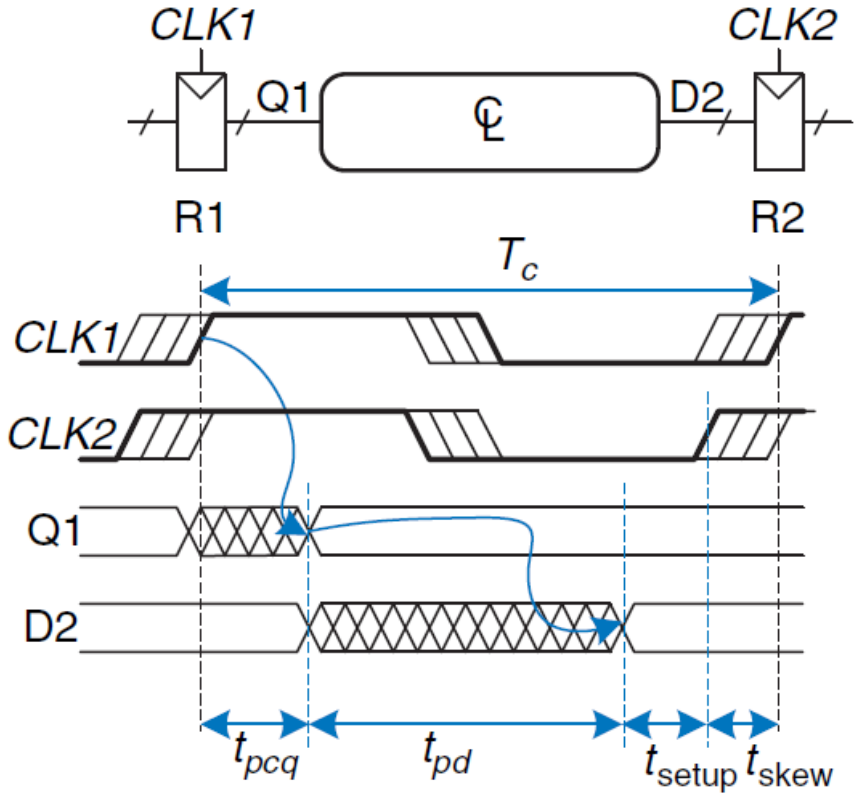
Timing Analysis with Clock Skew: Setup Time Constraint

Consider R1 receives the latest skewed clock and R2 receives the earliest:

- data has a limited time to propagate between registers and must setup before R2 samples it

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

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



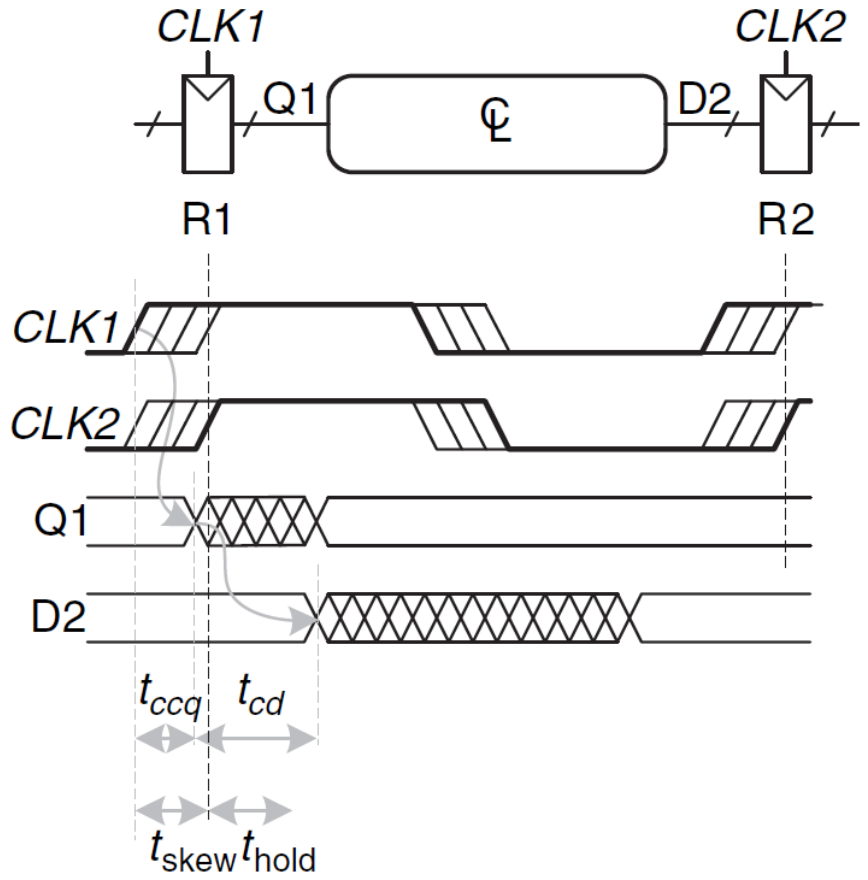
Timing Analysis with Clock Skew: Hold Time Constraint

In the worst case, R1 receives an early skewed clock, CLK1, and R2 receives a late skewed clock, CLK2

- data flows through register and combinatorial logic, but should not arrive until a hold time after the late clock

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

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



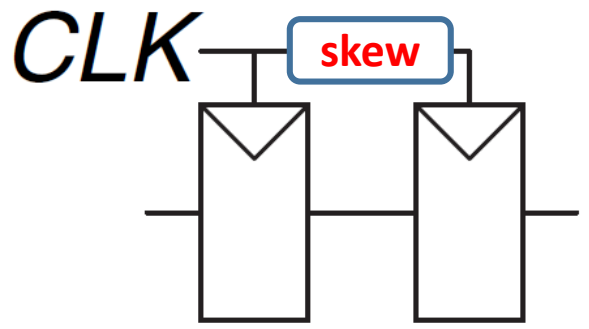
Timing Analysis with Clock Skew: Cascading flip-flops

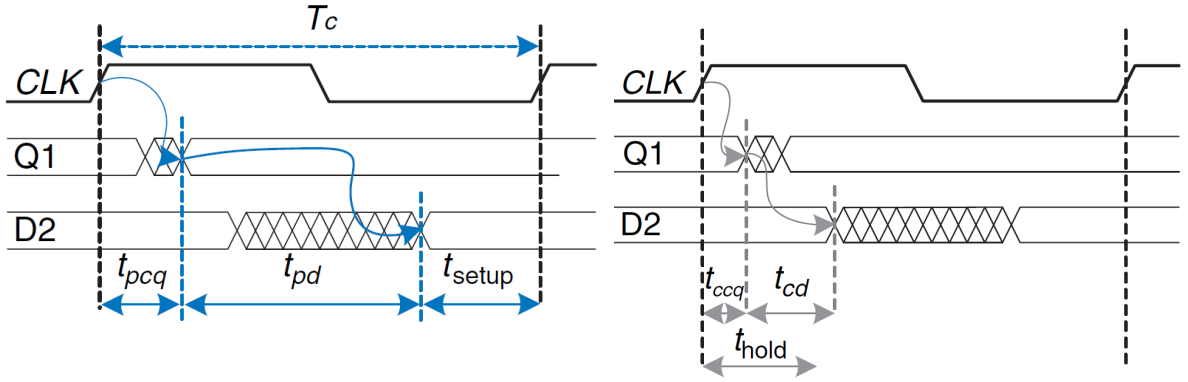
Analysis of a *cascaded* arrangement of two D-FF with clock skew, are there hold time problems?

- Even if $t_{hold} = 0$, a pair of back-to-back flip-flops can violate the equation:

$$t_{ccq} \geq t_{skew}$$

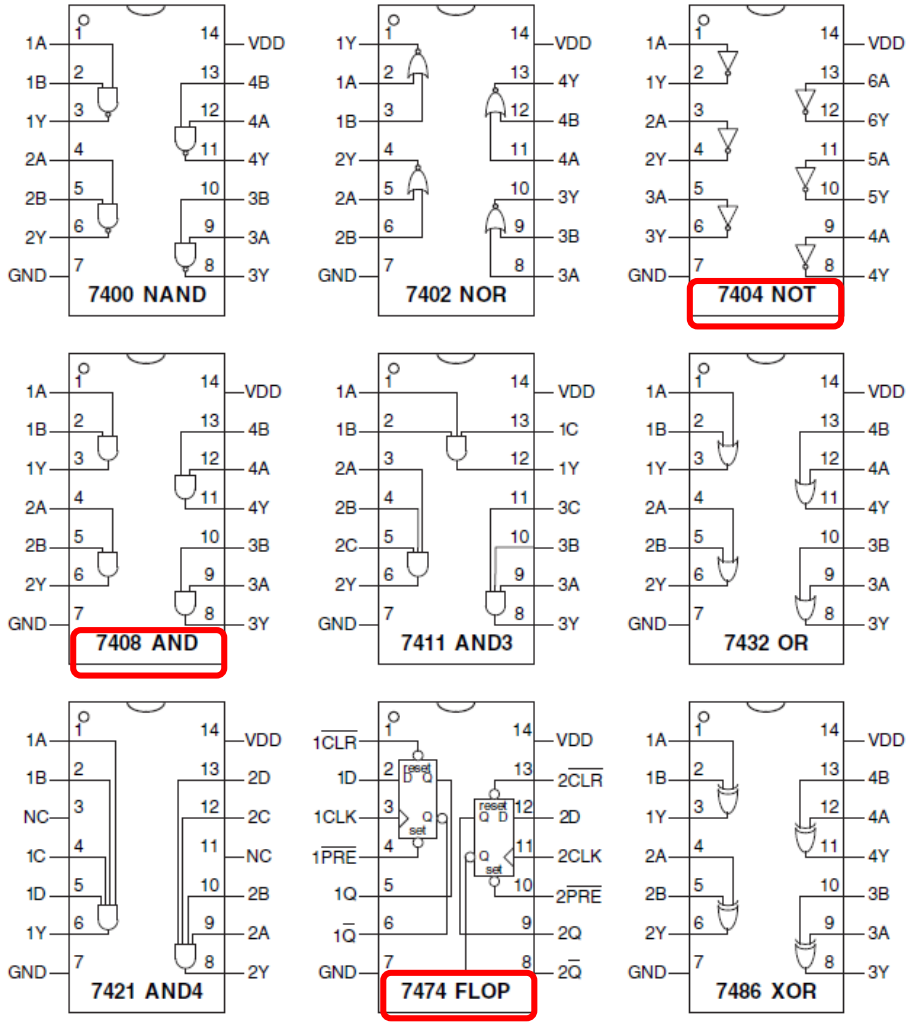
- If $t_{ccq} < t_{skew}$, flip-flops can intentionally designed to be particularly slow (large tccq)





Timing Analysis of Synchronous Sequential Logic: 74xx LOGIC

- 74xx-series logic: chips, each containing few logic gates, for example:
- 7404: six NOT gates
- 7408: four AND gates
- 7474: two flip-flops



Timing Analysis of Synchronous Sequential Logic: 74xx LOGIC

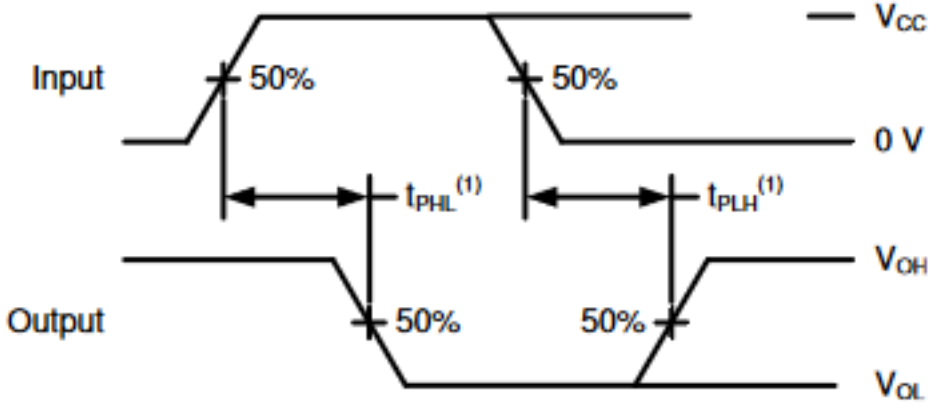
SNx4HC04 Hex Inverters

- The propagation delay, t_{pd}, is measured from when the input passes through $0.5V_{CC}$ to when the output passes through $0.5V_{CC}$
- If VCC is nominally 2V and the chip drives a capacitance of less than 50 pF, the propagation delay will not exceed 95ns (and typically will be much faster)
- The maximum between t_{PLH} and t_{PHL} is used for t_{pd}

6.8 Switching Characteristics - 54

over operating free-air temperature range; typical values measured at TA = 25°C (unless otherwise noted).

PARAMETER		FROM	TO	V_{CC}	Operating free-air temperature (T_A)									UNIT
					25°C			–40°C to 85°C			–55°C to 125°C			
					MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
t_{pd}	Propagation delay	A	Y	2 V		45	95			120			125	ns
				4.5 V		9	19			24			29	
				6 V		8	16			20			25	

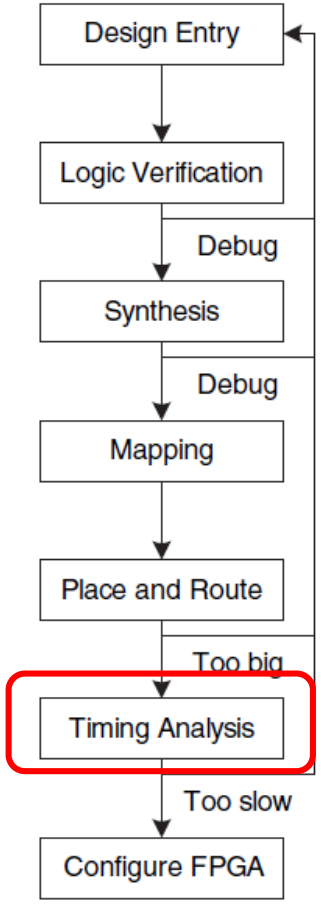


Timing Analysis of Synchronous Sequential Logic: FPGA

FPGA, Field Programmable Gate Array: array of reconfigurable gates

- Using software programming tools, both combinational and sequential logic logic designs can be implemented on FPGAs
- Modern FPGAs integrate other features such as multipliers, high-speed I/Os, analog-to-digital converters, large RAM arrays and processors

FPGA Design Flow



Design Entry: design specified with a hardware description language (HDL)

Logic Verification: The design is then simulated to verify that the logic is correct

Logic synthesis: converts the HDL into Boolean functions

Mapping: FPGA tool maps the functions onto the Logic Elements (LEs) of a specific chip

Place and route tool determines which functions go in which lookup tables and how they are wired together

Timing Analysis: compares the timing constraints against the actual circuit delays and reports any errors

Configuration: a file is generated specifying the contents of all the (Logic Elements)LEs and the programming of all the wires on the FPGA

Timing Analysis of Synchronous Sequential Logic: FPGA, exercise 3

A design is implemented as a FSM (Finite State Machine) on a FPGA having the timing characteristics shown beside

1. If the FSM has to run at 100 MHz, what is the maximum number of Logic Elements on the critical path?
2. What is the fastest speed at which her FSM could possibly run?

- FPGA Timings

name	value (ps)
t_{pcq}	199
t_{setup}	76
t_{hold}	0
t_{pd} (per LE)	381
t_{wire} (between LEs)	246

Timing Analysis of Synchronous Sequential Logic: FPGA, exercise 3

1. If the FSM has to run at 100 MHz, what is the maximum number of LEs on the critical path?

At 100 MHz, Tc, is 10 ns

- minimum combinational propagation delay, t_{pd} , at this cycle time:

- FPGA Timings

name	value (ps)
t_{pcq}	199
t_{setup}	76
t_{hold}	0
t_{pd} (per LE)	381
t_{wire} (between LEs)	246

Timing Analysis of Synchronous Sequential Logic: FPGA, exercise 3

2. What is the fastest speed at which her FSM could possibly run?

- The fastest speed is when using a single LE for the next state logic

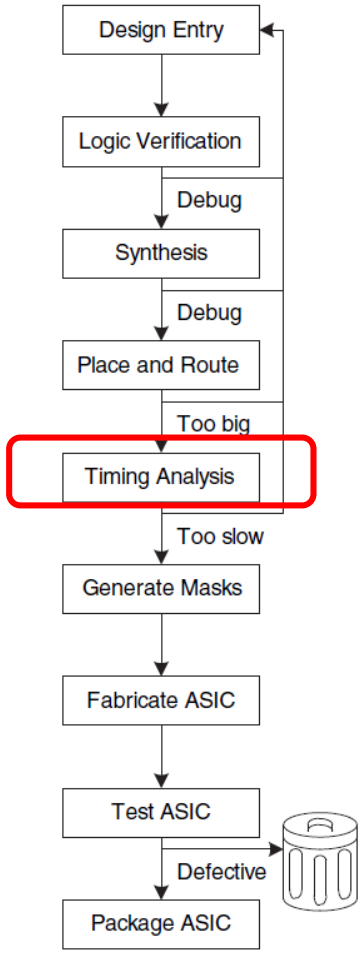
- FPGA Timings

name	value (ps)
t_{pcq}	199
t_{setup}	76
t_{hold}	0
t_{pd} (per LE)	381
t_{wire} (between LEs)	246

Timing Analysis of Synchronous Sequential Logic: ASIC

- Application-specific integrated circuits (ASICs) are chips designed for a particular purpose
- Examples of ASICSs: graphics accelerators, network interface chips, and cell phone chips
- ASIC are hardwired for a specific function
  - several times faster than an FPGA
  - occupies an order of magnitude less chip area than an FPGA with the same function
  - higher cost for production

Asic Design Flow



Design Entry: design specified with a hardware description language (HDL)

Logic Verification: The design is then simulated to verify that the logic is correct

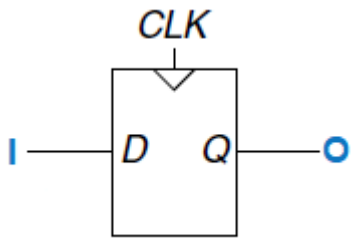
Logic synthesis: converts the HDL into a netlist consisting of logic gates and connections between the gates;

Place and route: the gates in this netlist are placed, and the wires are routed between gates.

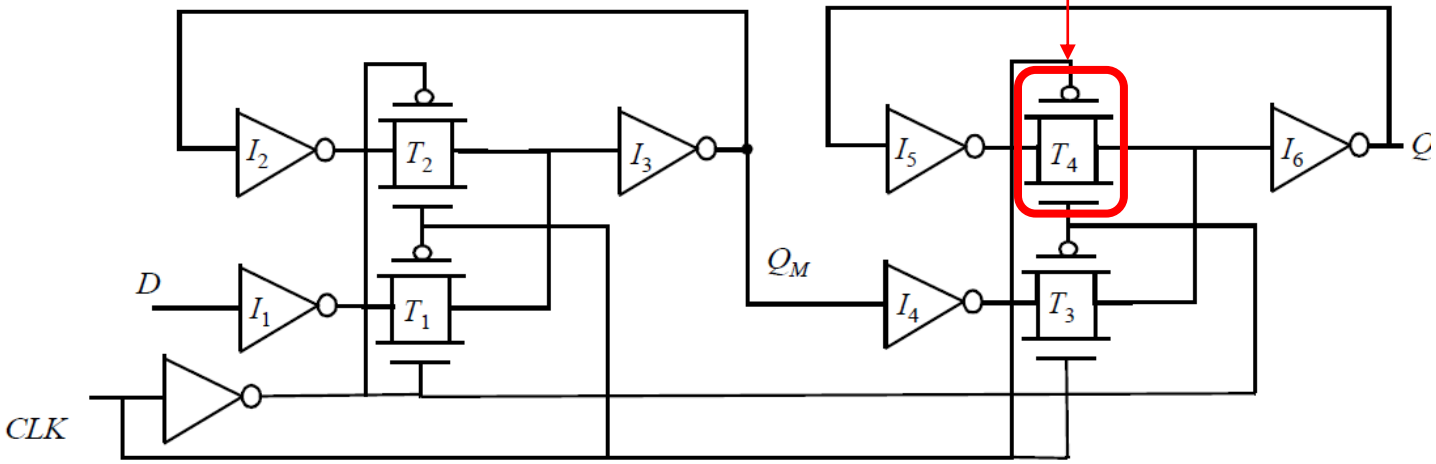
Timing Analysis: compares the timing constraints against the actual circuit delays and reports any errors

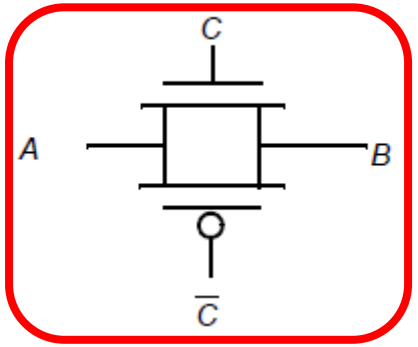
Mask generation: masks are generated and used to fabricate the ASIC

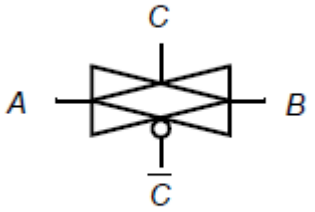
Timing Analysis of Multiplexer-based Master-Slave Registers



CMOS Transmission gate







Symbolic representation

CMOS Transmission Gate (TG):

control input C is a logic one, C='1'

- both transistors are on and short circuit (low resistance path) between A and B

control input C is a logic zero,, C='0'

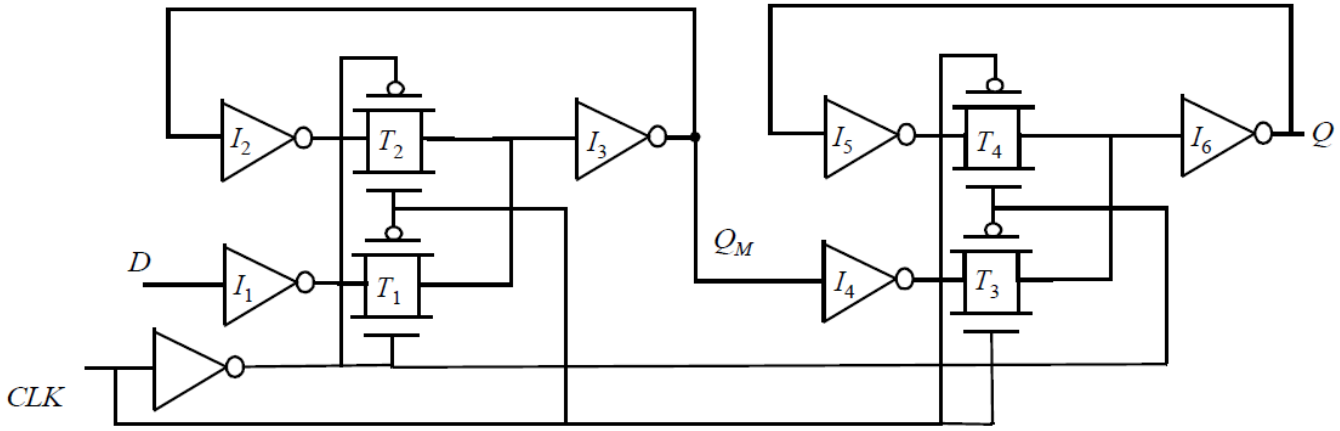
- both transistors are off and open circuit (high resistance path) between A and B

Timing Analysis of Multiplexer-based Master-Slave Registers

Study of factors that affect timing parameters: set-up time, hold time and propagation delay

Assumptions:

- propagation delay of inverters is t_{pd_inv}
- propagation delay of the transmission gate is t_{pd_tx}
- contamination delay is 0
- inverter delay to derive CLKb from CLK has a delay equal to 0



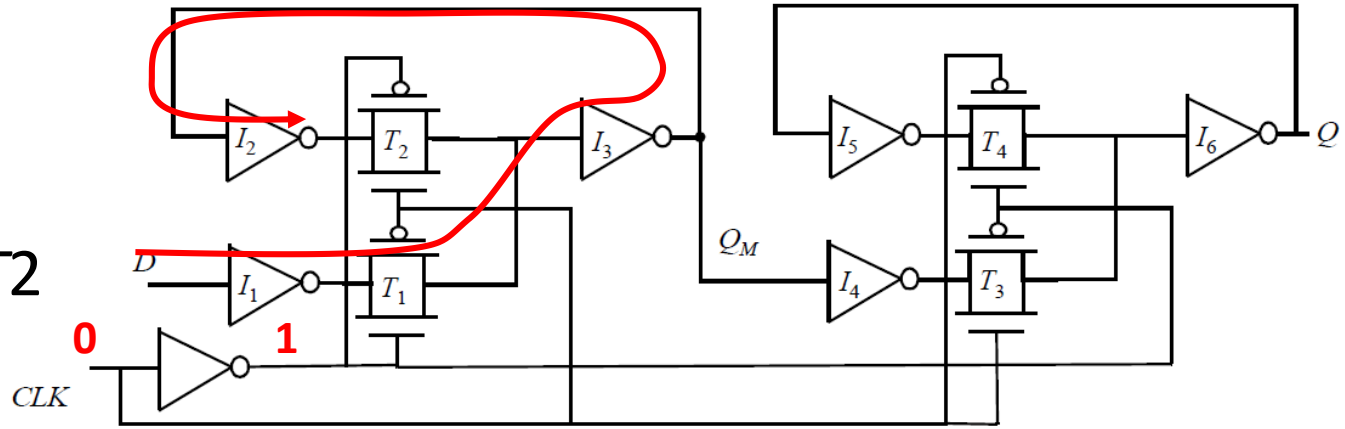
Timing Analysis of Multiplexer-based Master-Slave Registers

Set-up time: the input D must have stabilized at least some *set-up time* before the rising edge of the clock

The input *D* has to propagate through *I*1, *T*1, *I*3 and *I*2 before the rising edge of the clock to ensure that the node voltages on both terminals of the transmission gate *T*2 are at the same value

Otherwise, it is possible for the cross-coupled pair *I*2 and *I*3 to settle to an incorrect value

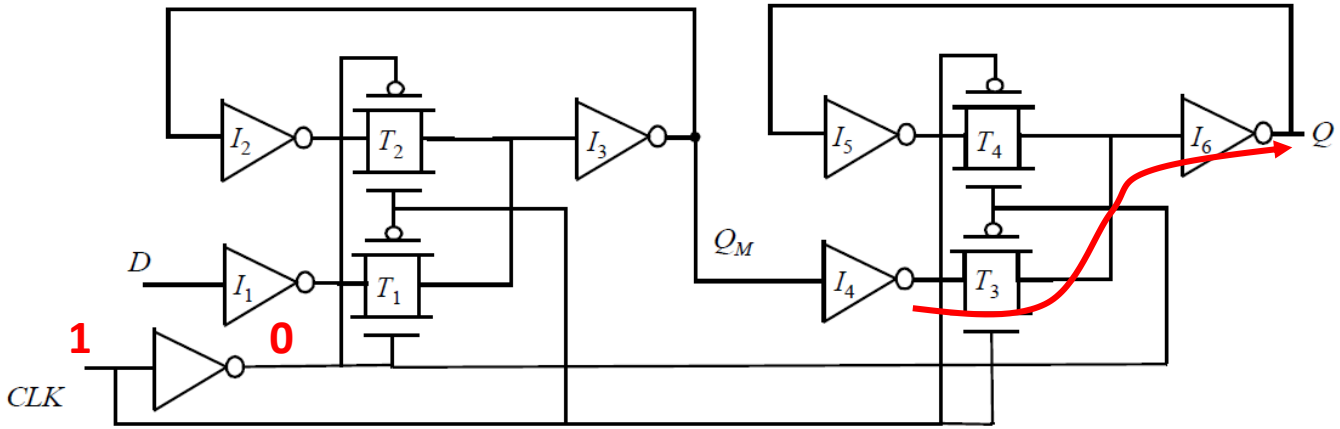
$t_{setup} = 3 * t_{pd\_inv} + t_{pd\_tx}$



Timing Analysis of Multiplexer-based Master-Slave Registers

- *propagation delay* : time for the value of *QM* to propagate to the output *Q*
- the output of *I*4 is valid before the rising edge of clock
- delay t_{cq} is the delay through *T*3 and *I*6

$t_{pcq} = t_{pd\_tx} + t_{pd\_inv}$

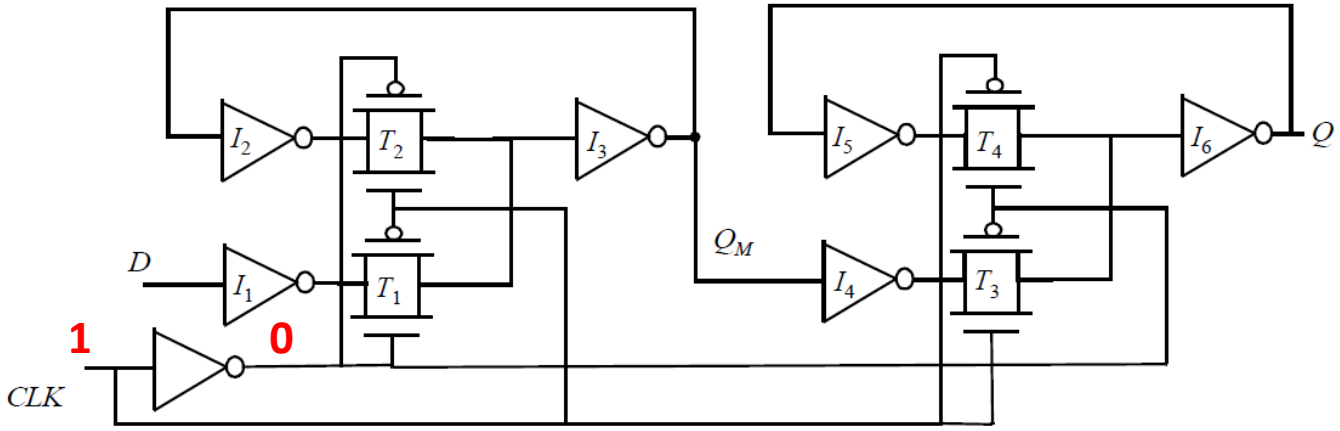


Timing Analysis of Multiplexer-based Master-Slave Registers

- *hold time* : the input must remain stable for at least some **hold time** after the rising edge of the clock
- the transmission gate *T*1 turns off when clock goes high
- any changes in the *D*-input

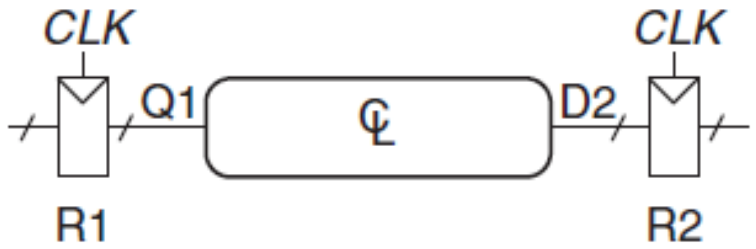
after clock rising are not seen by the input

t_{hold} =0



Timing of Synchronous Sequential Logic: Summary

- The maximum delay constraint limits the number of consecutive gates on the critical path of a high-speed circuits
- Setup and hold time constraints dictate the maximum and minimum delays of the combinational logic between flip-flops in sequential circuits
- Modern flip-flops are usually designed so that the minimum delay through the combinational logic is 0
- Clock skew effectively increases both the setup time and the hold time.



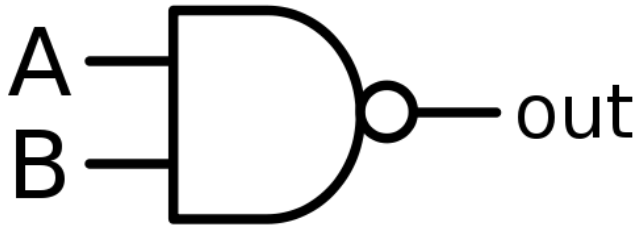
Term	Name
t_{pd}	Logic Propagation Delay
t_{cd}	Logic Contamination Delay
t_{pcq}	Latch/Flop Clock-to-Q Propagation Delay
t_{ccq}	Latch/Flop Clock-to-Q Contamination Delay
t_{setup}	Latch/Flop Setup Time
t_{hold}	Latch/Flop Hold Time

Exercises

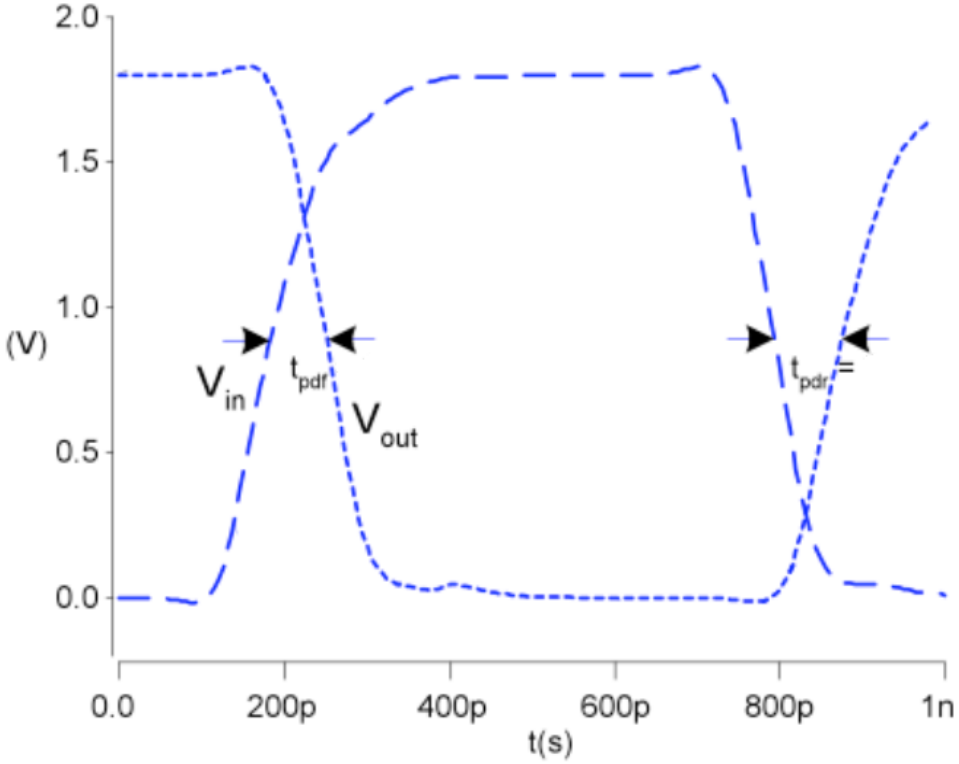
Timing Analysis using LTspice

Timing Analysis using LTspice : exercise 1

Obtain the t_{pd} time of a NAND2 gate using LTspice



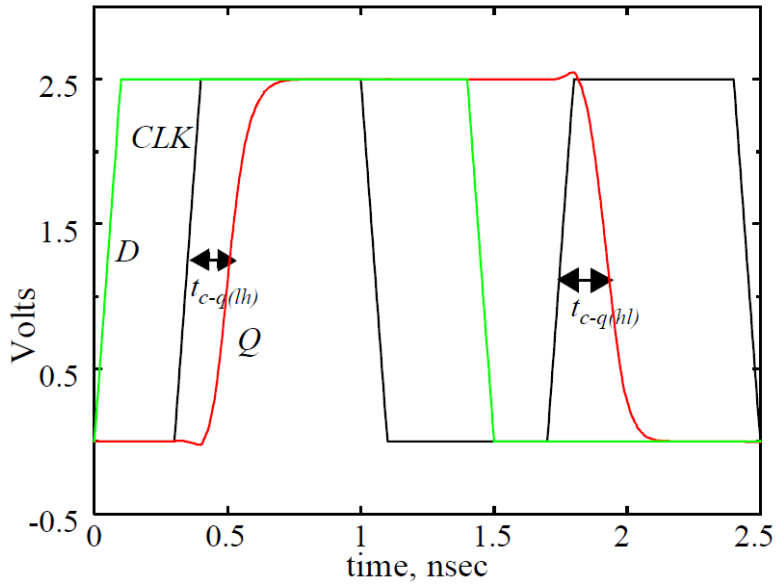
Input Data Pattern	Delay
A = *B*= 0→1	
A = 1, *B*= 0→1	
A= 0→1, *B* = 1	
A=*B*=1→0	
A=1, *B* = 1→0	
A= 1→0, *B* = 1	



Timing Analysis using LTspice : exercise 2

Obtain the propagation delay of the D-FF using LTspice, assume that:

- a load of a single inverter is connected at each D-FF output
- the D-FF is initially in reset state



Simulation of *propagation delay*.

Timing Analysis using LTspice : exercise 3

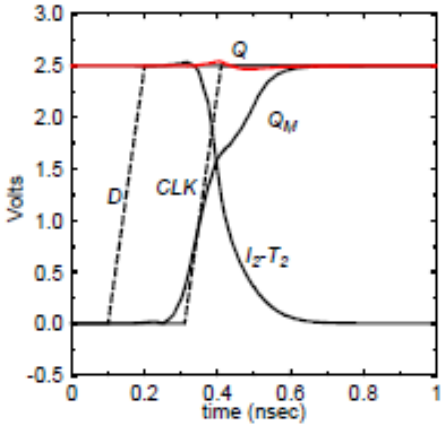
Obtain the set-up time, t_{setup}, of the D-FF using LTspice :

- progressively skew the input D with respect to the clock edge until the circuit fails

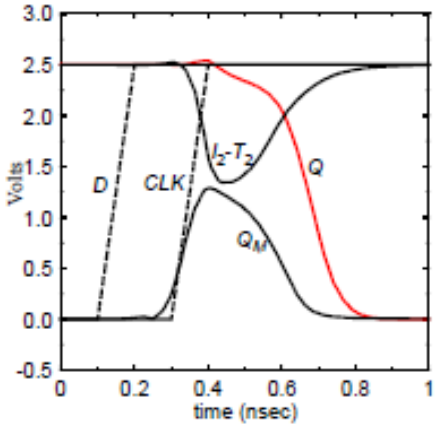
Assume that:

- a load of a single inverter is connected at each D-FF output

No set-up time violation!



Set-up time violation!



Timing Analysis using LTspice : exercise 4

Obtain the hold time, t_{hold}, of the D-FF using LTspice :

- progressively skew the *D* input edge relative to the clock signal till the circuit stop functioning (for this design, the *hold time* is 0 - i.e., the inputs can be changed on the clock edge)

References

- David Harris and Sarah Harris. 2007. Digital Design and Computer Architecture. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.
- Neil Weste and David Harris. 2010. CMOS VLSI Design: A Circuits and Systems Perspective (4th. ed.). Addison-Wesley Publishing Company, USA.
- Rabaey, J. M.; Chandrakasan, A. & Nikolic, B. (2004), Digital integrated circuits- A design perspectiv*e* , Prentice Hall .
- R. Jacob Baker. 2010. CMOS Circuit Design, Layout, and Simulation (3rd. ed.). Wiley-IEEE Press.