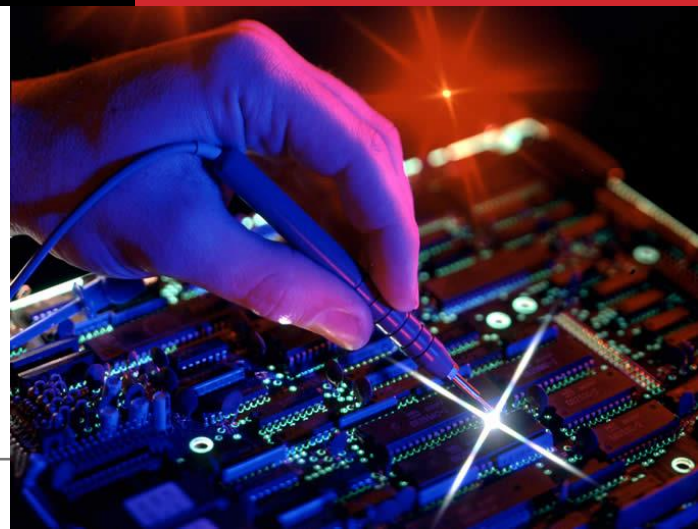
Computer Architecture & Microprocessor System

DIGITAL CIRCUIT TIMING

Dennis A. N. Gookyi

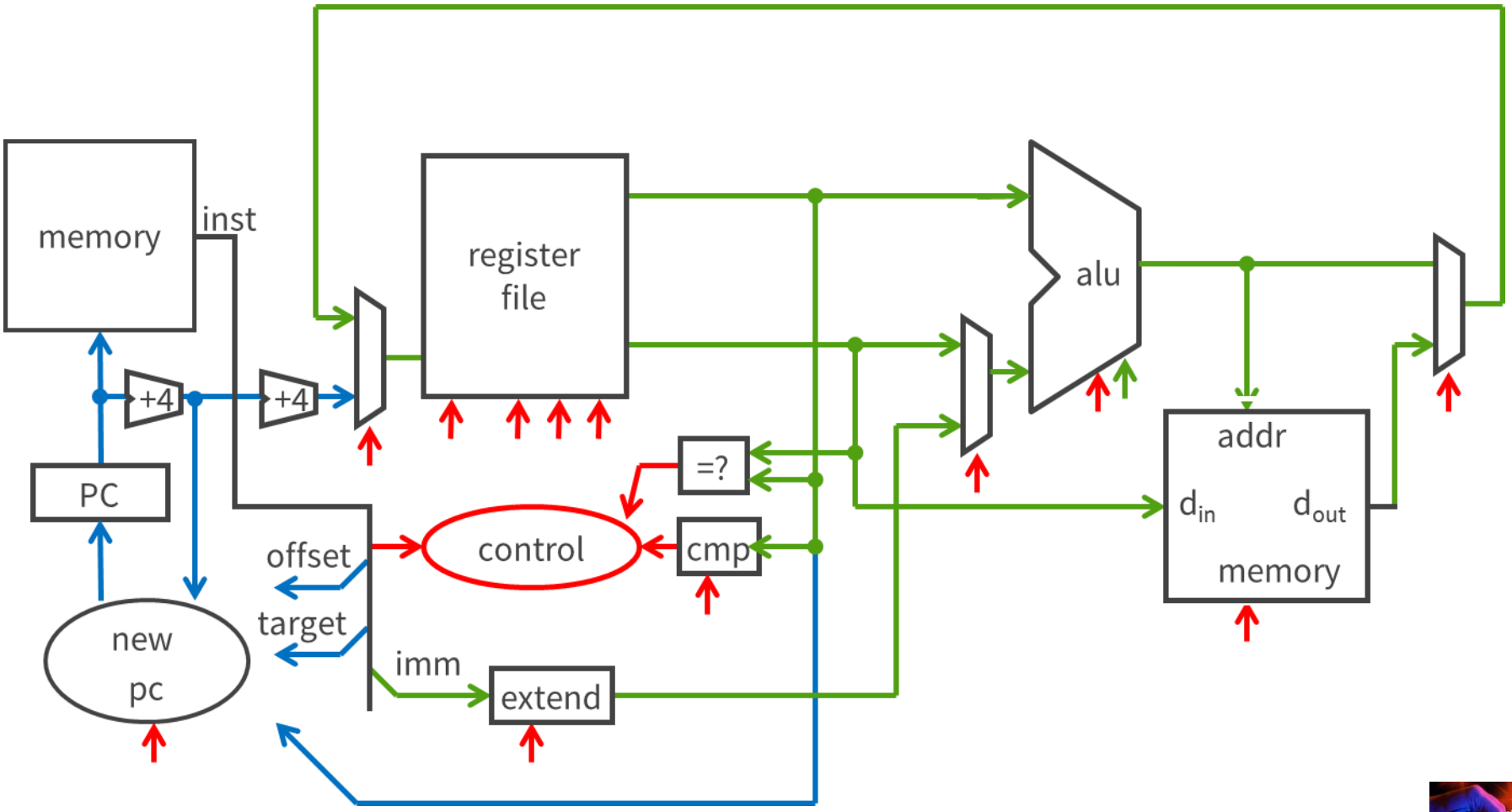


CONTENTS

- DIGITAL CIRCUIT TIMING

BIG PICTURE: BUILDING A PROCESSOR

- Single cycle processor



INTRODUCTION

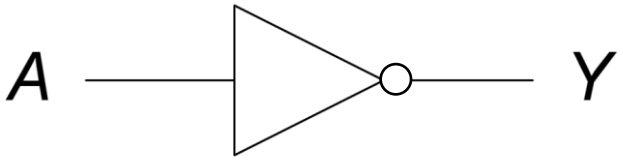
- Until now, we investigated logical functionality

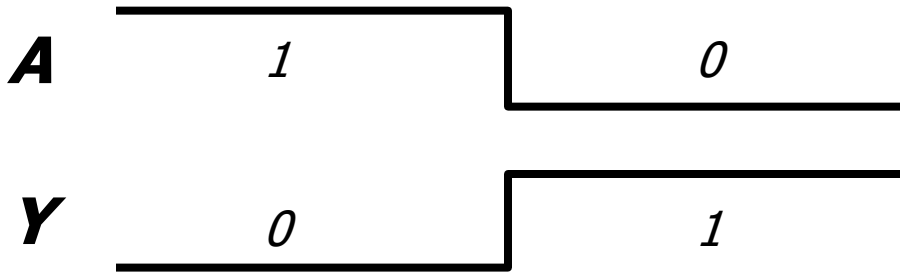
- What about timing?
  - How fast is a circuit?
  - How can we make a circuit faster?
  - What happens if we run a circuit too fast?

- A design that is logically correct can still fail because of real-world implementation issues

COMBINATIONAL CIRCUIT TIMING

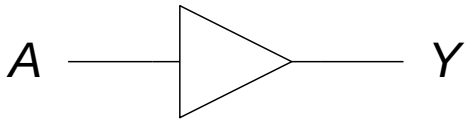
- "Digital logic" is a convenient abstraction
  - Output changes immediately with the input

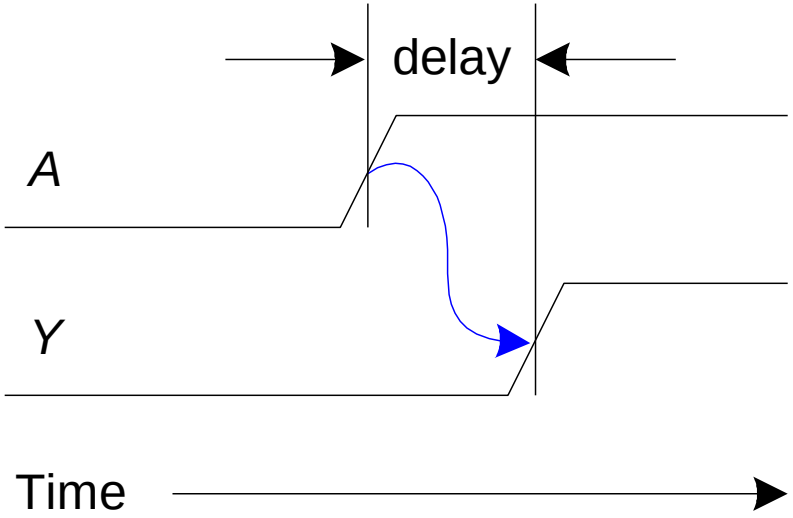




COMBINATIONAL CIRCUIT TIMING

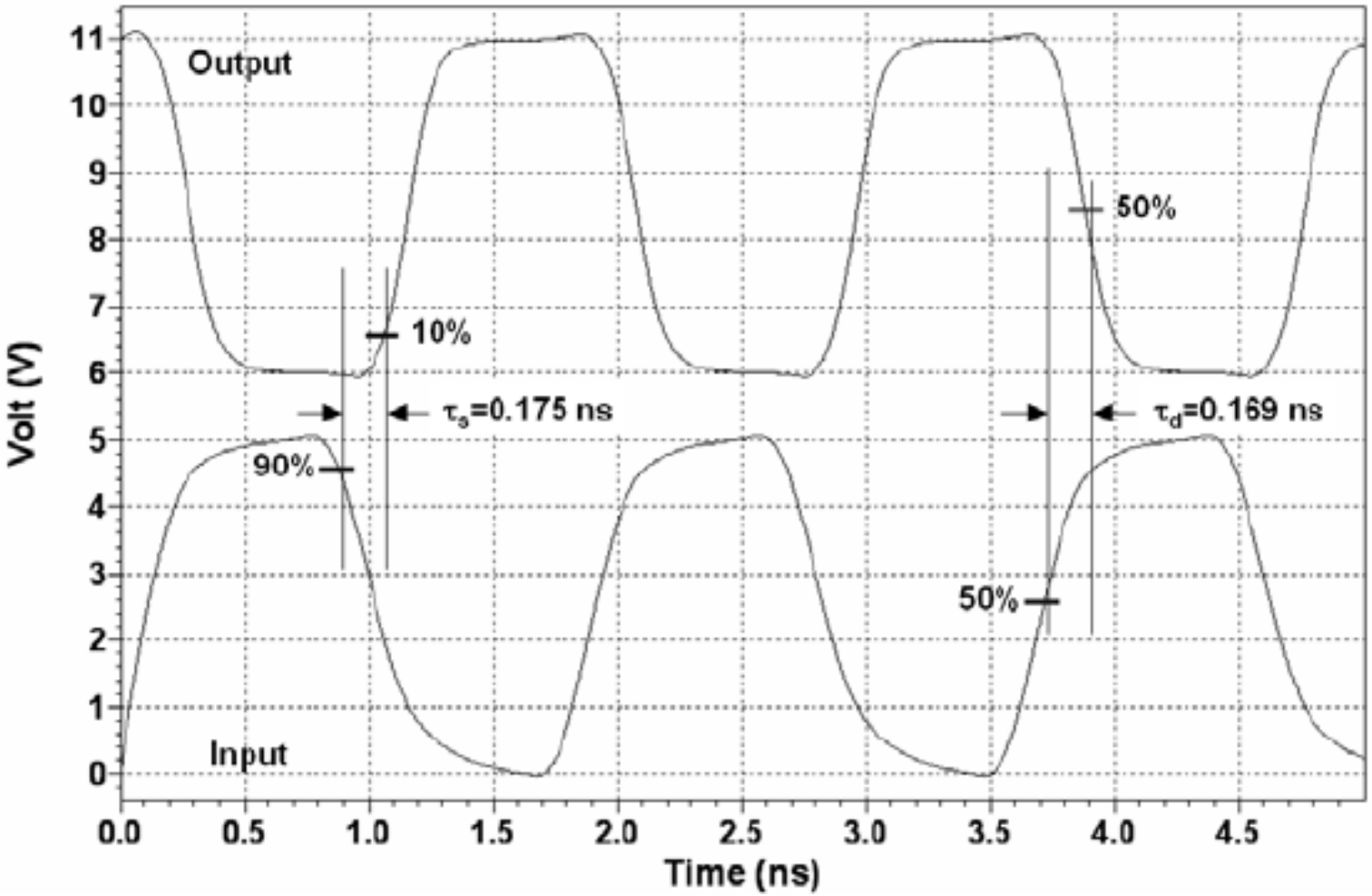
- Combinational Circuit Delay
  - In reality, outputs are delayed from inputs
    - Transistors take a finite amount of time to switch





COMBINATIONAL CIRCUIT TIMING

- Real Inverter Delay Example



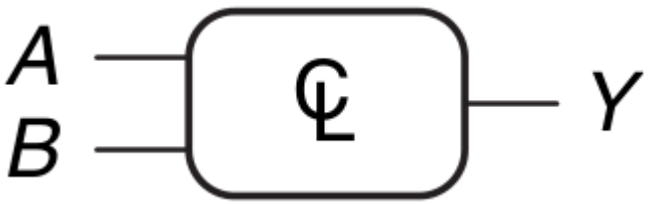
COMBINATIONAL CIRCUIT TIMING

- Circuit Delay and Its Variation
  - Delay is fundamentally caused by
    - Capacitance and resistance in a circuit
    - Finite speed of light (not so fast on a nanosecond scale!)
  - Anything affecting these quantities can change delay:
    - Rising (i.e., 0 -> 1) vs. falling (i.e., 1 -> 0) inputs
    - Different inputs have different delays
    - Changes in environment (e.g., temperature)
    - Aging of the circuit
  - We have a range of possible delays from input to output

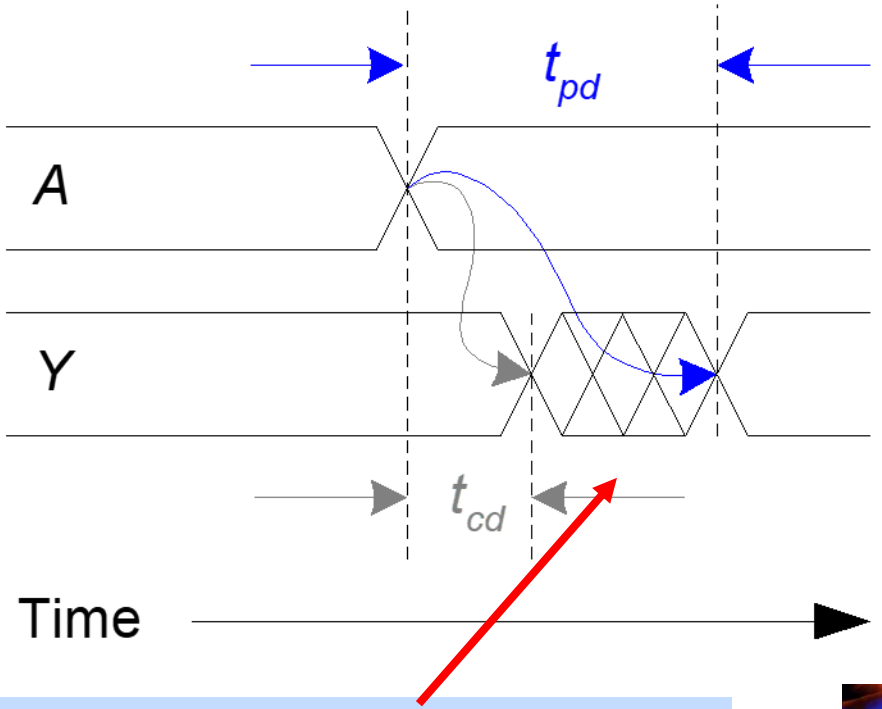
COMBINATIONAL CIRCUIT TIMING

- Delays from Input to Output Y
  - Contamination delay (t_{cd}): delay until Y starts changing
  - Propagation delay (t_{pd}): delay until Y finishes changing

Example Circuit



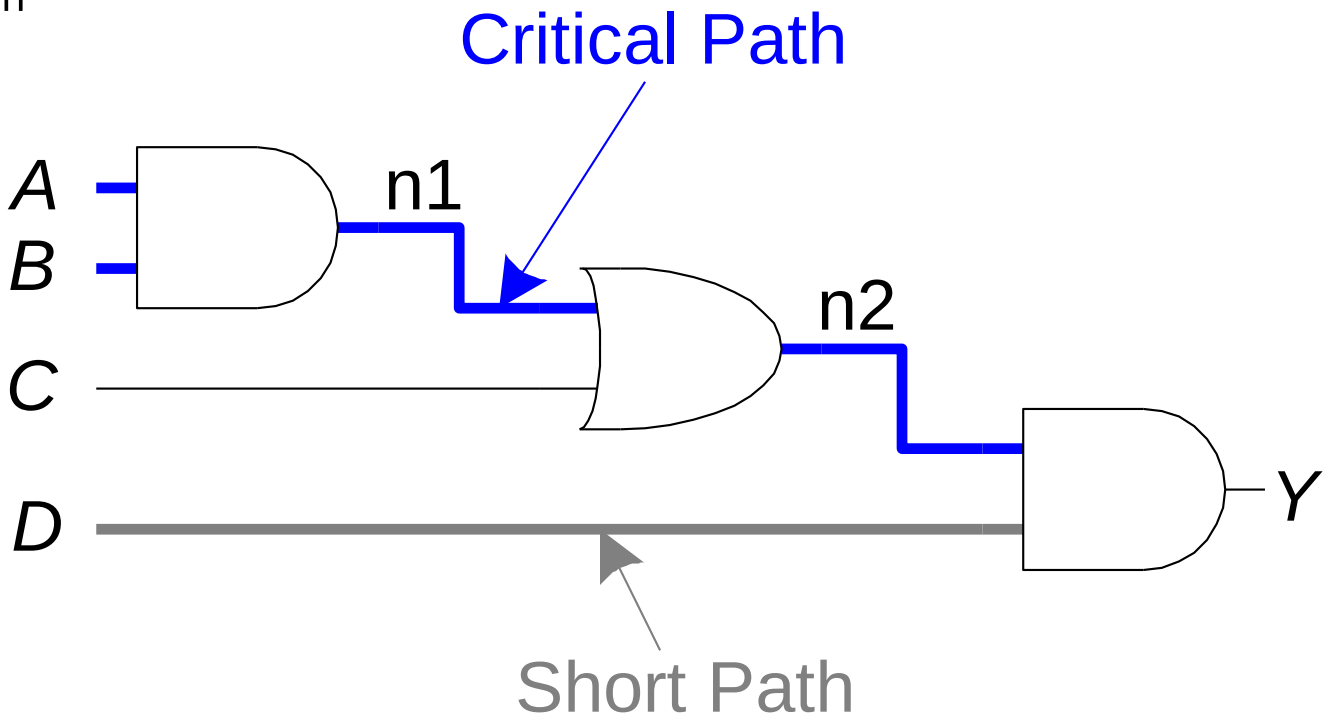
Effect of Changing Input 'A'



Cross-hatching means value is changing

COMBINATIONAL CIRCUIT TIMING

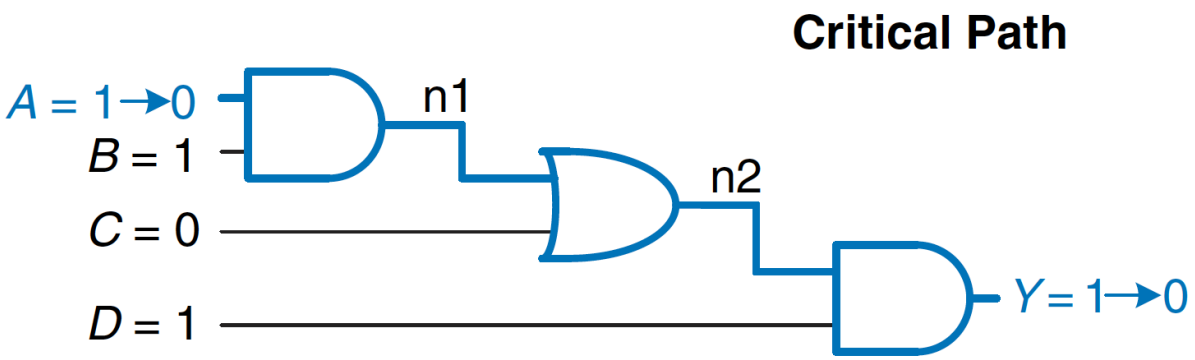
- Calculating Longest & Shortest Delay Paths
  - We care about both the longest and shortest delay paths in a circuit

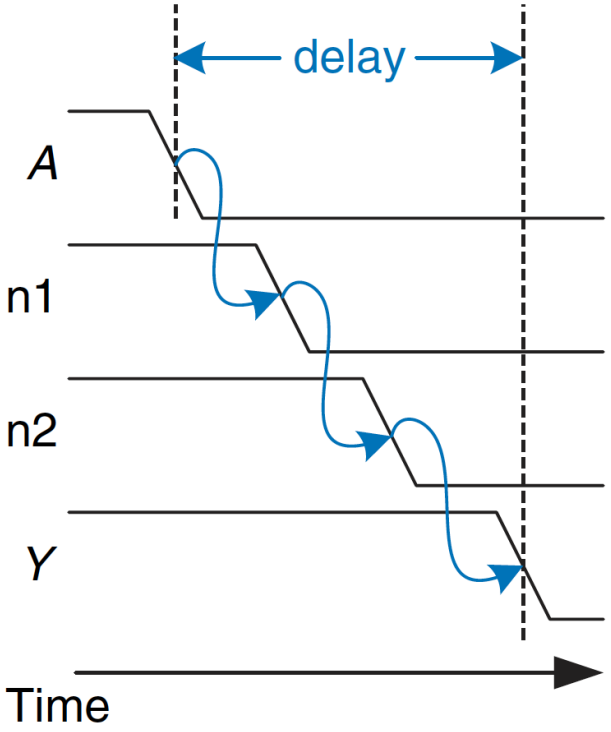


  - Critical (Longest) Path: $t_{pd} = 2\,t_{pd\_AND} + t_{pd\_OR}$
  - Shortest Path: $t_{cd} = t_{cd\_AND}$

COMBINATIONAL CIRCUIT TIMING

- Calculating Longest & Shortest Delay Paths
  - We care about both the longest and shortest delay paths in a circuit

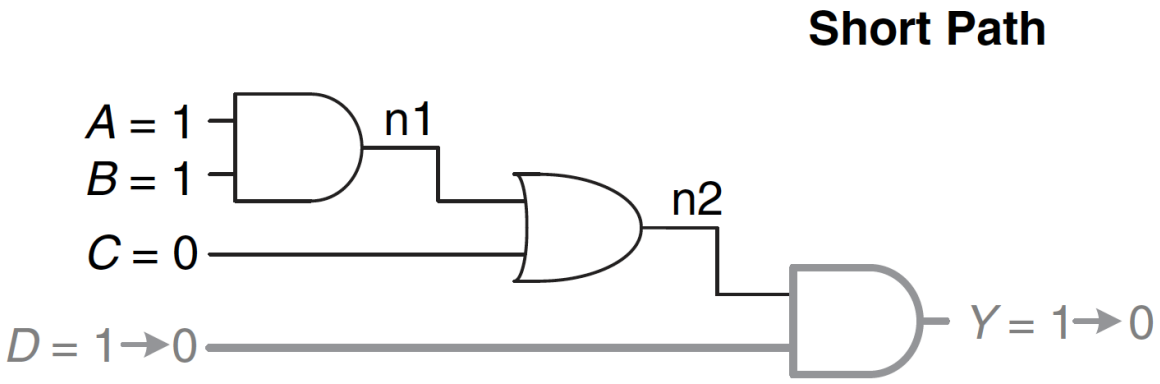


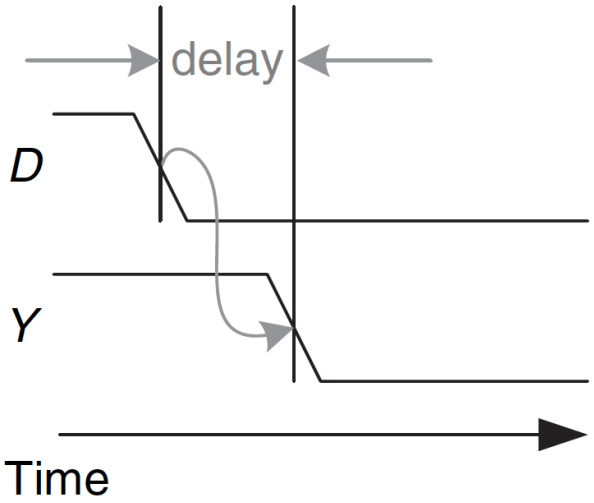


- Critical (Longest) Path: $t_{pd} = 2\,t_{pd\_AND} + t_{pd\_OR}$
- Shortest Path: $t_{cd} = t_{cd\_AND}$

COMBINATIONAL CIRCUIT TIMING

- Calculating Longest & Shortest Delay Paths
  - We care about both the longest and shortest delay paths in a circuit

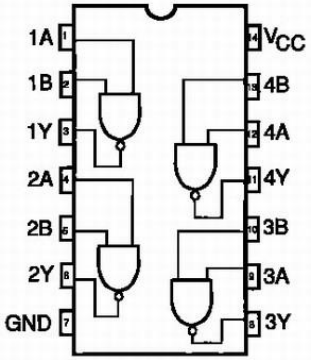




- Critical (Longest) Path: $t_{pd} = 2\,t_{pd\_AND} + t_{pd\_OR}$
- Shortest Path: $t_{cd} = t_{cd\_AND}$

COMBINATIONAL CIRCUIT TIMING

- Example t_{pd} for a Real NAND-2 Gate



Symbol	Parameter	Conditions	25 °C			–40 °C to +125 °C		Unit
			Min	Typ	Max	Max (85 °C)	Max (125 °C)	
74HC00								
t_{pd}	propagation delay	nA, nB to nY; see Figure 6 [1]						
		V_{CC} = 2.0 V	-	25	-	115	135	ns
		V_{CC} = 4.5 V	-	9	-	23	27	ns
		V_{CC} = 5.0 V; C_L = 15 pF	-	7		-	-	ns
		V_{CC} = 6.0 V	-	7	-	20	23	ns

- Heavy dependence on voltage and temperature

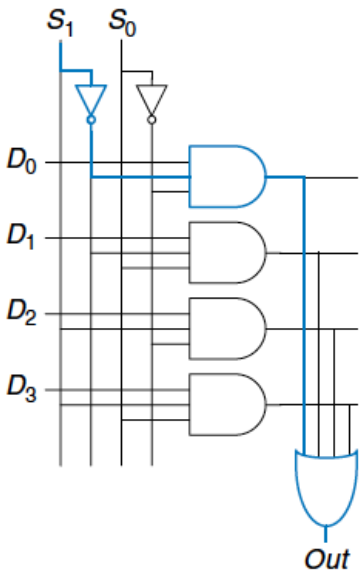
COMBINATIONAL CIRCUIT TIMING

- Example Worst-Case t_{pd}
  - Two different implementations of a 4:1 multiplexer

Gate Delays

Gate	t_{pd} (ps)
NOT	30
2-input AND	60
3-input AND	80
4-input OR	90
tristate (A to Y)	50
tristate (enable to Y)	35

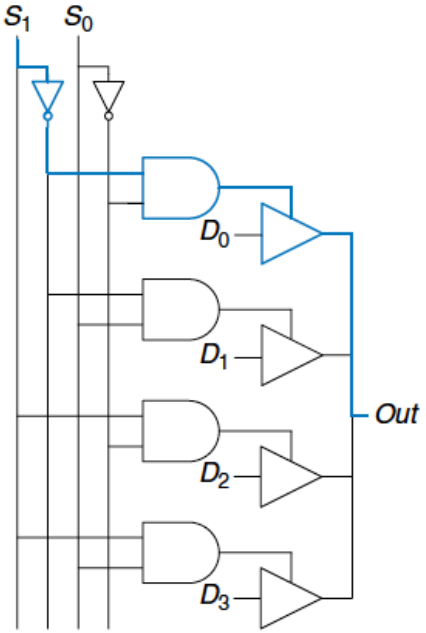
Implementation 1



$t_{pd\_sy} = t_{pd\_INV} + t_{pd\_AND3} + t_{pd\_OR4}$
$= 30\text{ ps} + 80\text{ ps} + 90\text{ ps}$
$= 200\text{ ps}$

(a) $t_{pd\_dy} = t_{pd\_AND3} + t_{pd\_OR4}$
$= 170\text{ ps}$

Implementation 2



$t_{pd\_sy} = t_{pd\_INV} + t_{pd\_AND2} + t_{pd\_TRI\_SY}$
$= 30\text{ ps} + 60\text{ ps} + 35\text{ ps}$
$= 125\text{ ps}$

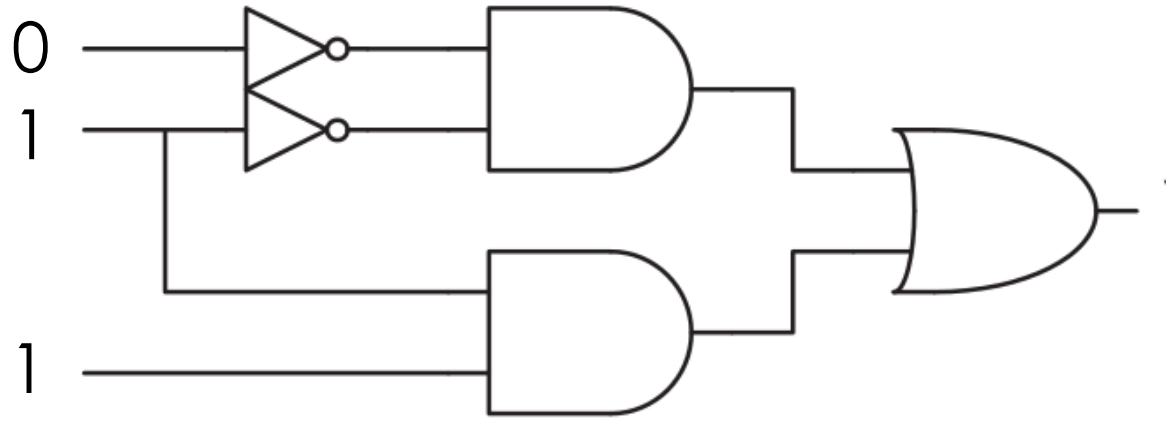
(b) $t_{pd\_dy} = t_{pd\_TRI\_AY}$
$= 50\text{ ps}$

- Different designs lead to very different delays

COMBINATIONAL CIRCUIT TIMING

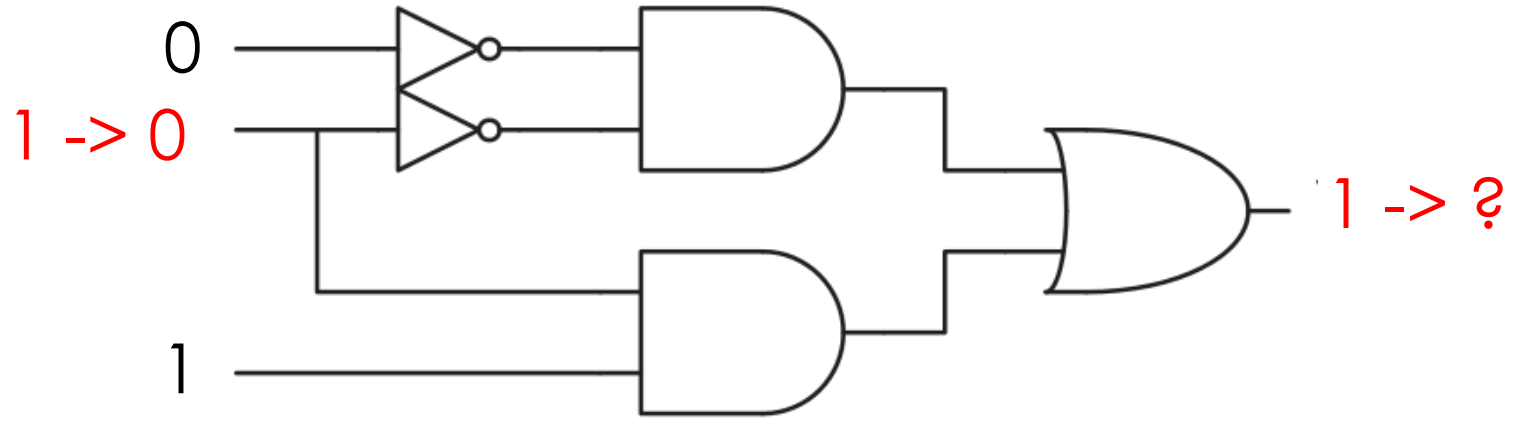
- Output Glitches
  - Glitch: one input transition causes multiple output transitions

Circuit initial state



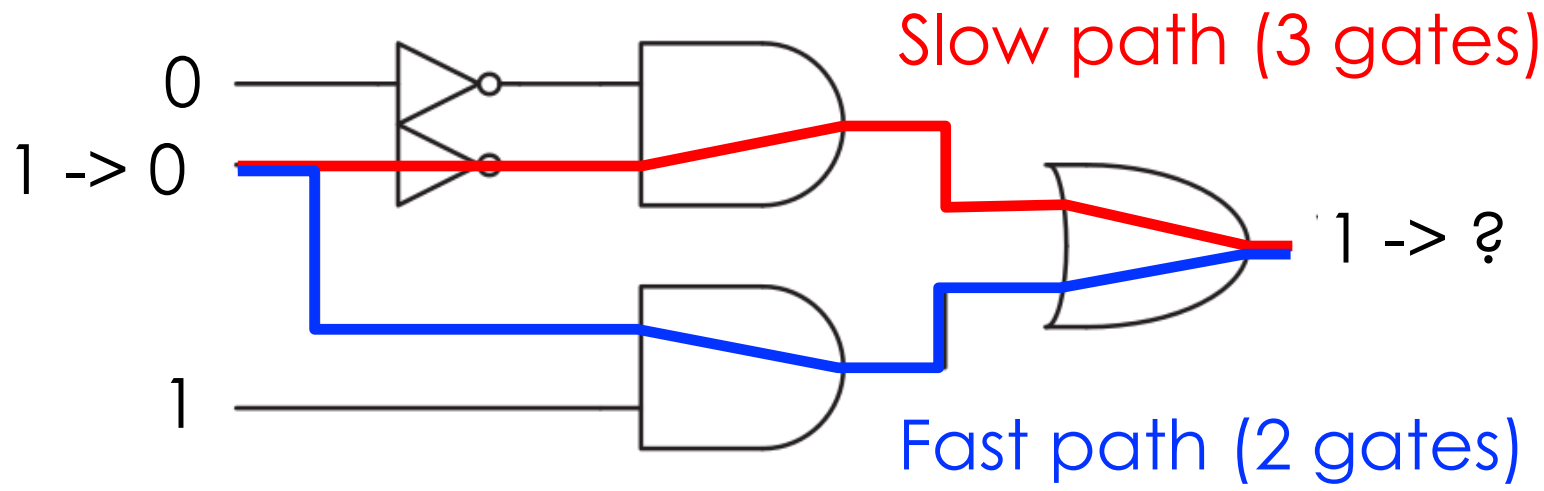
COMBINATIONAL CIRCUIT TIMING

- Output Glitches
  - Glitch: one input transition causes multiple output transitions



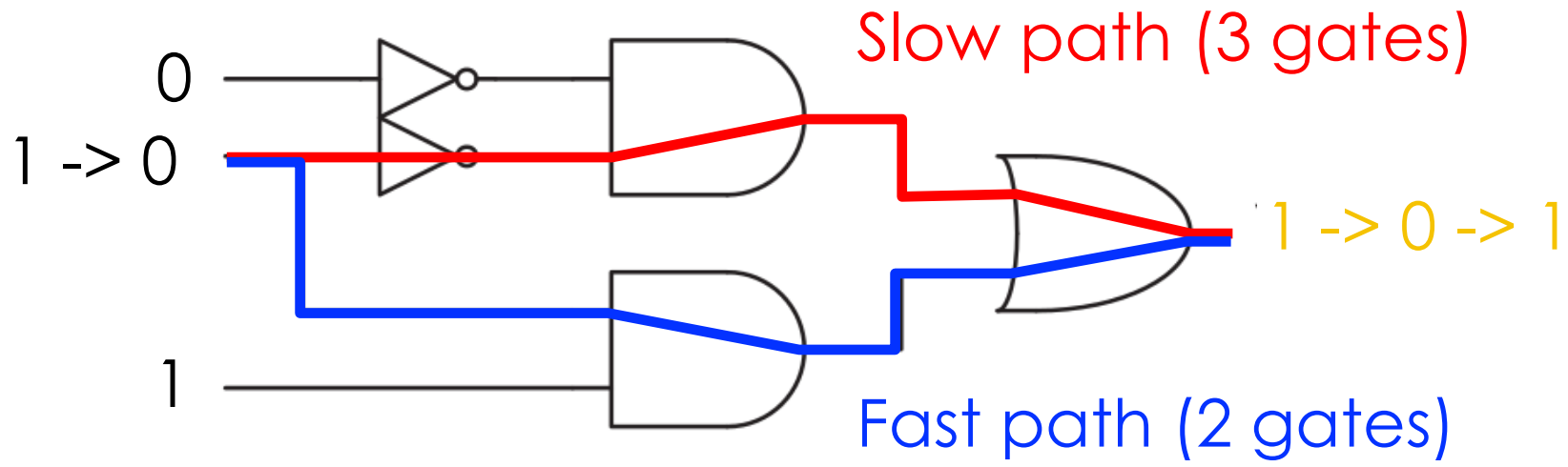
COMBINATIONAL CIRCUIT TIMING

- Output Glitches
  - Glitch: one input transition causes multiple output transitions



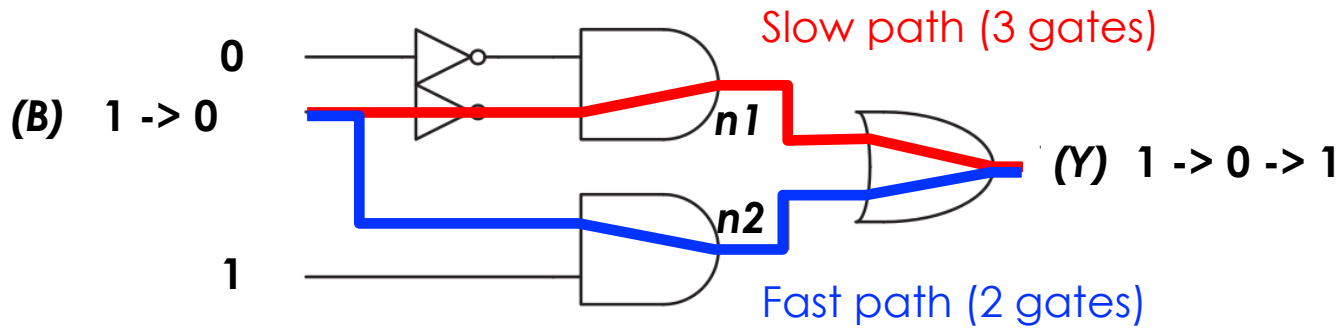
COMBINATIONAL CIRCUIT TIMING

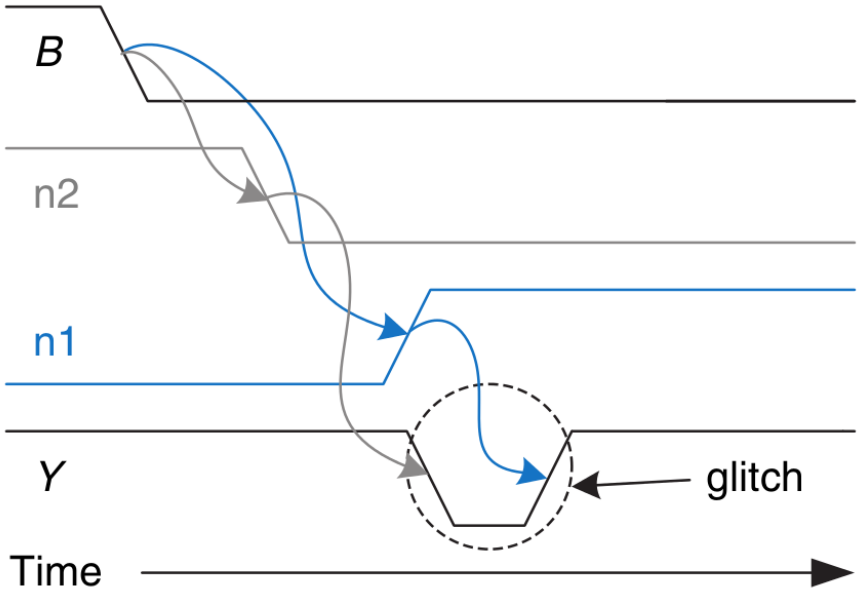
- Output Glitches
  - Glitch: one input transition causes multiple output transitions



COMBINATIONAL CIRCUIT TIMING

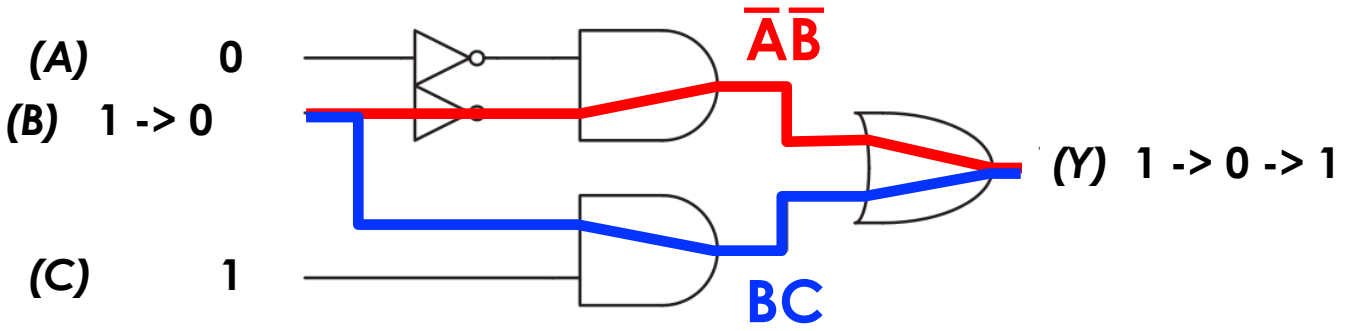
- Output Glitches
  - Glitch: one input transition causes multiple output transitions

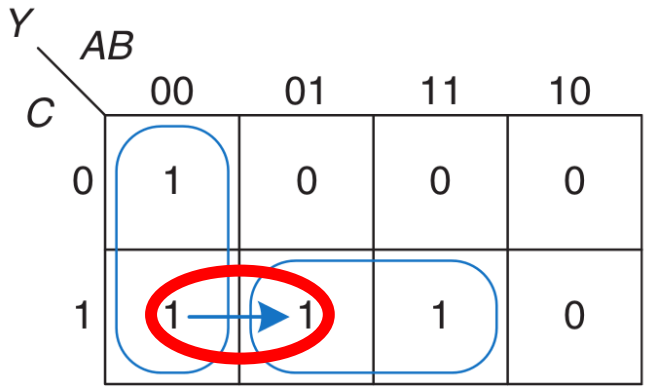




COMBINATIONAL CIRCUIT TIMING

- Avoiding Glitches Using K-Maps
  - Glitches are visible in K-maps
    - K-maps show the results of a change in a single input
    - A glitch occurs when moving between prime implicants

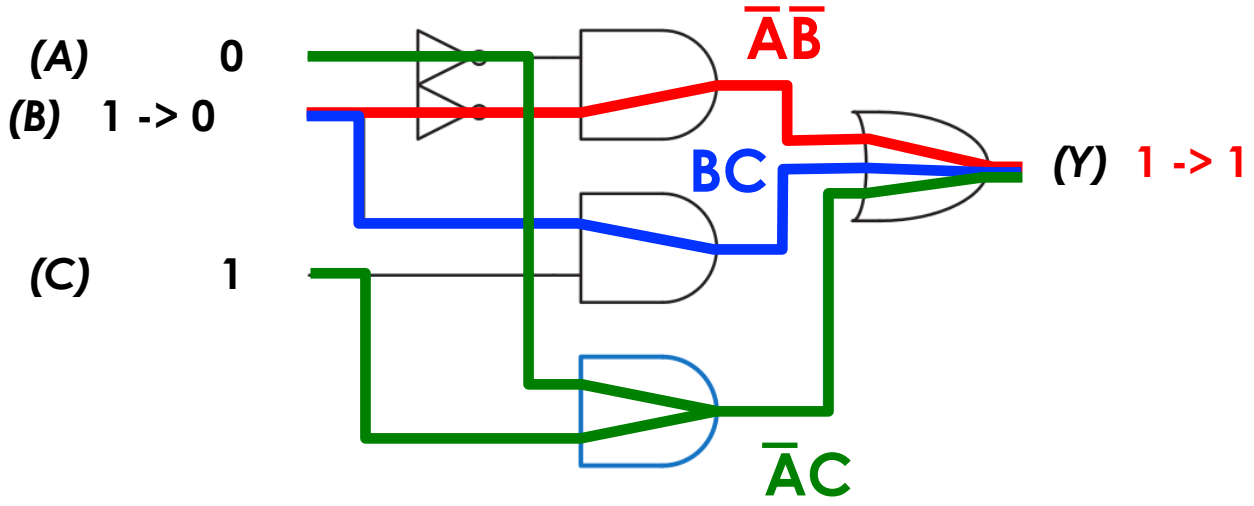


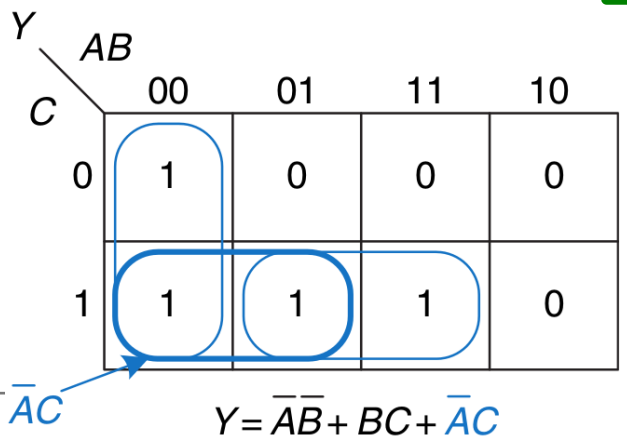


$Y = \overline{A}\overline{B} + BC$

COMBINATIONAL CIRCUIT TIMING

- Avoiding Glitches Using K-Maps
  - We can fix the issue by adding in the consensus term
    - Ensures no transition between different prime implicants



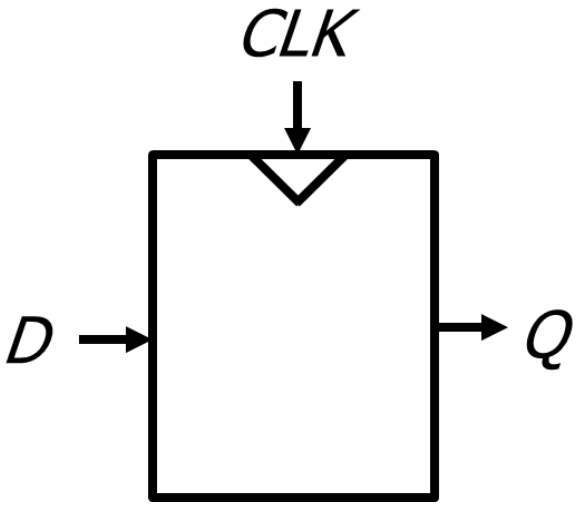


COMBINATIONAL CIRCUIT TIMING

- Avoiding Glitches
  - Question: Do we always care about glitches?
    - Fixing glitches is undesirable
    - More chip area
    - More power consumption
    - More design effort
  - The circuit is eventually guaranteed to converge to the right value regardless of glitchiness
  - Answer: No, not always
    - If we only care about the long-term steady state output, we can safely ignore glitches
    - Up to the designer to decide if glitches matter in their application
      - When examining simulation output, important to recognize glitches

SEQUENTIAL CIRCUIT TIMING

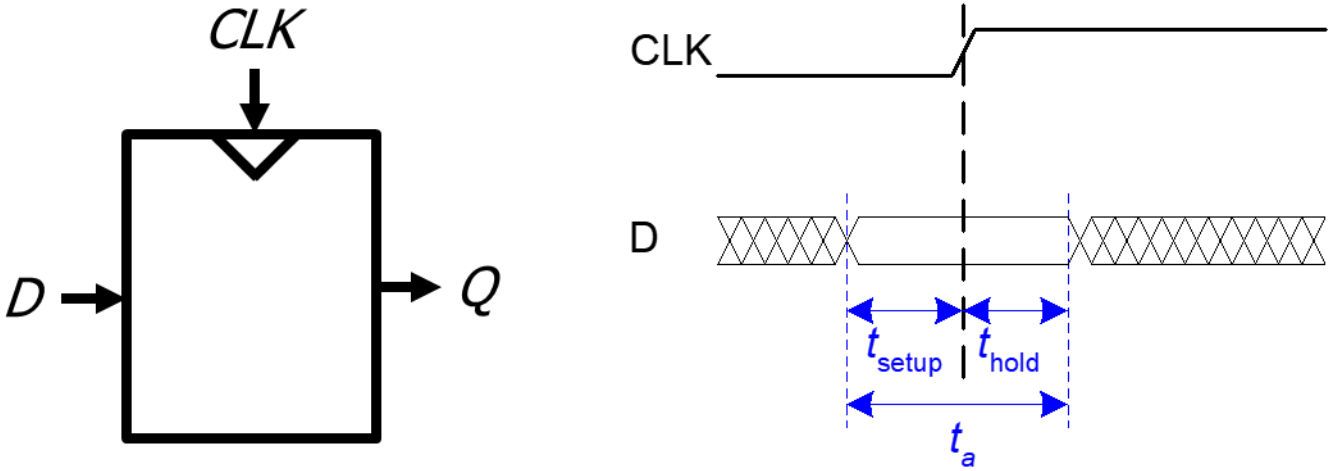
- D flip-flops
  - Flip-flop samples D at the active clock edge
  - It outputs the sampled value to Q
  - It "stores" the sampled value until the next active clock edge



  - The D flip-flop is made from combinational elements
  - D, Q, CLK all have timing requirements!

SEQUENTIAL CIRCUIT TIMING

- D Flip-Flop Input Timing Constraints
  - D must be stable when sampled (i.e., at active clock edge)

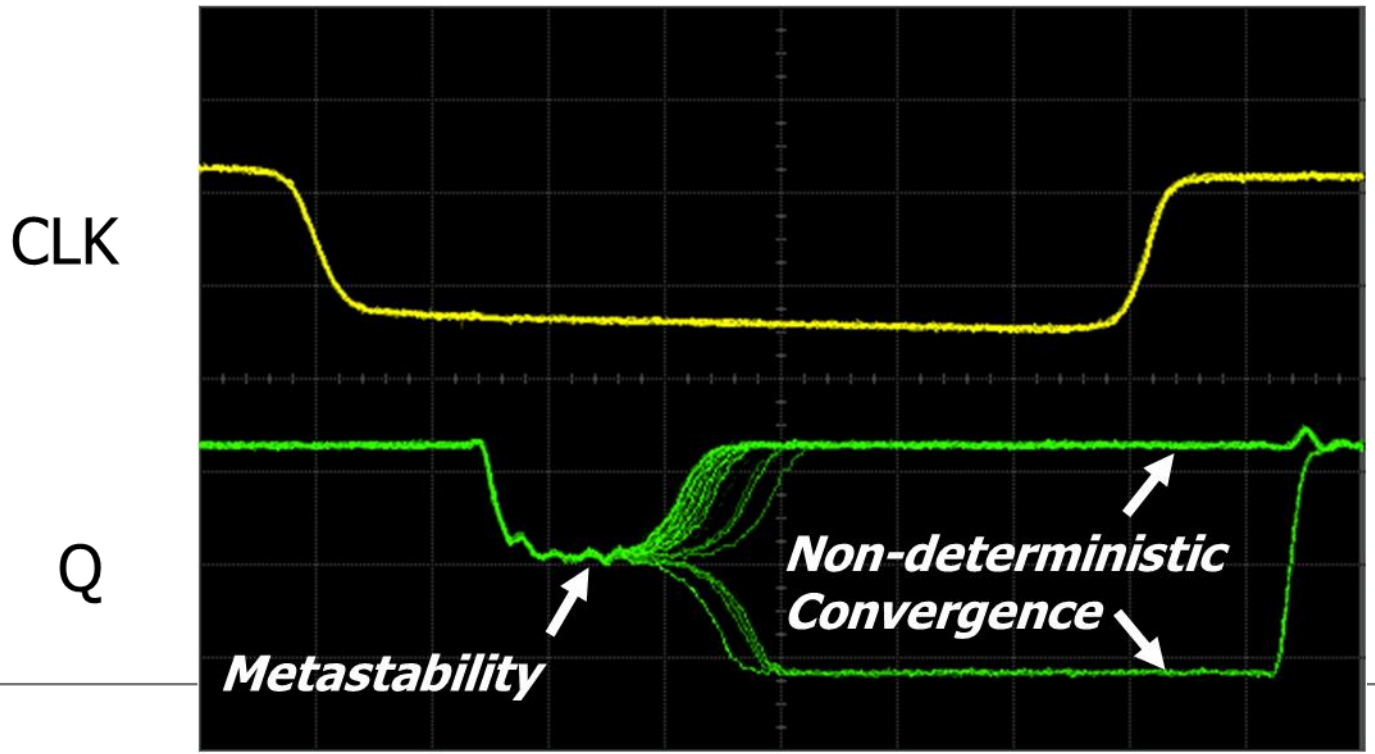


  - Setup time (t_{setup}): time before the clock edge that data must be stable (i.e. not changing)
  - Hold time (t_{hold}): time after the clock edge that data must be stable
  - Aperture time (t_a): time around clock edge that data must be stable ($t_a = t_{setup} + t_{hold}$)

SEQUENTIAL CIRCUIT TIMING

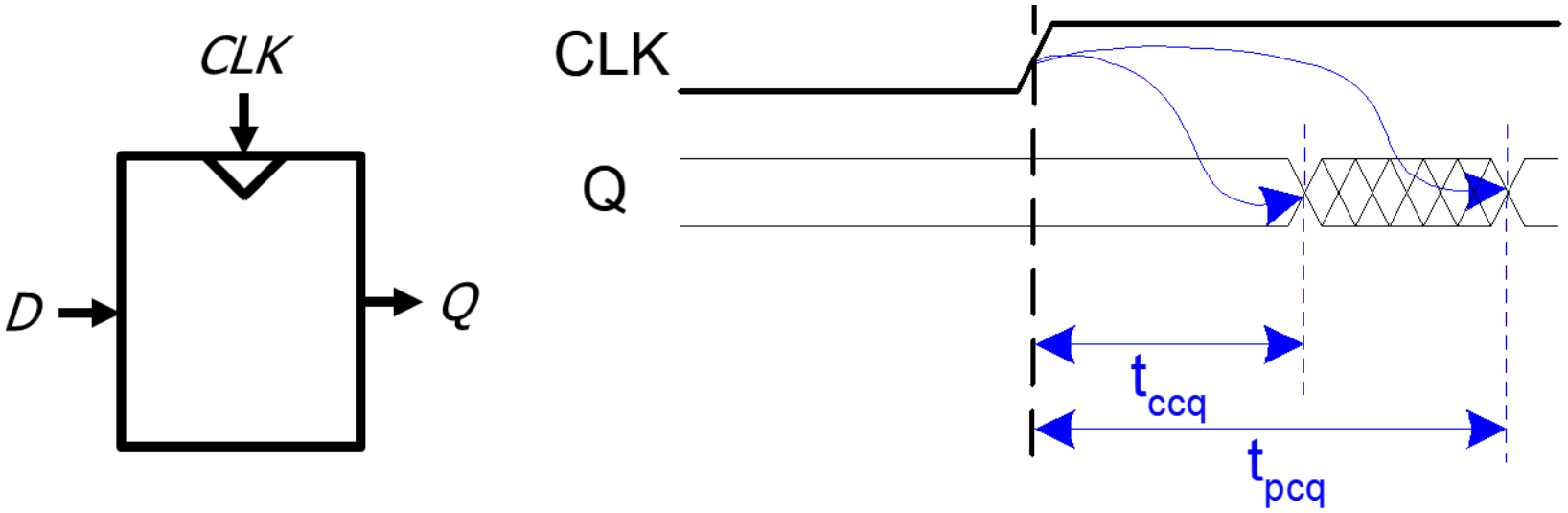
- Violating Input Timing: Metastability
  - If D is changing when sampled, metastability can occur
  - Flip-flop output is stuck somewhere between '1' and '0'
  - Output eventually settles non-deterministically

Example Timing Violations (NAND RS Latch)



SEQUENTIAL CIRCUIT TIMING

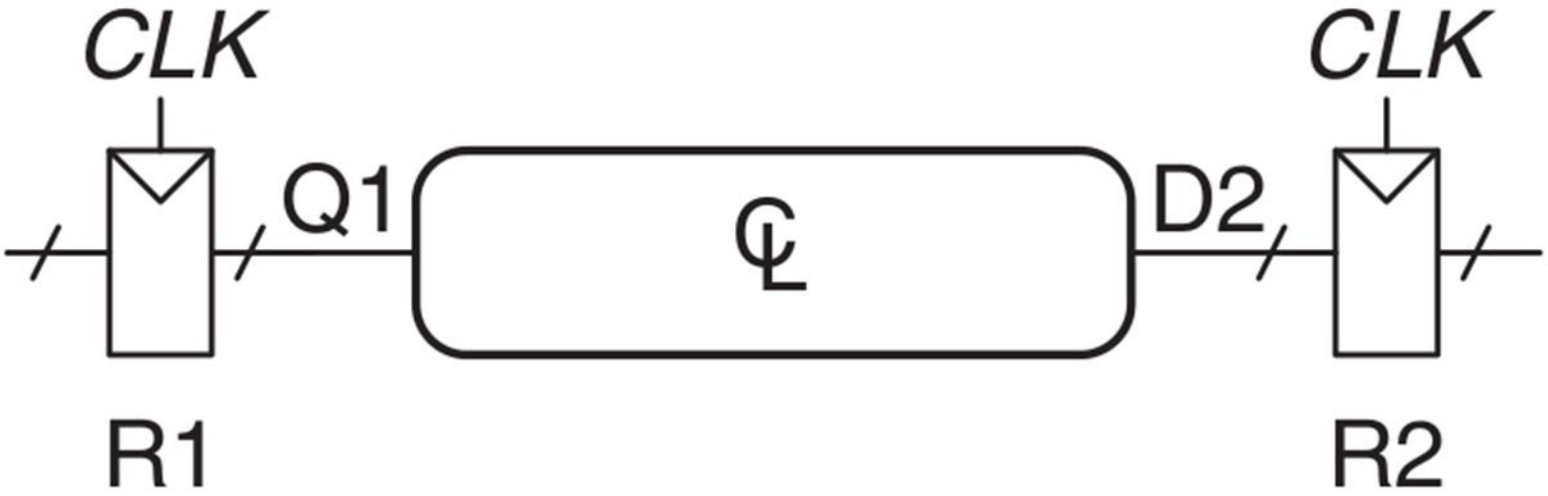
- Flip-Flop Output Timing



- Contamination delay clock-to-q (t_{ccq}): earliest time after the clock edge that Q starts to change (i.e., is unstable)
- Propagation delay clock-to-q (t_{pcq}): latest time after the clock edge that Q stops changing (i.e., is stable)

SEQUENTIAL CIRCUIT TIMING

- Sequential System Design

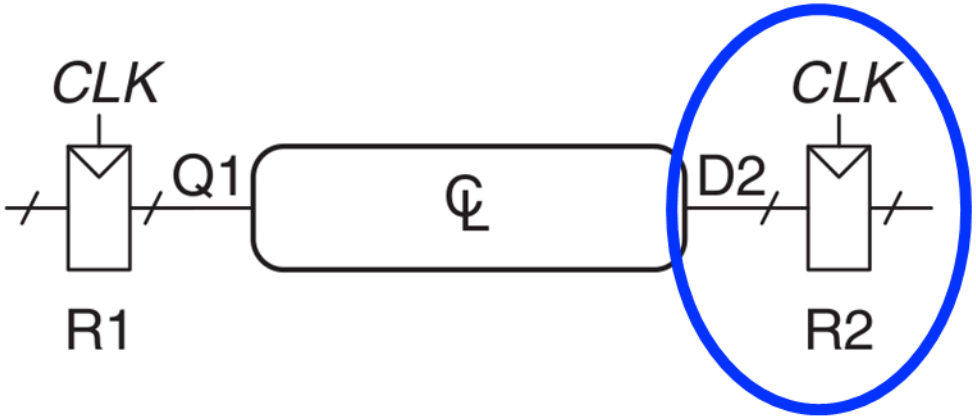


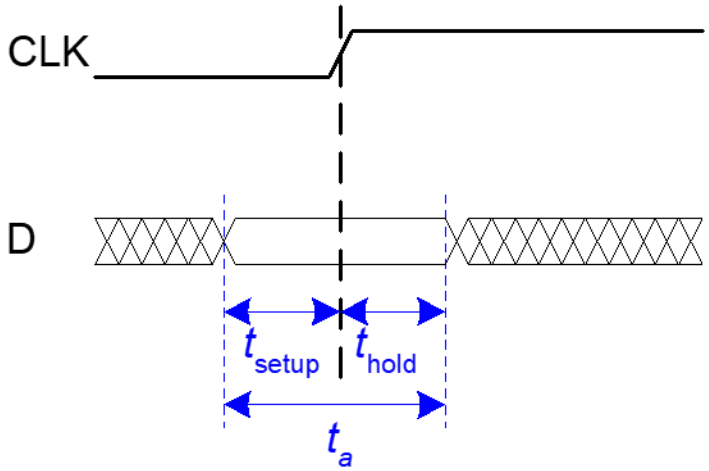
- Multiple flip-flops are connected with combinational logic
- Clock runs with period Tc (cycle time)

- Must meet timing requirements for both R1 and R2!

SEQUENTIAL CIRCUIT TIMING

- Ensuring Correct Sequential Operation
  - Need to ensure correct input timing on R2
  - Specifically, D2 must be stable:
    - at least t_{setup} before the clock edge
    - at least until t_{hold} after the clock edge



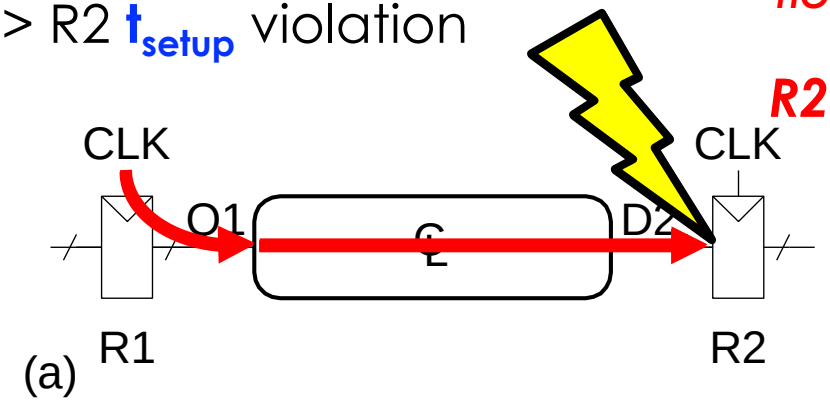


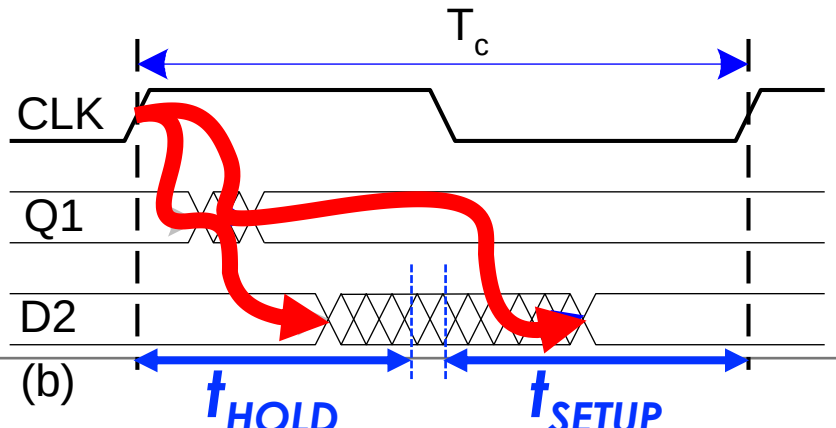
SEQUENTIAL CIRCUIT TIMING

- Ensuring Correct Sequential Operation
  - This means there is both a minimum and maximum delay between two flip-flops
  - CL **too fast** -> R2 t_{hold} violation
  - CL **too slow** -> R2 t_{setup} violation

Potential R2 t_{HOLD} VIOLATION!

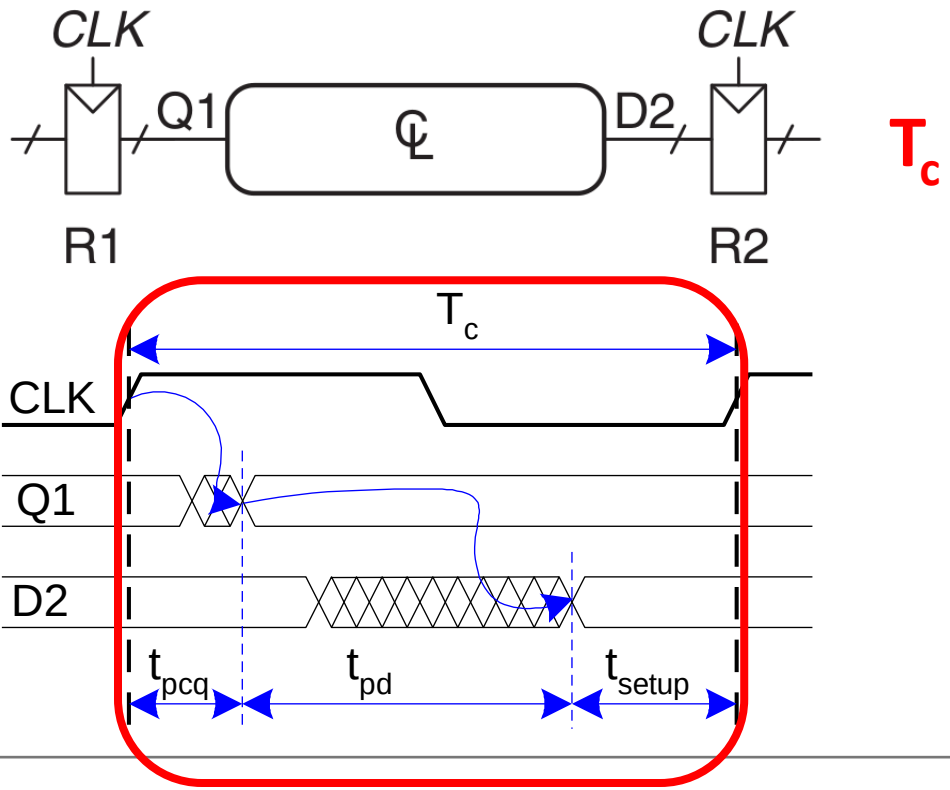
Potential R2 t_{SETUP} VIOLATION!





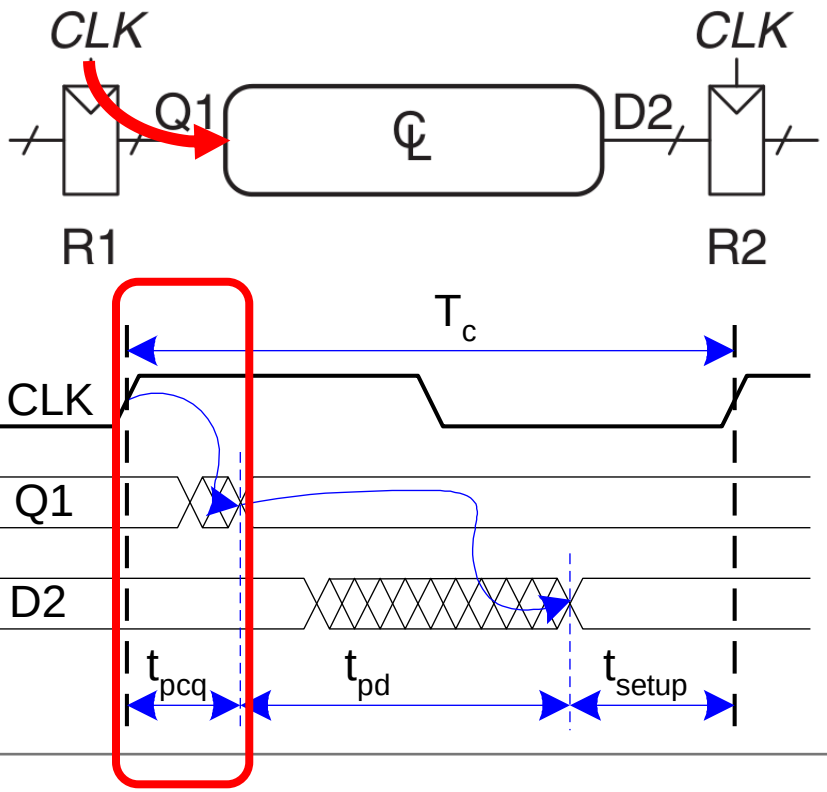
SEQUENTIAL CIRCUIT TIMING

- Setup Time Constraint
  - Safe timing depends on the maximum delay from R1 to R2
  - The input to R2 must be stable at least t_{setup} before the clock edge



SEQUENTIAL CIRCUIT TIMING

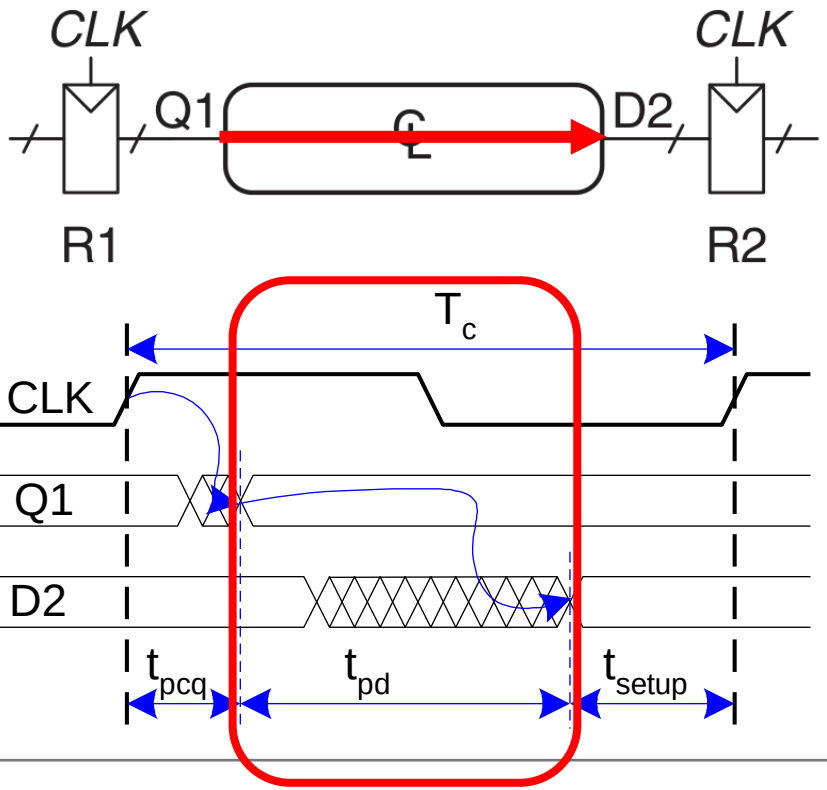
- Setup Time Constraint
  - Safe timing depends on the maximum delay from R1 to R2
  - The input to R2 must be stable at least t_{setup} before the clock edge



$T_c > t_{pcq}$

SEQUENTIAL CIRCUIT TIMING

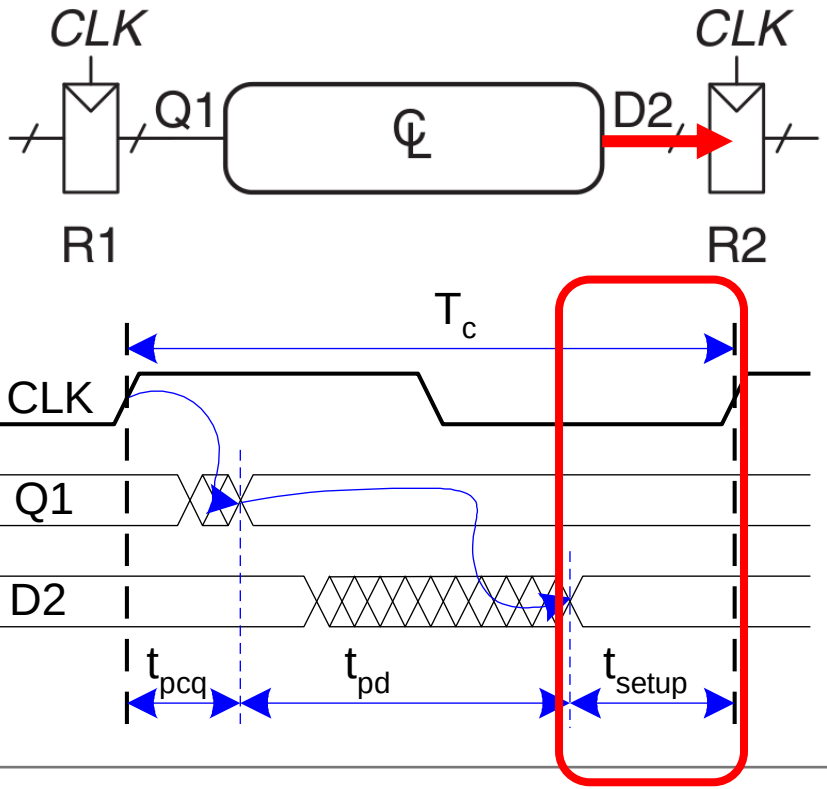
- Setup Time Constraint
  - Safe timing depends on the maximum delay from R1 to R2
  - The input to R2 must be stable at least t_{setup} before the clock edge



$$T_c > t_{pcq} + t_{pd}$$

SEQUENTIAL CIRCUIT TIMING

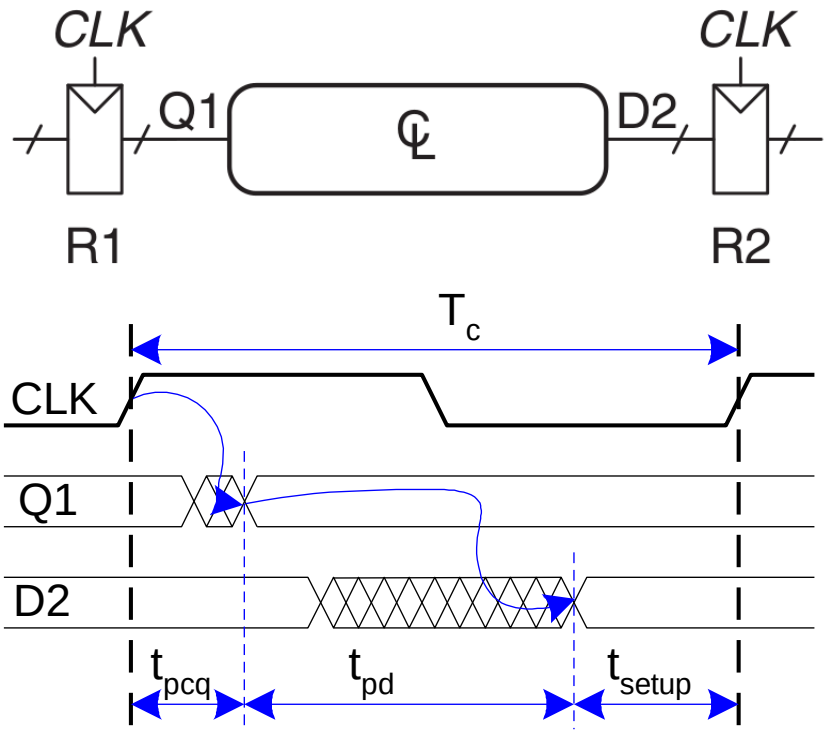
- Setup Time Constraint
  - Safe timing depends on the maximum delay from R1 to R2
  - The input to R2 must be stable at least t_{setup} before the clock edge



$$T_c > t_{pcq} + t_{pd} + t_{setup}$$

SEQUENTIAL CIRCUIT TIMING

- Setup Time Constraint
  - Safe timing depends on the maximum delay from R1 to R2
  - The input to R2 must be stable at least t_{setup} before the clock edge



Wasted work

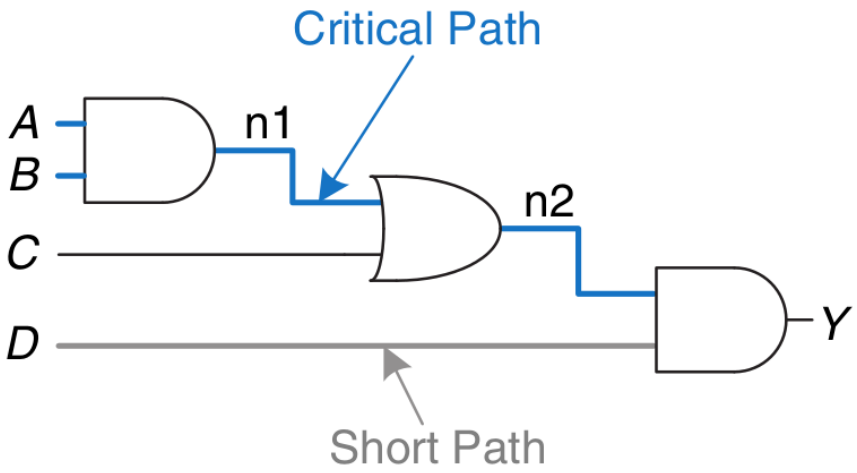
$$T_c > t_{pcq} + t_{pd} + t_{setup}$$

Useful work

Sequencing overhead: amount of time **wasted** each cycle due to sequencing element timing requirements

SEQUENTIAL CIRCUIT TIMING

- t_{setup} Constraint and Design Performance



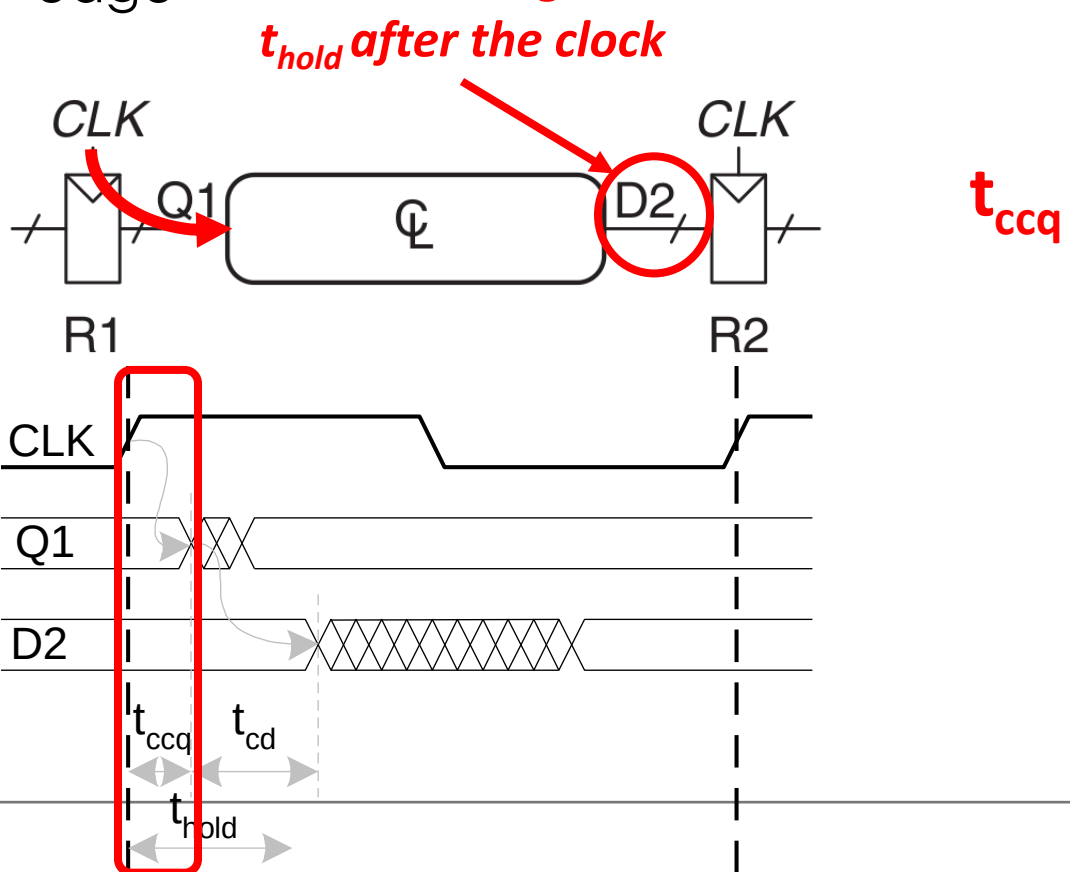
- **Critical path: path with the longest t_{pd}**

$$T_c > t_{pcq} + t_{pd} + t_{setup}$$

- **Overall design performance is determined by the critical path t_{pd}**
  - Determines the **minimum clock period** (i.e., **max operating frequency**)
  - If the critical path is too **long**, the design will run **slowly**
  - If critical path is too **short**, each cycle will do very **little useful work**
    - i.e., most of the cycle will be **wasted** in sequencing overhead

SEQUENTIAL CIRCUIT TIMING

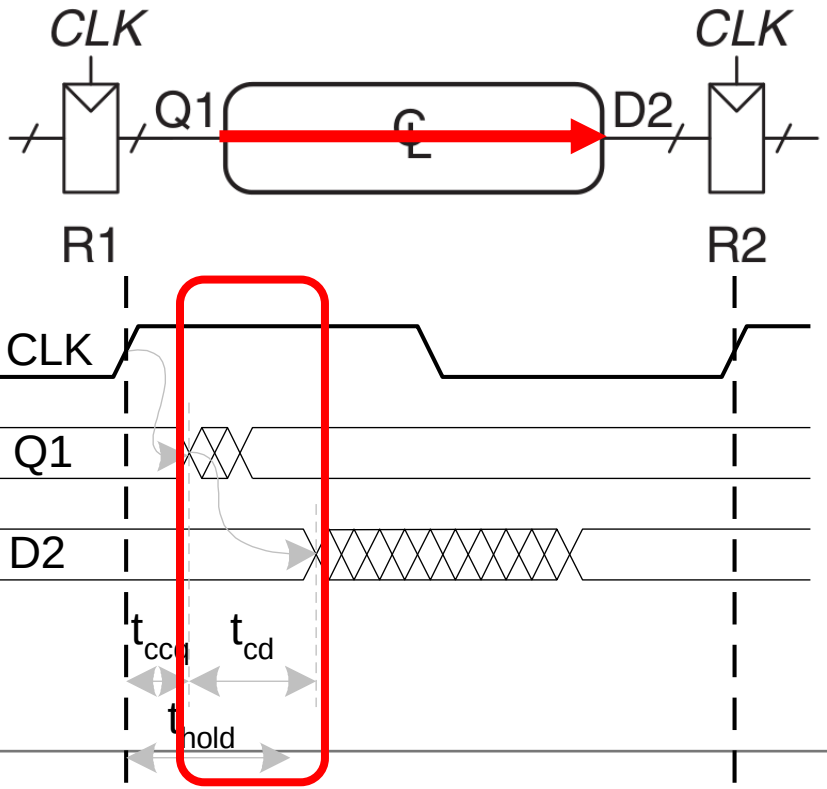
- Hold Time Constraint
  - Safe timing depends on the minimum delay from R1 to R2
  - D2 (i.e., R2 input) must be stable for at least t_{hold} after the clock edge



t_{ccq}

SEQUENTIAL CIRCUIT TIMING

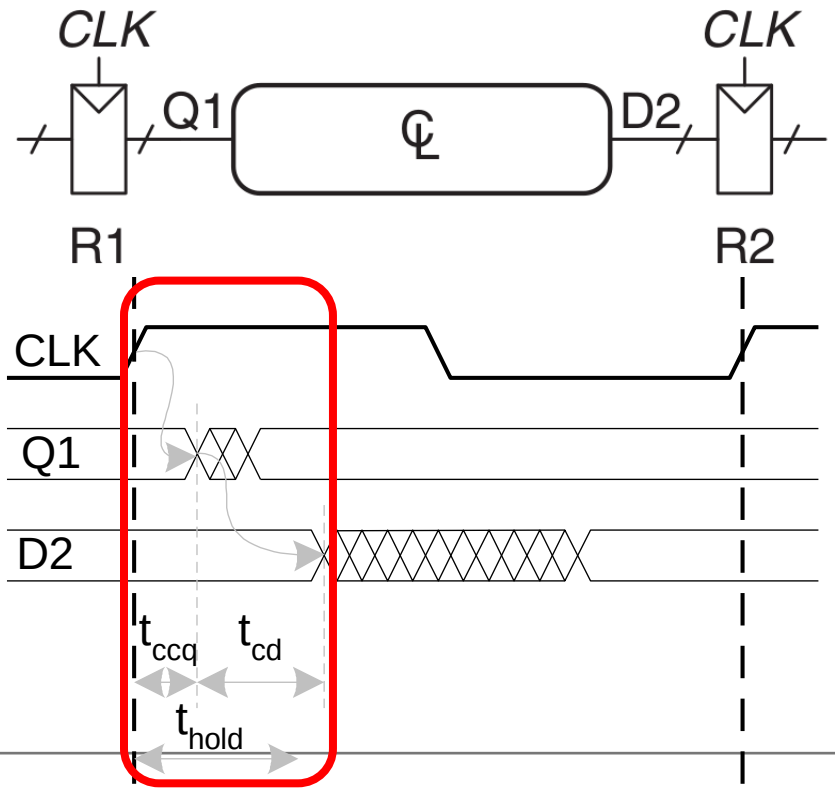
- Hold Time Constraint
  - Safe timing depends on the minimum delay from R1 to R2
  - D2 (i.e., R2 input) must be stable for at least t_{hold} after the clock edge



$t_{ccq} + t_{cd}$

SEQUENTIAL CIRCUIT TIMING

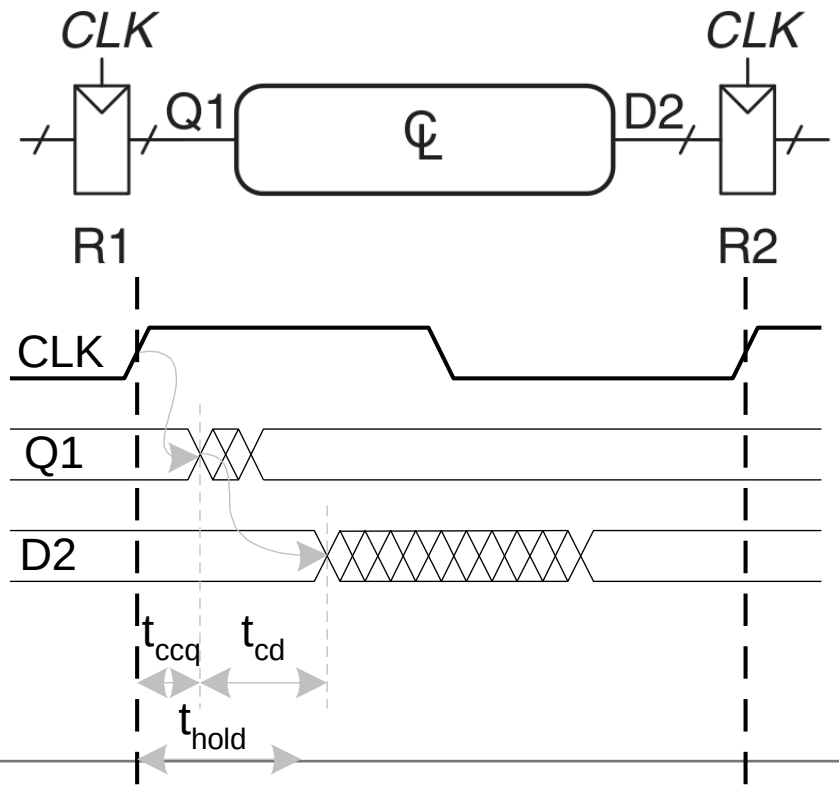
- Hold Time Constraint
  - Safe timing depends on the minimum delay from R1 to R2
  - D2 (i.e., R2 input) must be stable for at least t_{hold} after the clock edge



$$t_{ccq} + t_{cd} > t_{hold}$$

SEQUENTIAL CIRCUIT TIMING

- Hold Time Constraint
  - Safe timing depends on the minimum delay from R1 to R2
  - D2 (i.e., R2 input) must be stable for at least t_{hold} after the clock edge



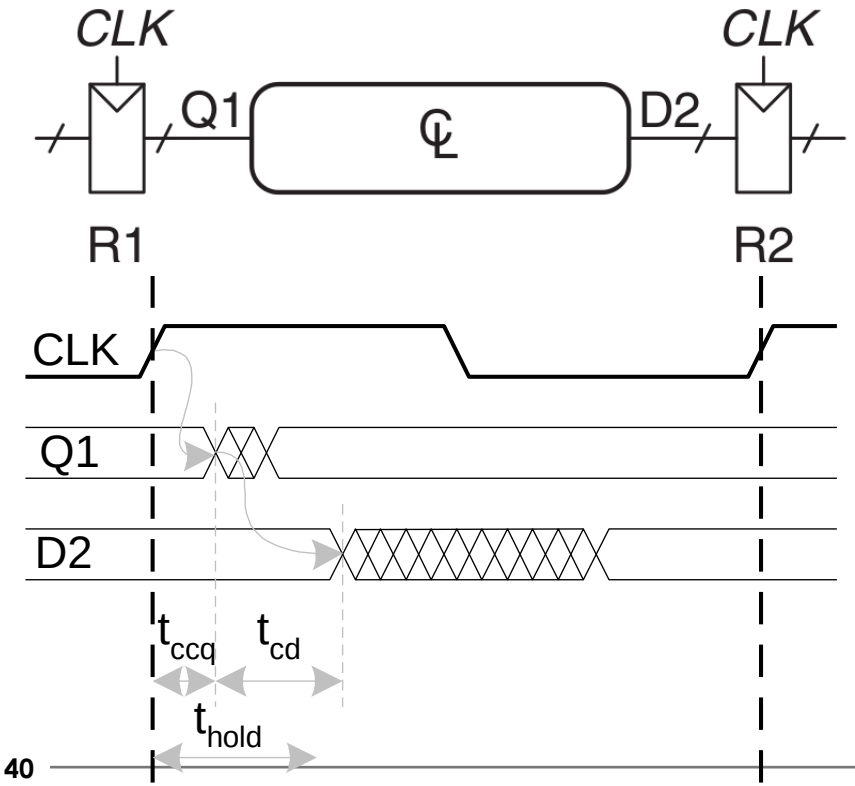
$$t_{ccq} + t_{cd} > t_{hold}$$

$$t_{cd} > t_{hold} - t_{ccq}$$

We need to have a **minimum** combinational delay!

SEQUENTIAL CIRCUIT TIMING

- Hold Time Constraint
  - Safe timing depends on the minimum delay from R1 to R2
  - D2 (i.e., R2 input) must be stable for at least t_{hold} after the clock edge



$$t_{ccq} + t_{cd} > t_{hold}$$

$$t_{cd} > t_{hold} - t_{ccq}$$

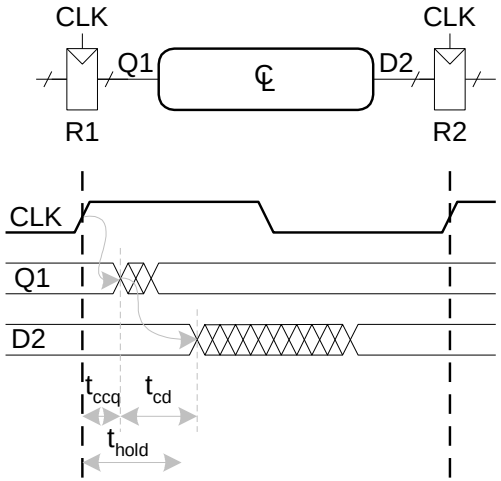
Does **NOT** depend on T_c!

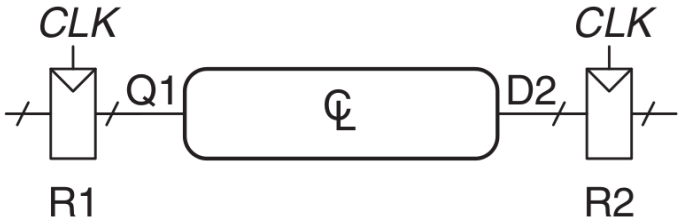
Very hard to fix t_{hold} violations after manufacturing- must modify circuits!

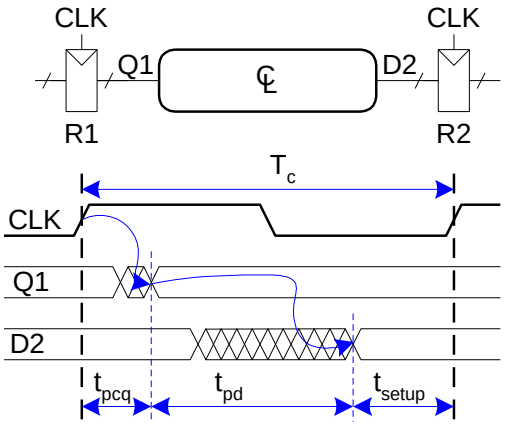
SEQUENTIAL CIRCUIT TIMING

- Sequential Timing Summary

t_{ccq} / t_{pcq}	**clock-to-q** delay (contamination/propagation)
t_{cd} / t_{pd}	**combinational logic** delay (contamination/propagation)
t_{setup}	time that **FF inputs** must be stable **before** next clock edge
t_{hold}	time that **FF inputs** must be stable **after** a clock edge
T_c	clock period

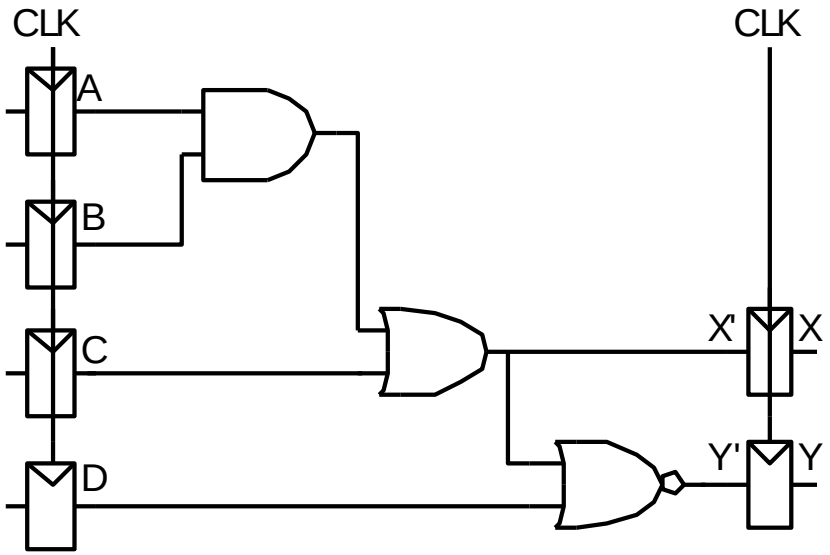






TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
t_{pd} (per gate)	= 35 ps
t_{cd} (per gate)	= 25 ps

t_{pd} =

t_{cd} =

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$

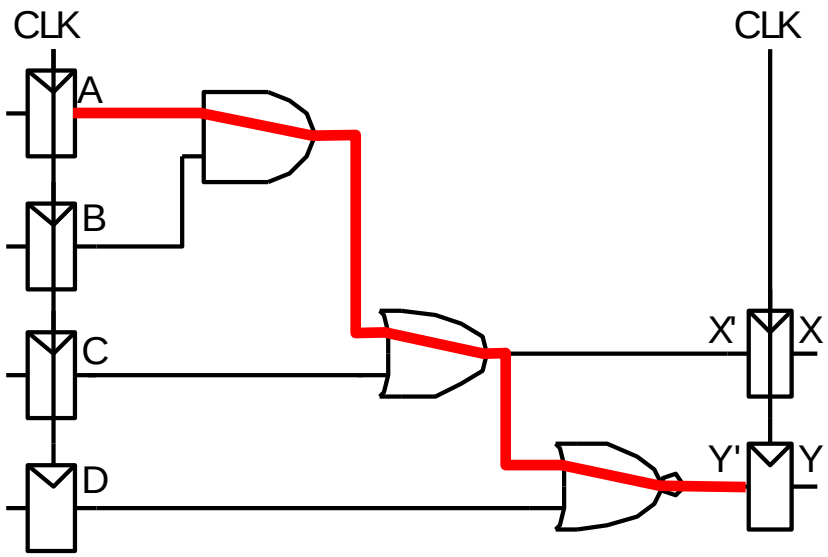
$f_{max} = 1/T_c =$

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} =

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$

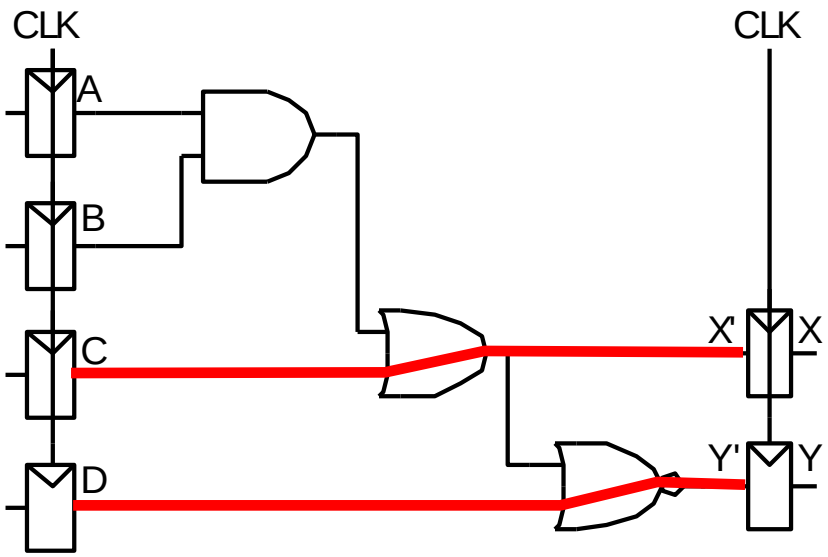
$f_{max} = 1/T_c =$

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$

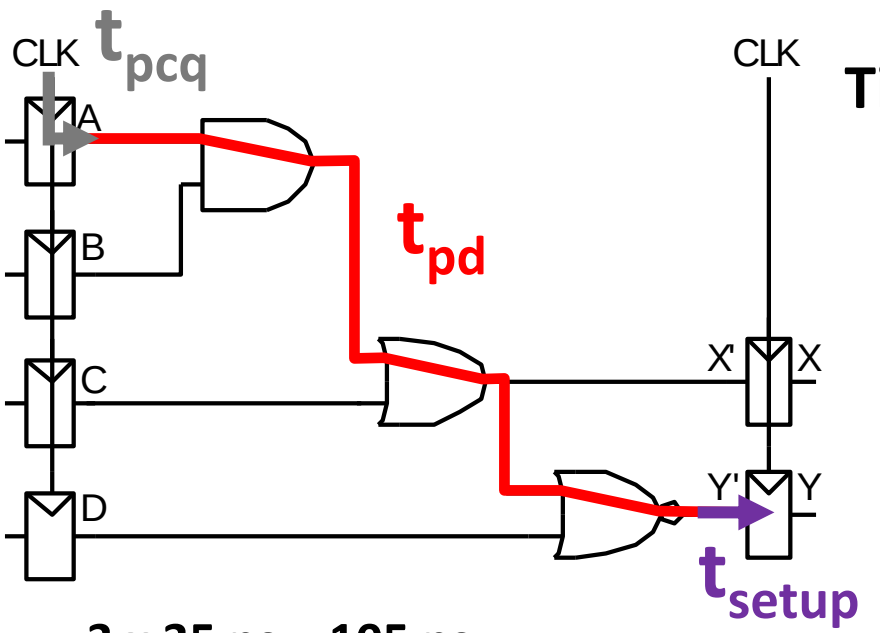
$f_{max} = 1/T_c =$

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$ (50 + 105 + 60) ps = 215 ps

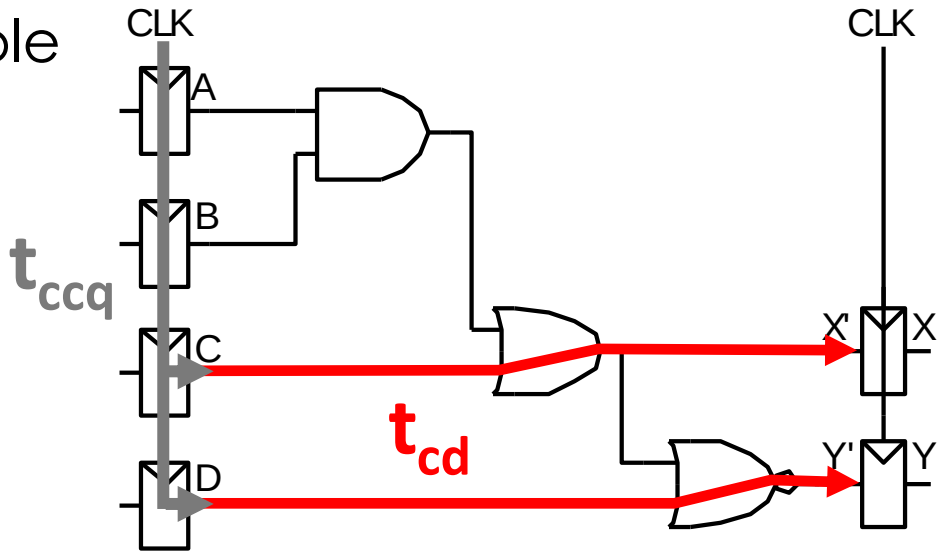
$f_{max} = 1/T_c =$ 4.65 GHz

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c > (50 + 105 + 60)$ ps = 215 ps

$f_{max} = 1/T_c = 4.65$ GHz

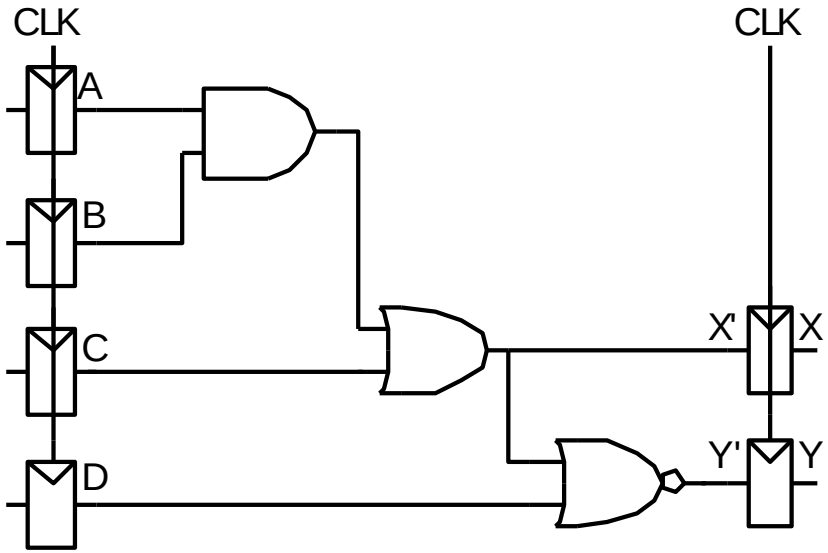
Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 25) ps > 70 ps ?

TIMING ANALYSIS

- Example



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c > (50 + 105 + 60)$ ps = 215 ps

$f_{max} = 1/T_c = 4.65$ GHz

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

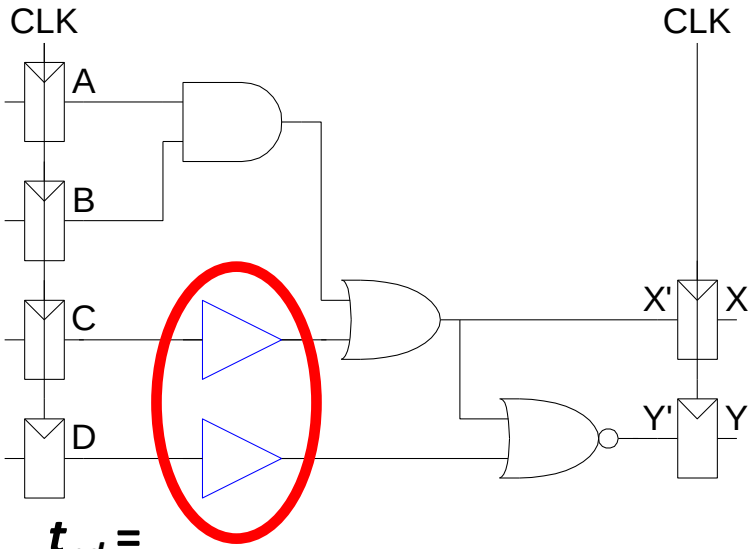
(30 + 25) ps > 70 ps ?

FAIL

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



t_{pd} =

t_{cd} =

Timing Characteristics

	t_{ccq}	= 30 ps
	t_{pcq}	= 50 ps
	t_{setup}	= 60 ps
	t_{hold}	= 70 ps
per gate	t_{pd}	= 35 ps
per gate	t_{cd}	= 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$

$f_c =$

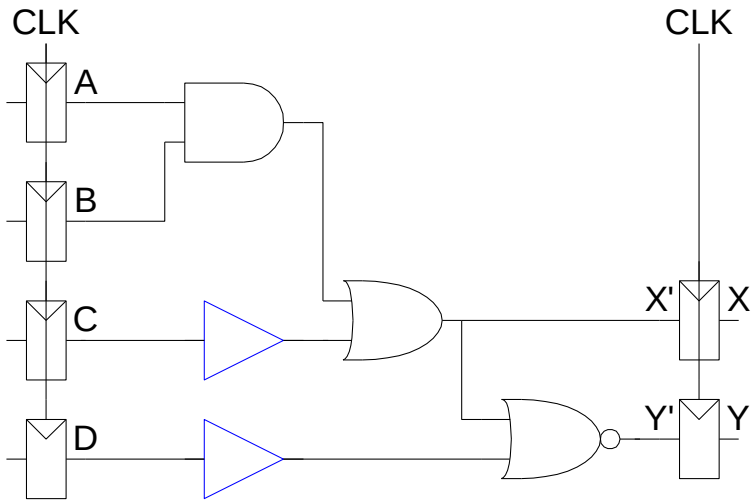
Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Timing Characteristics

	t_{ccq}	= 30 ps
	t_{pcq}	= 50 ps
	t_{setup}	= 60 ps
	t_{hold}	= 70 ps
per gate	t_{pd}	= 35 ps
per gate	t_{cd}	= 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$

$f_c =$

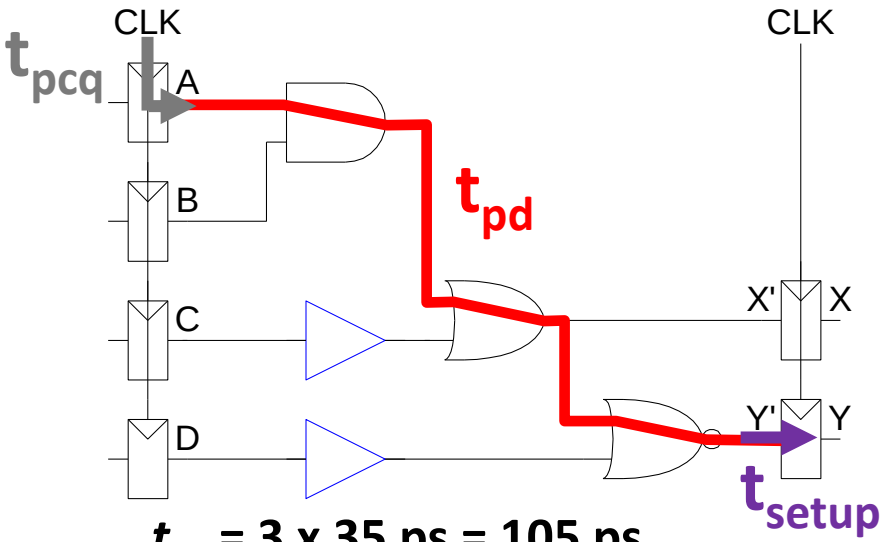
Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Timing Characteristics

	t_{ccq}	= 30 ps
	t_{pcq}	= 50 ps
	t_{setup}	= 60 ps
	t_{hold}	= 70 ps
per gate	t_{pd}	= 35 ps
per gate	t_{cd}	= 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c >$ (50 + 105 + 60) ps = 215 ps

$f_c = 1/T_c$ = 4.65 GHz

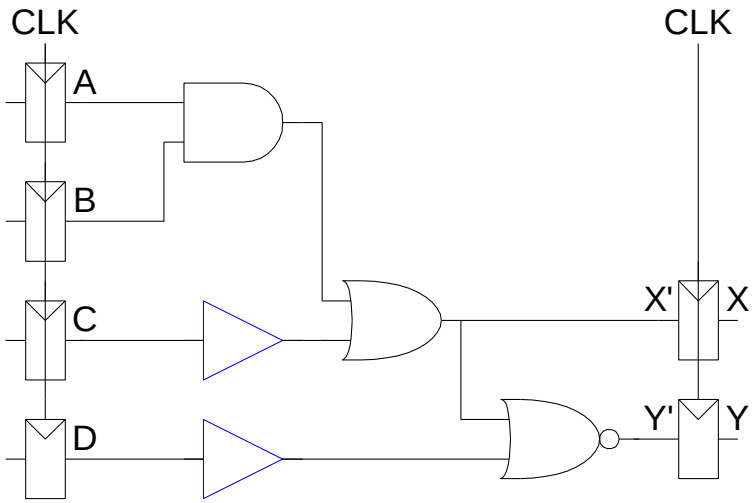
Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c > (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c =$ 4.65 GHz

Note: no change to max frequency!

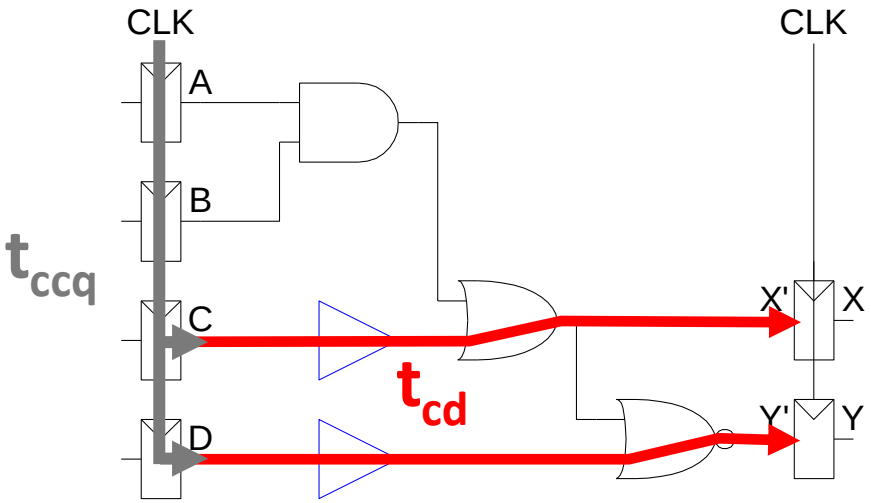
Check hold time constraint

$t_{ccq} + t_{cd} > t_{hold}$?

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate t_{pd}	= 35 ps
per gate t_{cd}	= 25 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c > (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c = 4.65$ GHz

Check hold time constraints:

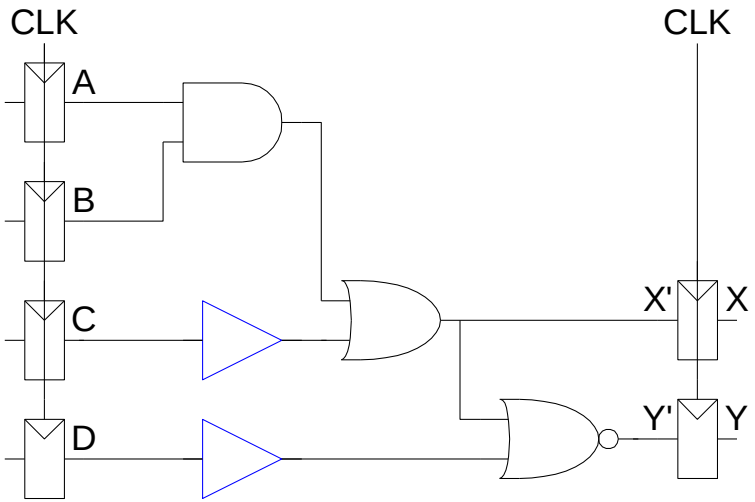
$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 50) ps > 70 ps ?

TIMING ANALYSIS: FIXING HOLD TIME VIOLATION

- Example

Add buffers to the short paths:



Timing Characteristics

t_{ccq}	= 30 ps
t_{pcq}	= 50 ps
t_{setup}	= 60 ps
t_{hold}	= 70 ps
per gate: t_{pd}	= 35 ps
per gate: t_{cd}	= 25 ps

t_{pd} = 3 x 35 ps = 105 ps

t_{cd} = 2 x 25 ps = 50 ps

Check setup time constraints:

$T_c > t_{pcq} + t_{pd} + t_{setup}$

$T_c > (50 + 105 + 60)$ ps = 215 ps

$f_c = 1/T_c = 4.65$ GHz

Check hold time constraints:

$t_{ccq} + t_{cd} > t_{hold}$?

(30 + 50) ps > 70 ps ?

PASS

CLOCK SKEW

- More timing issues
  - To make matters worse, clocks have delay too!
    - The clock does **not** reach all parts of the chip at the same time!
  - Clock skew: time difference between two clock edges