

---

# Electronics Systems

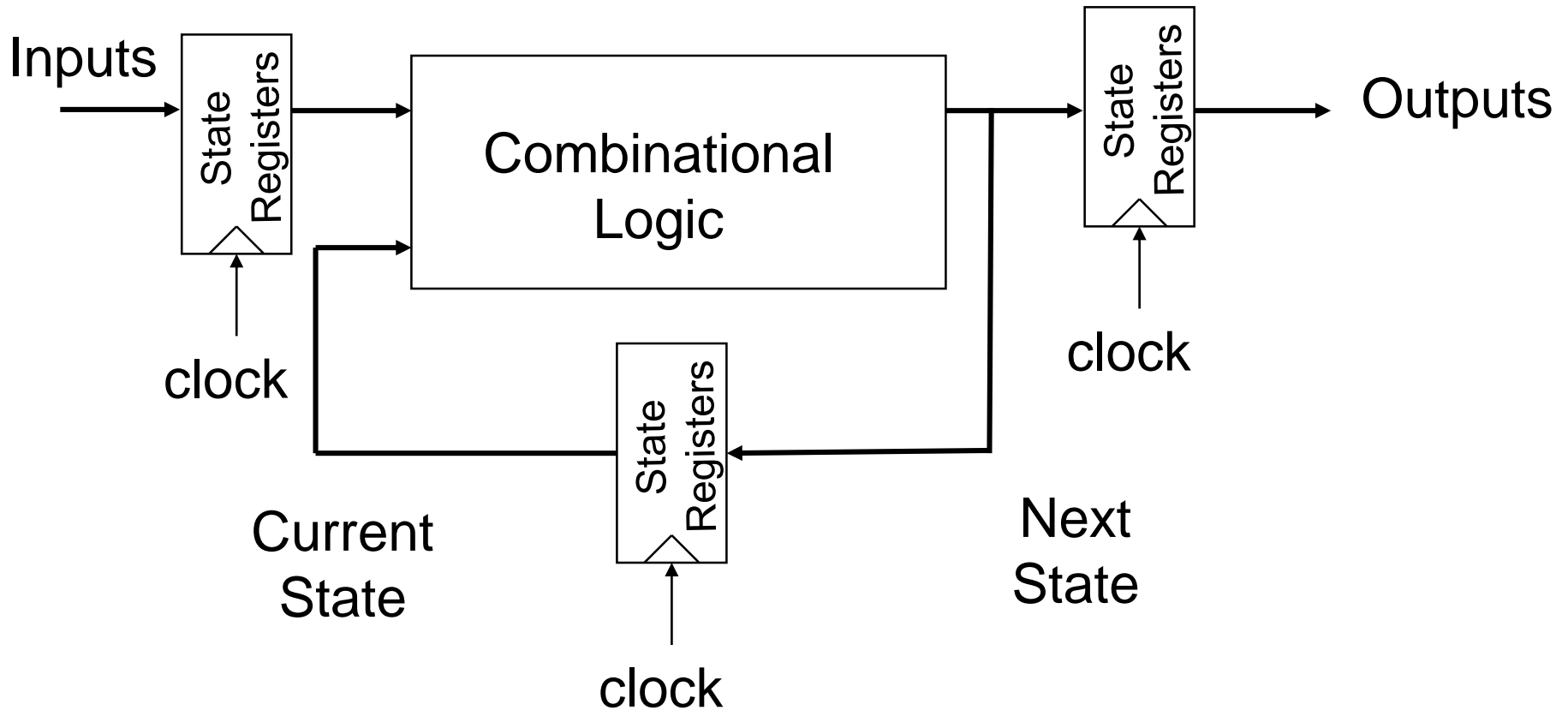
## Timing Issues

Luca Fanucci

[Adapted from Rabaey's *Digital Integrated Circuits*, Second Edition, ©2003  
J. Rabaey, A. Chandrakasan, B. Nikolic]

# Sequential Logic

---



# Static vs Dynamic Storage

---

## ❑ Static storage

- preserve state as long as the power is on
- have positive feedback (**regeneration**) with an internal connection between the output and the input
- useful when updates are infrequent (clock gating)

## ❑ Dynamic storage

- store state on parasitic capacitors
- only hold state for short periods of time (milliseconds)
- require periodic refresh
- usually simpler, so higher speed and lower power

# Latches vs Flipflops

---

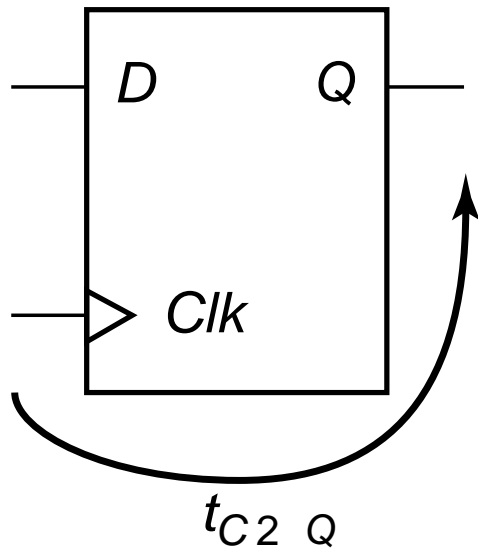
## ❑ Latches

- **level sensitive** circuit that passes inputs to Q when the clock is high (or low) - **transparent** mode
- input sampled on the falling edge of the clock is held stable when clock is low (or high) - **hold** mode

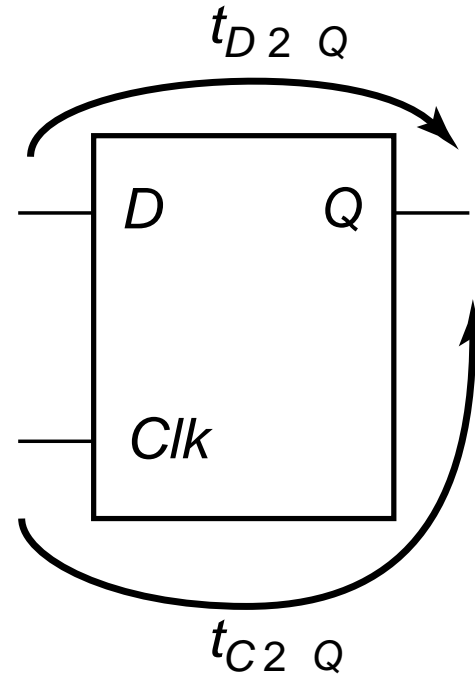
## ❑ Flipflops (edge-triggered)

- **edge sensitive** circuits that sample the inputs on a clock transition
  - positive edge-triggered:  $0 \rightarrow 1$
  - negative edge-triggered:  $1 \rightarrow 0$
- built using latches (e.g., master-slave flipflops)

# Characterizing Timing



Register



Latch

# Timing Classifications

---

## ❑ Synchronous systems

- All memory elements in the system are simultaneously updated using a globally distributed periodic synchronization signal (i.e., a global clock signal)
- Functionality is ensured by strict constraints on the clock signal generation and distribution to minimize
  - Clock skew (spatial variations in clock edges)
  - Clock jitter (temporal variations in clock edges)

## ❑ Asynchronous systems

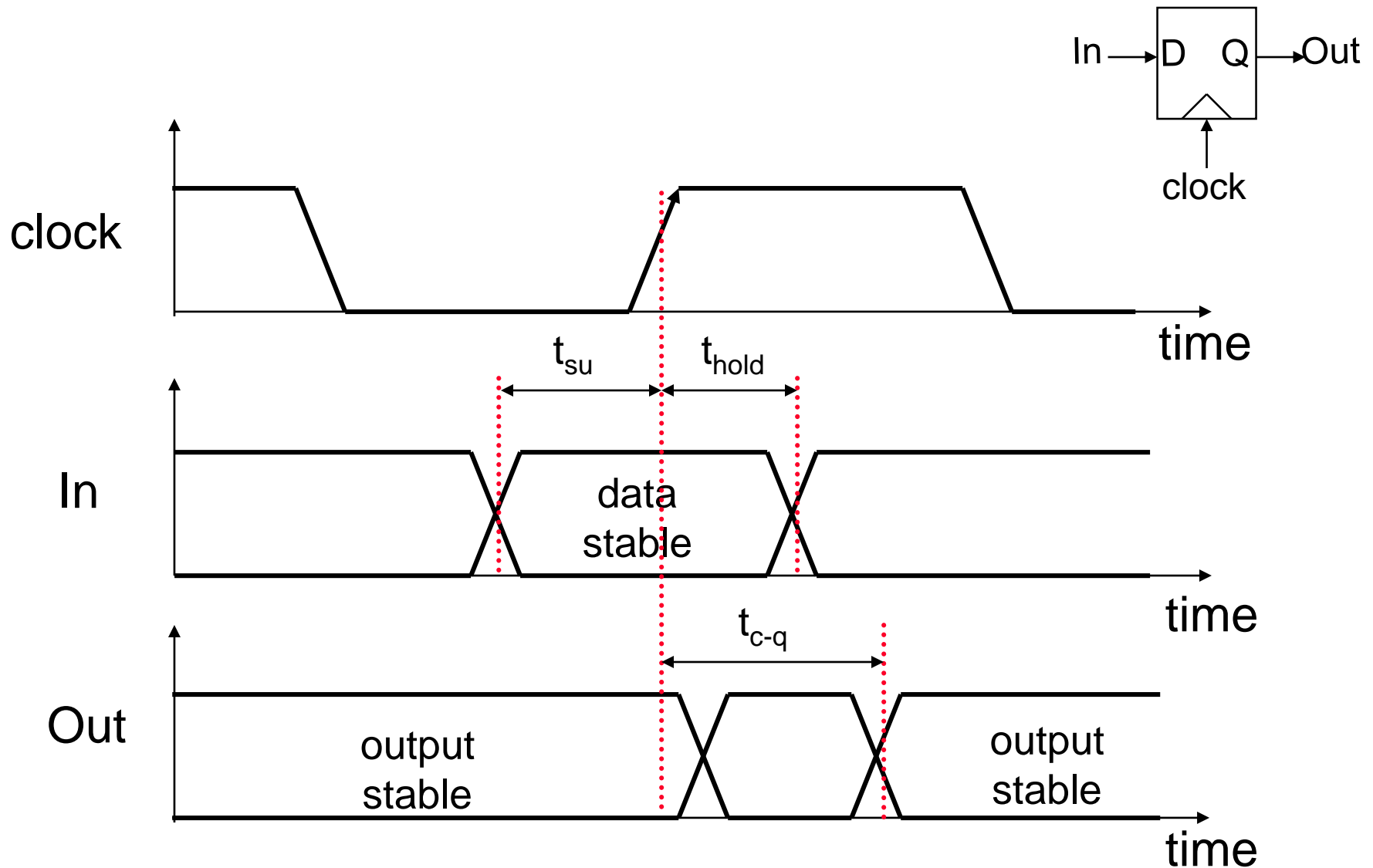
- Self-timed (controlled) systems
- No need for a globally distributed clock, but have asynchronous circuit overheads (handshaking logic, etc.)

## ❑ Hybrid systems

- Synchronization between different clock domains
- Interfacing between asynchronous and synchronous domains

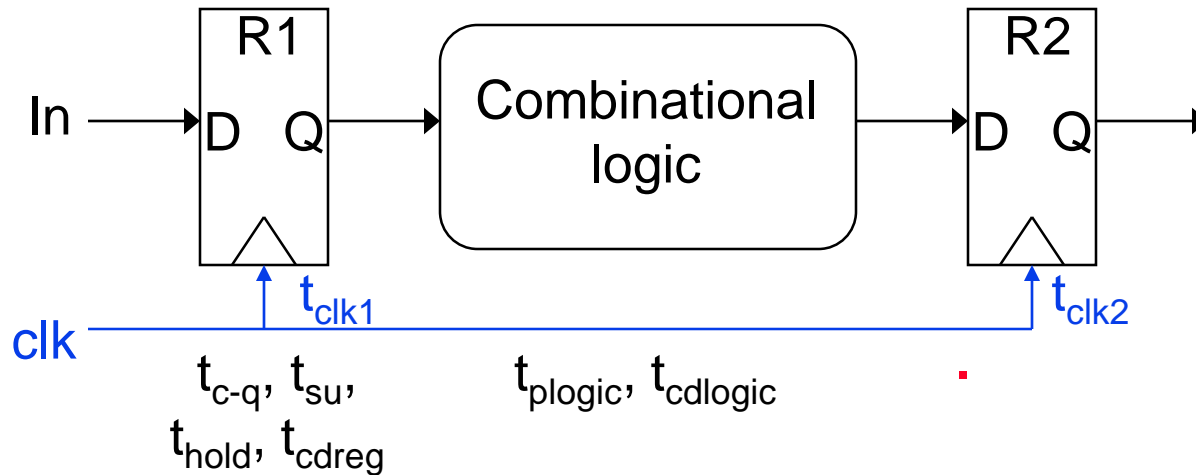


# Timing Metrics





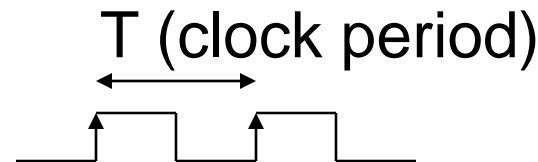
# Review: Synchronous Timing Basics



- Under ideal conditions (i.e., when  $t_{clk1} = t_{clk2}$ )

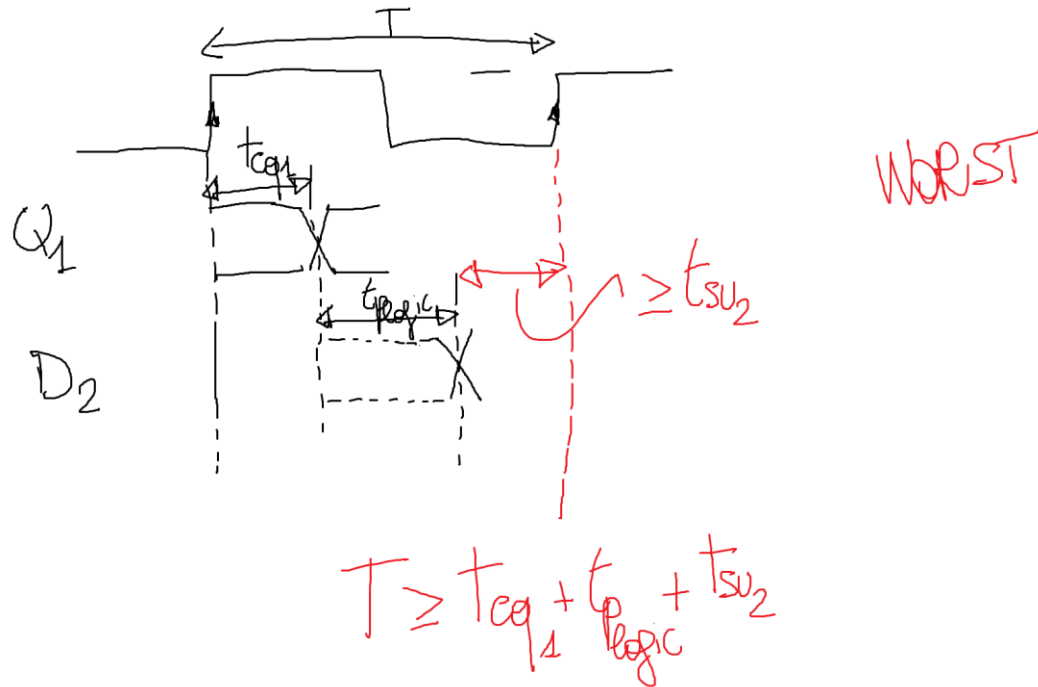
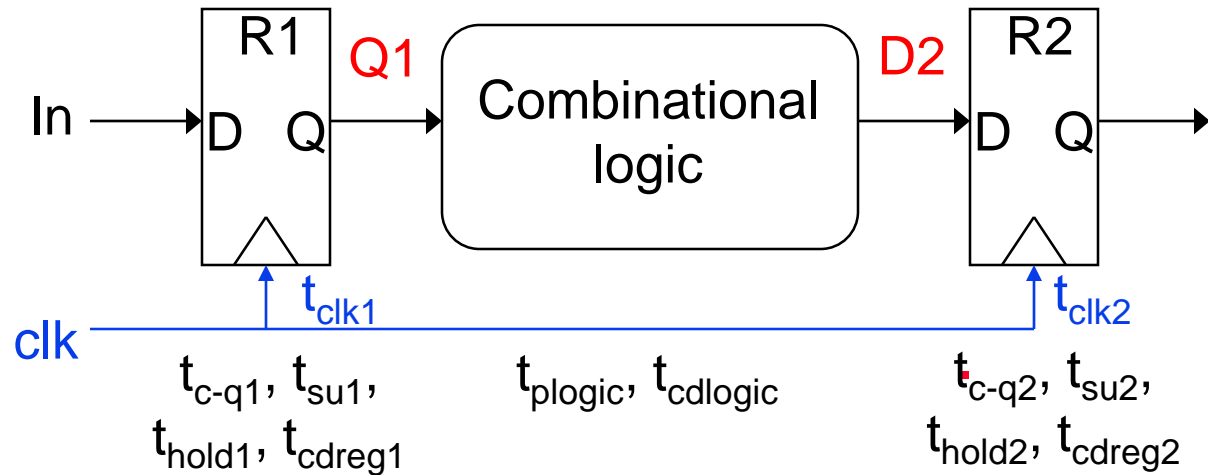
$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

$$t_{hold} \leq t_{cdlogic} + t_{cdreg}$$

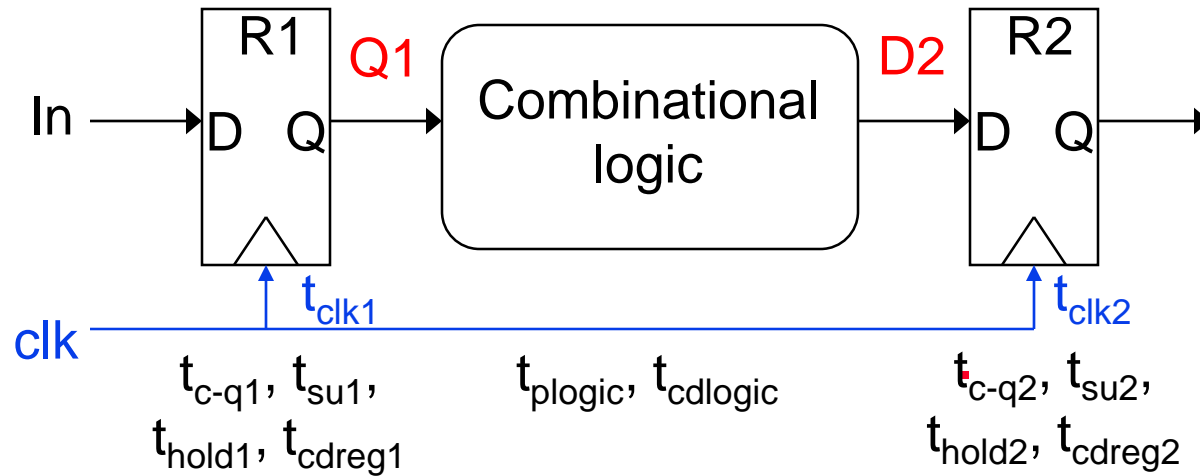


- the **contamination delay** is the minimum amount of time from when an input changes until any output starts to change its value. This change in value does not imply that the value has reached a stable condition.

# Set-Up Time Violation

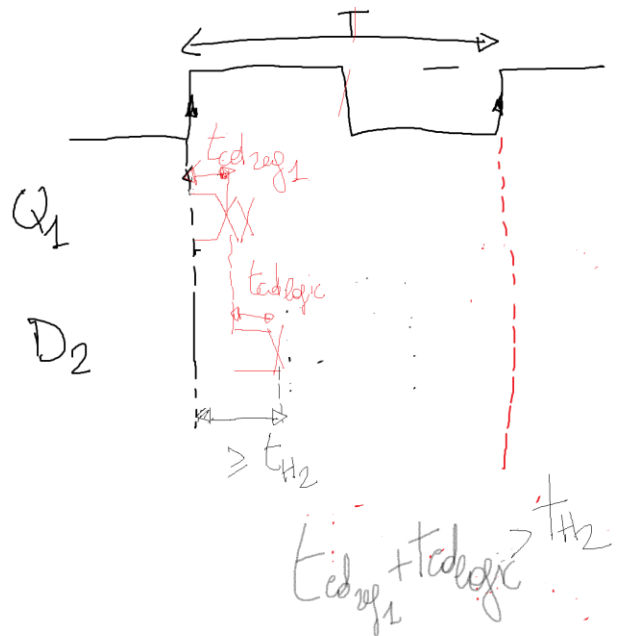
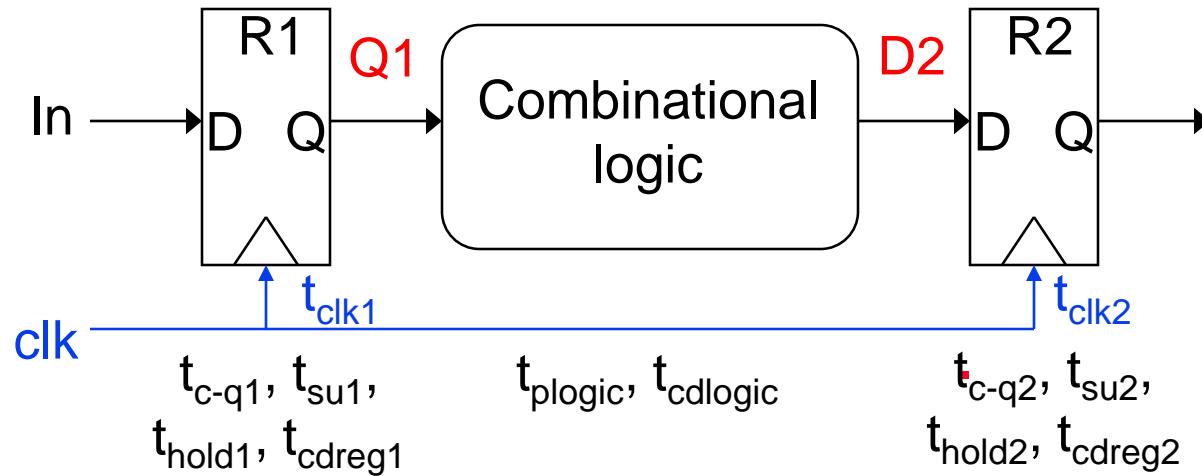


# Set-Up Time Violation

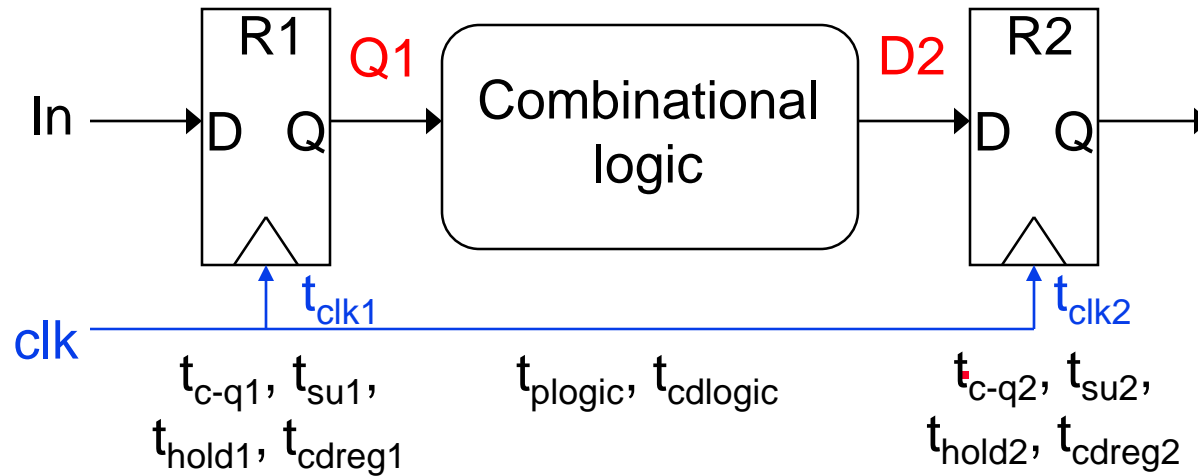


$$T \geq t_{c-q1} + t_{plogic} + t_{su1}$$

# Hold Time Violation

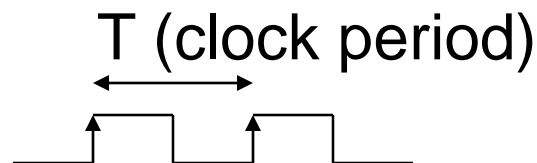
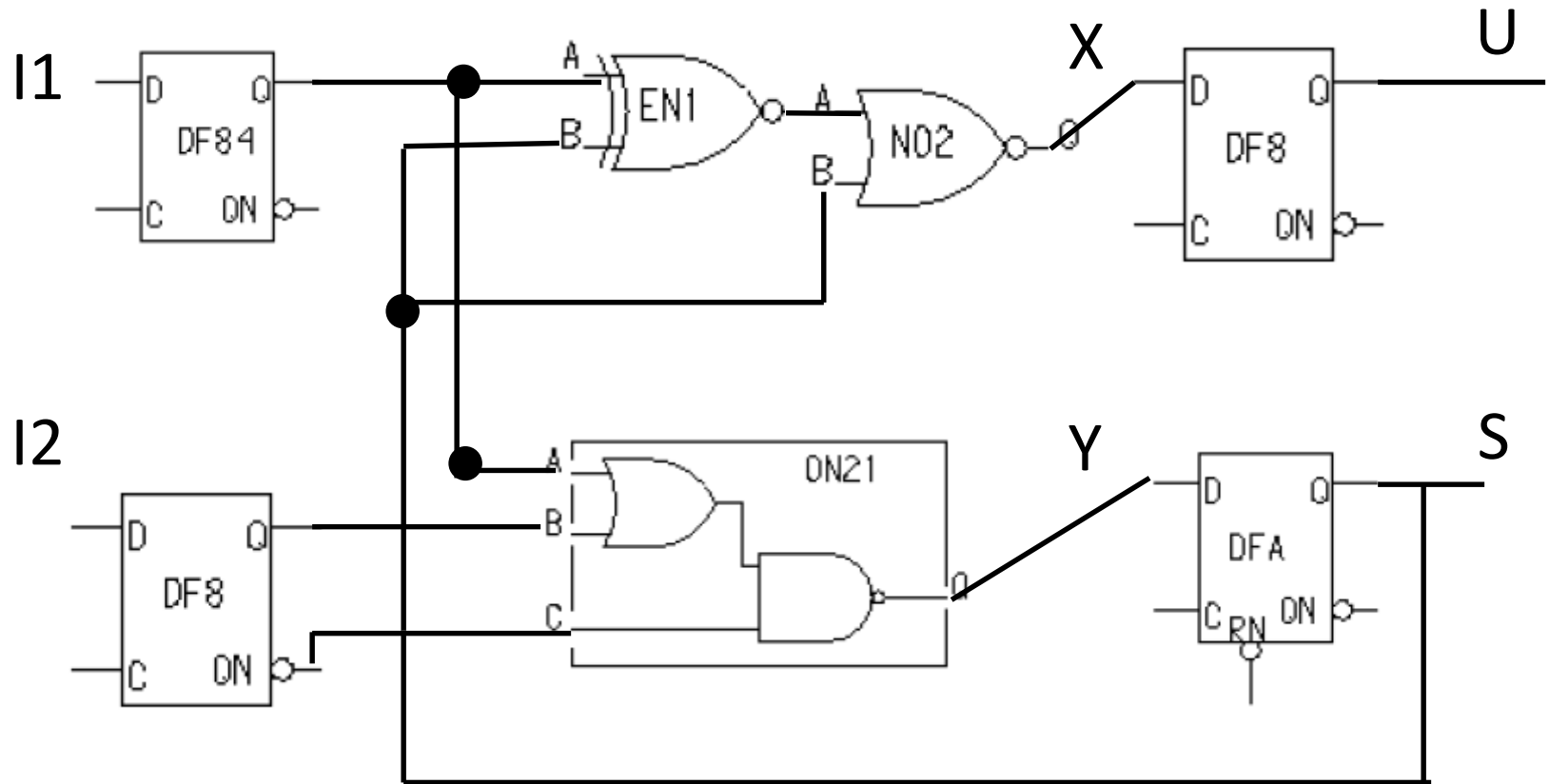


# Hold Time Violation



$$t_{hold2} \leq t_{cdlogic} + t_{cdreg1}$$

# Example for evaluating minimum Clock Period

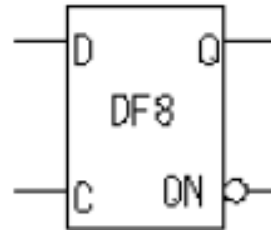


$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

DF8 is a static, master-slave D flip-flop with 1x drive strength.

### Truth Table

C	D	Q	QN
$\uparrow$	0	0	1
$\uparrow$	1	1	0



### Capacitance

Pin	Cap [pF]
C	0,010
D	0,010

### Area

0.423 mils<sup>2</sup>  
273  $\mu\text{m}^2$

### Power

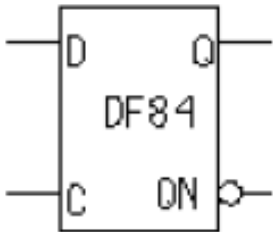
1.736  $\mu\text{W}/\text{MHz}$

Load	0,015	0,020	0,035
Delay C->Q	0,75	0,80	1,0
Delay C->QN	0,74	0,79	0,99
	Setup	Hold	
D->C	0,17	0,04	

DF84 is a static, master-slave D flip-flop with 4x drive strength.

Truth Table

C	D	Q	QN
↑	0	0	1
↑	1	1	0



Capacitance

Pin	Cap [pF]
C	0,010
D	0,010

Area

0.48 mils<sup>2</sup>  
310 μm<sup>2</sup>

Power

2.88 μW/MHz

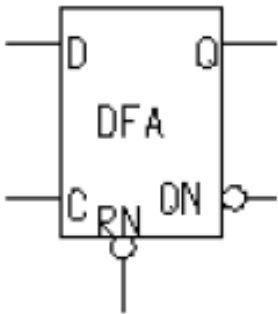
Load	0,015	0,020	0,035
Delay C->Q	0,65	0,70	0,90
Delay C->QN	0,64	0,69	0,89
	Setup	Hold	
D->C	0,17	0,04	



DFA is a static, master-slave D flip-flop with 1x drive strength. RESET is asynchronous and active low.

Truth Table

C	D	RN	Q	QN
↑	0	1	0	1
↑	1	1	1	0
X	X	0	0	1



Capacitance

Pin	Cap [pF]
C	0,010
D	0,010
RN	0,020

Area

0.536 mils<sup>2</sup>  
346 μm<sup>2</sup>

Power

1.773 μW/MHz

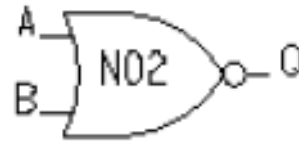
	Setup	Hold
D->C	0,16	0,04
RN->C	0,01	0,73

Load	0,015	0,020	0,035
Delay C->Q	0,84	0,90	1,2
Delay C->QN	0,84	0,89	1,1

NO2 is a 2-input NOR gate with 1x drive strength.

**Truth Table**

A	B	Q
0	0	1
X	1	0
1	X	0



**Capacitance**

Pin	Cap [pF]
A	0,015
B	0,015

**Area**

0.085 mils<sup>2</sup>  
55  $\mu\text{m}^2$

**Power**

0.422  $\mu\text{W}/\text{MHz}$

Load [pF]	0,010	0,015	0,020
Delay A->Q [ns]	0,13	0,24	0,31
Delay B->Q [ns]	0,11	0,23	0,29

EN1 is a 2-input EXCLUSIVE-NOR (XNOR) gate with 1x drive strength.

Truth Table

A	B	Q
0	0	1
0	1	0
1	0	0
1	1	1



Capacitance

Pin	Cap [pF]
A	0,020
B	0,020

Area

0.169 mils<sup>2</sup>  
109 μm<sup>2</sup>

Power

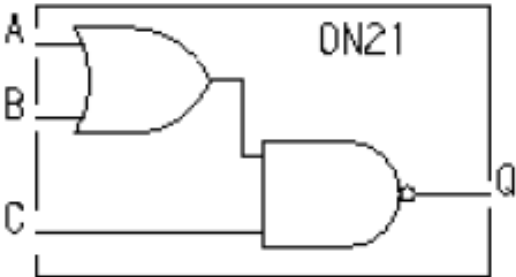
0.576 μW/MHz

Load [pF]	0,010	0,015	0,020
Delay A->Q [ns]	0,28	0,35	0,40
Delay B->Q [ns]	0,27	0,33	0,39

ON21 is an OR / NAND circuit providing the logical function  $Q = \text{NOT} [(A+B).C]$ .

Truth Table

A	B	C	Q
0	0	X	1
X	X	0	1
X	1	1	0
1	X	1	0



Capacitance

Pin	Cap [pF]
A	0,015
B	0,015
C	0,015

Area

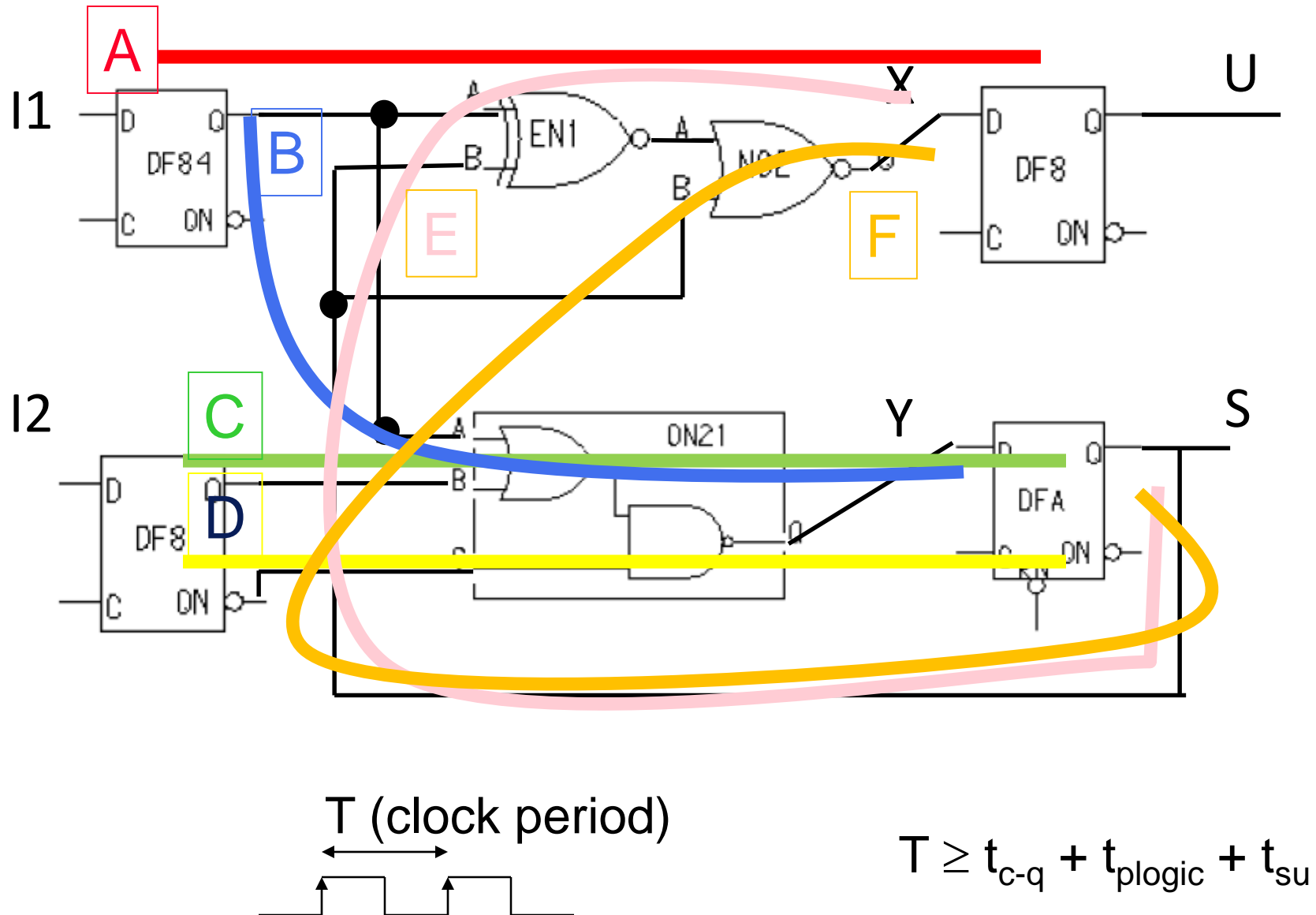
0.141 mils<sup>2</sup>  
91 μm<sup>2</sup>

Power

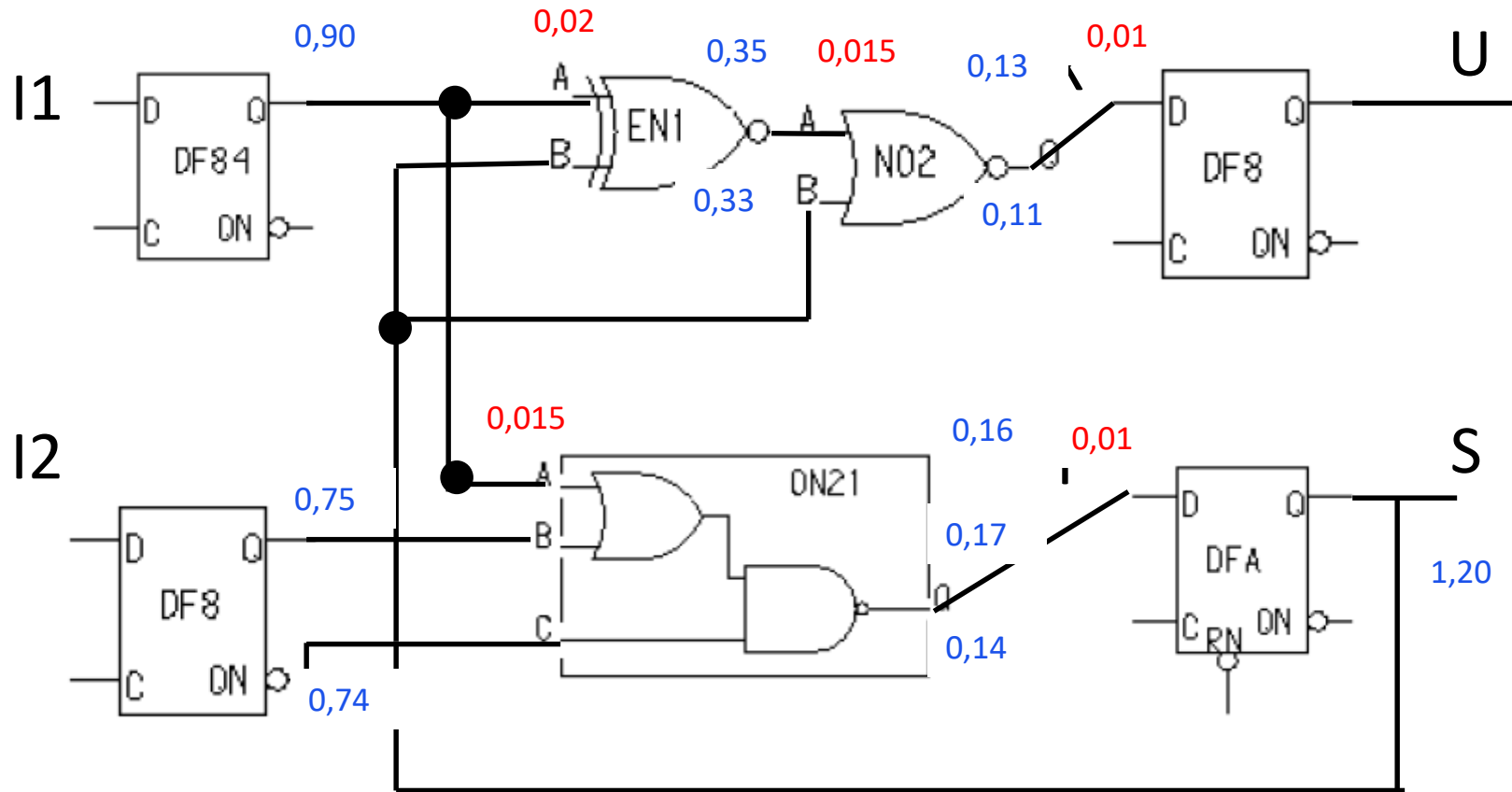
0.446 μW/MHz

Load [pF]	0,010	0,015	0,020
Delay A->Q [ns]	0,16	0,23	0,34
Delay B->Q [ns]	0,17	0,24	0,35
Delay C->Q [ns]	0,14	0,20	0,30

# Example for evaluating minimum Clock Period

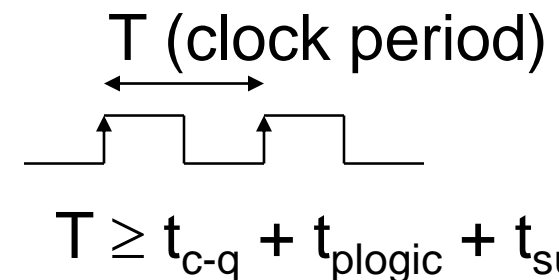


# Example for evaluating minimum Clock Period

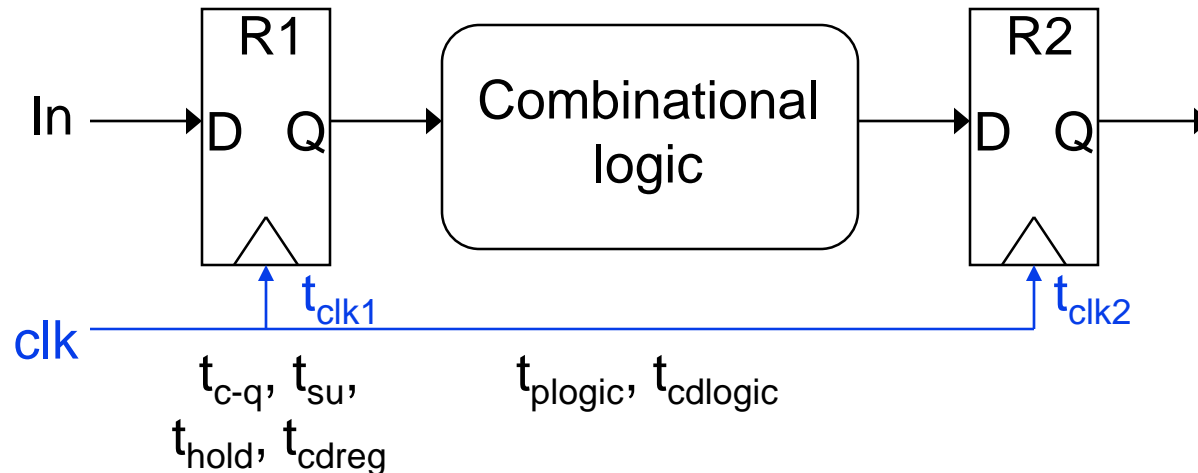


A	0,9	0,35	0,13	0,17	1,55
B	0,9	0,16		0,16	1,22
C	0,75	0,17		0,16	1,08
D	0,74	0,14		0,16	1,04
E	1,2	0,11		0,17	1,48
F	1,2	0,33	0,13	0,17	1,83

546,45 MHz



# Review: Synchronous Timing Basics



- Under ideal conditions (i.e., when  $t_{clk1} = t_{clk2}$ )

$$T \geq t_{c-q} + t_{plogic} + t_{su}$$

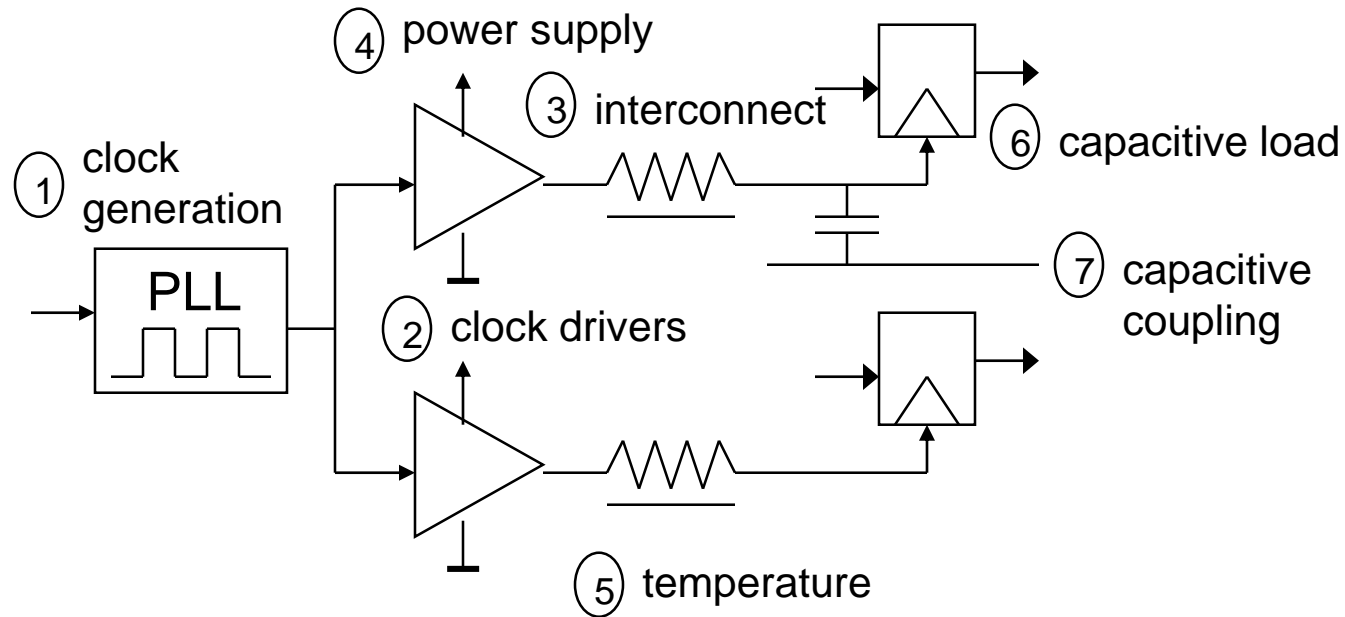
$$t_{hold} \leq t_{cdlogic} + t_{cdreg}$$

- Under real conditions, the clock signal can have both spatial (**clock skew**) and temporal (**clock jitter**) variations
  - skew is constant from cycle to cycle (by definition); skew can be positive (clock and data flowing in the same direction) or negative (clock and data flowing in opposite directions)
  - jitter causes  $T$  to change on a cycle-by-cycle basis

# **Review: Synchronous Timing Basics**



# Sources of Clock Skew and Jitter in Clock Network



## ❑ Skew

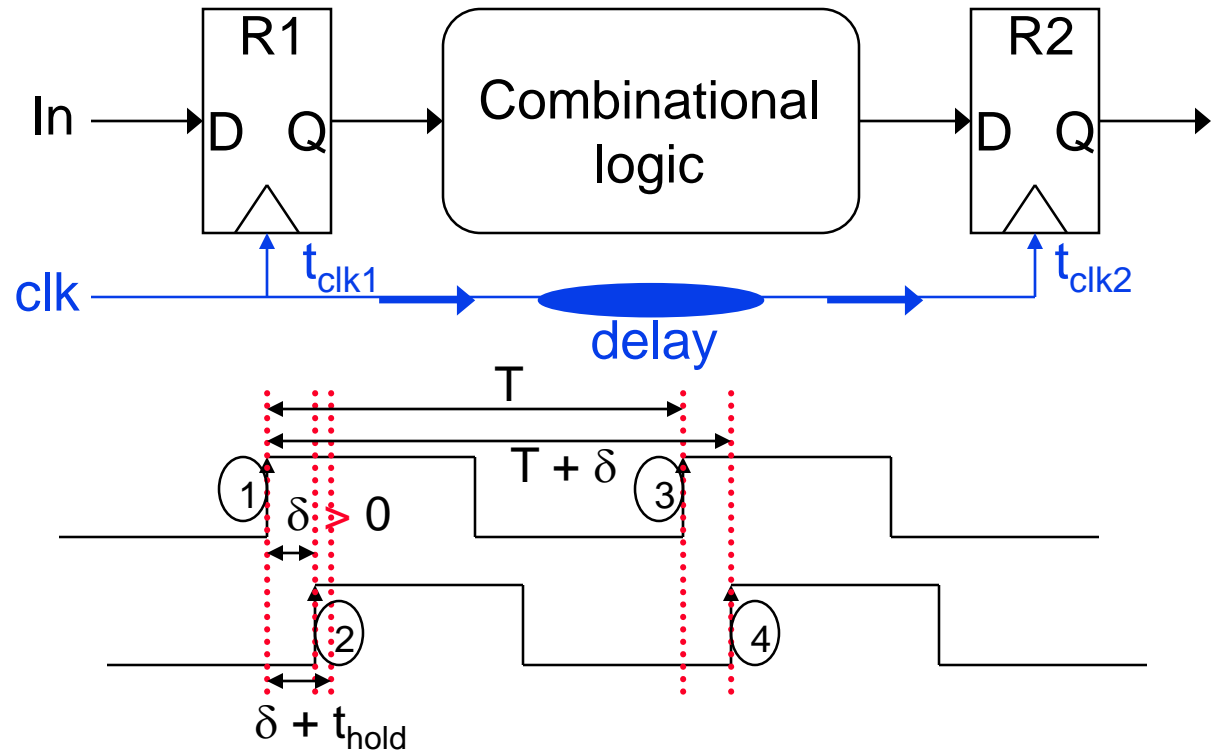
- manufacturing device variations in clock drivers
- interconnect variations
- environmental variations (power supply and temperature)

## ❑ Jitter

- clock generation
- capacitive loading and coupling
- environmental variations (power supply and temperature)

# Positive Clock Skew

- ❑ Clock and data flow in the same direction

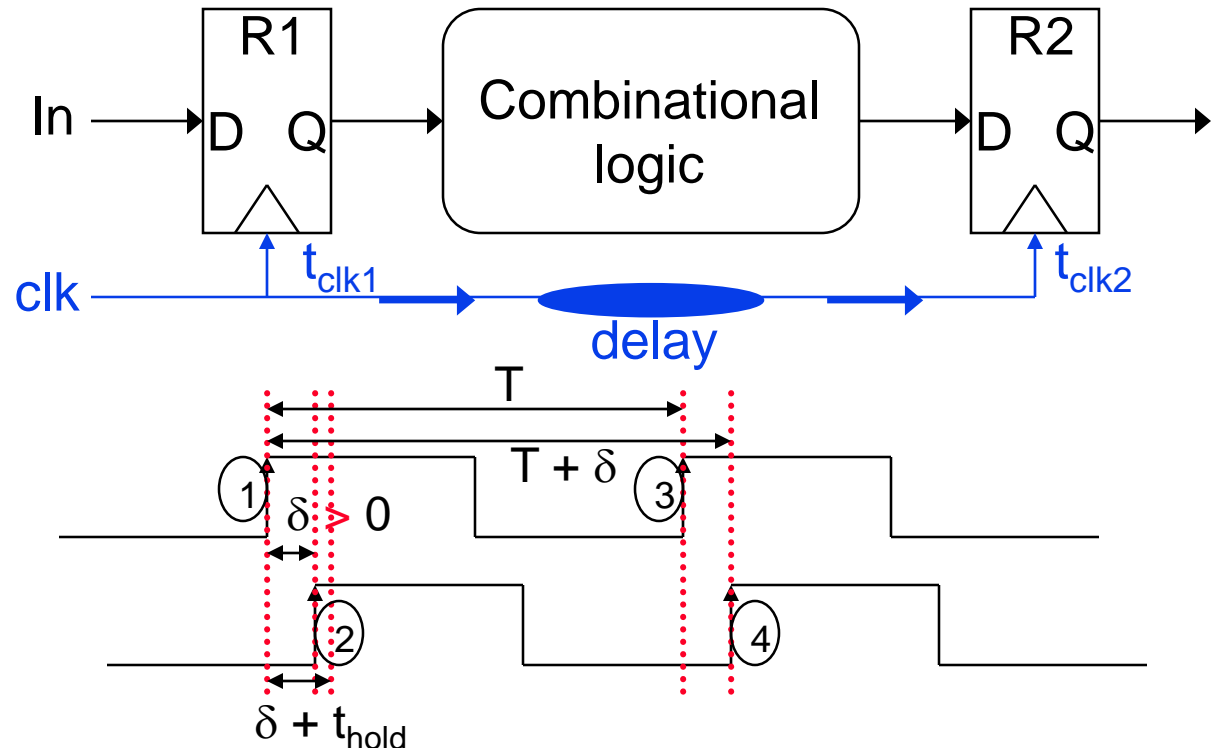


$T$  :

$t_{hold}$  :

# Positive Clock Skew

- Clock and data flow in the same direction



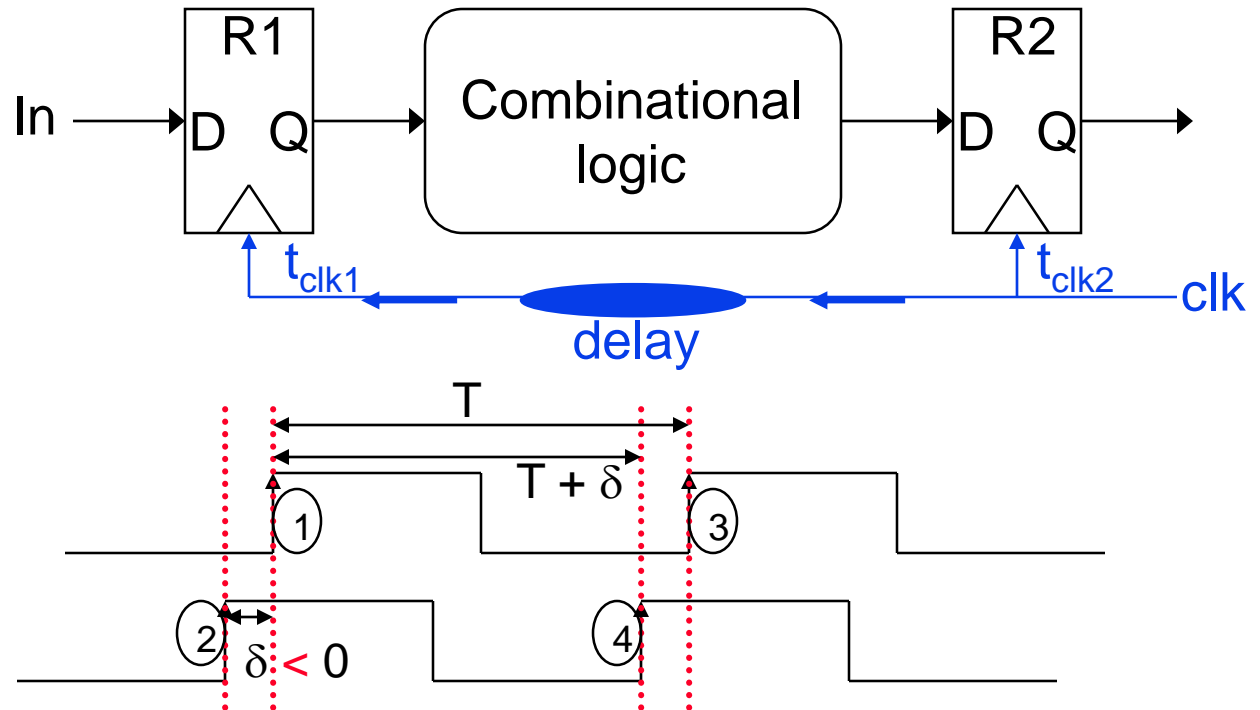
$$T: \quad T + \delta \geq t_{c-q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c-q} + t_{plogic} + t_{su} - \delta$$

$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

- $\delta > 0$ : Improves performance, but makes  $t_{hold}$  harder to meet. If  $t_{hold}$  is not met (race conditions), the circuit malfunctions independent of the clock period!

# Negative Clock Skew

- ❑ Clock and data flow in opposite directions

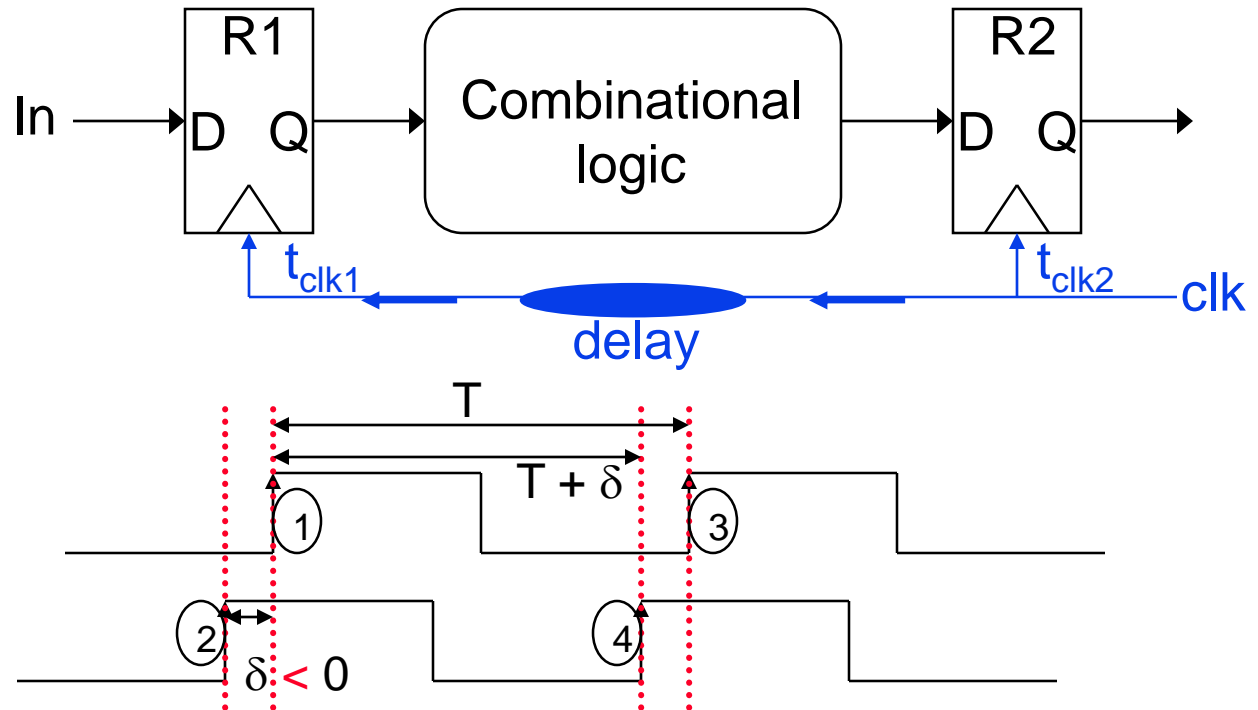


$T$  :

$t_{hold}$  :

# Negative Clock Skew

- ❑ Clock and data flow in opposite directions



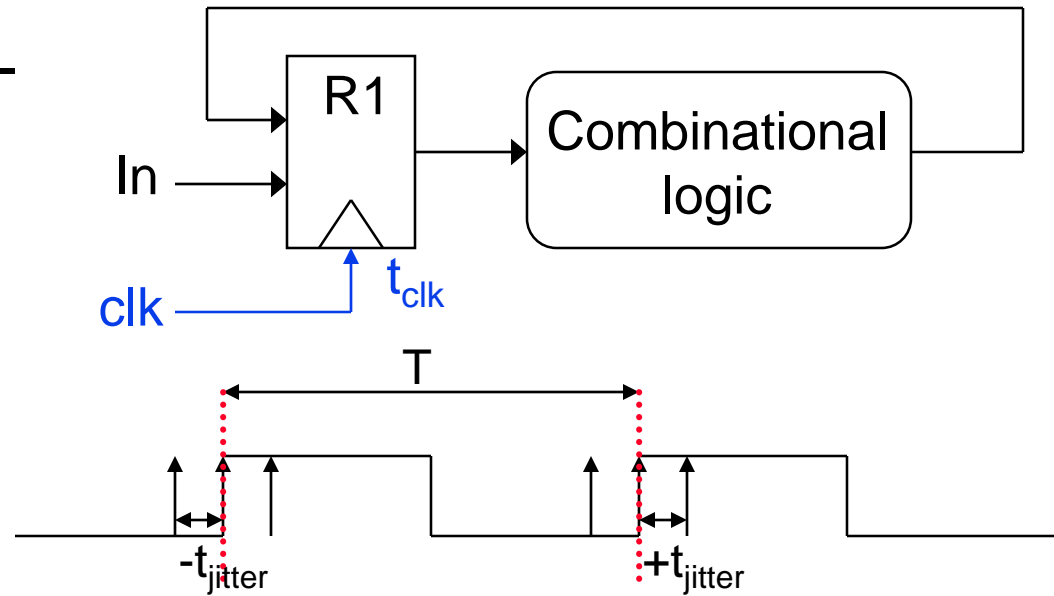
$$T: \quad T + \delta \geq t_{c-q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c-q} + t_{plogic} + t_{su} - \delta$$

$$t_{hold}: \quad t_{hold} + \delta \leq t_{cdlogic} + t_{cdreg} \quad \text{so} \quad t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta$$

- ❑  $\delta < 0$ : Degrades performance, but  $t_{hold}$  is easier to meet (eliminating race conditions)

# Clock Jitter

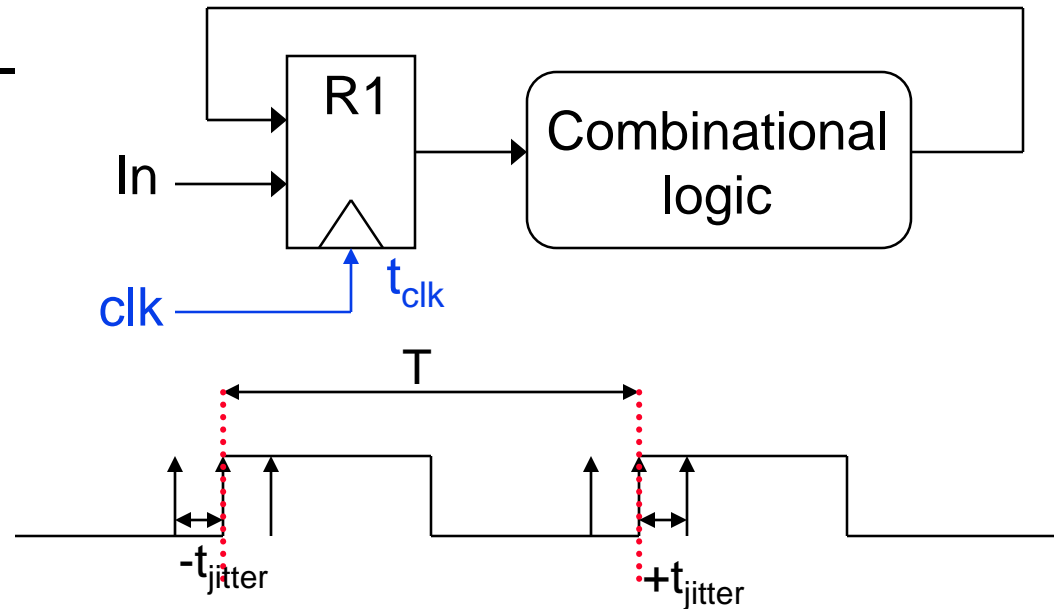
- ❑ Jitter causes  $T$  to vary on a cycle-by-cycle basis



$T$  :

# Clock Jitter

- ❑ Jitter causes  $T$  to vary on a cycle-by-cycle basis

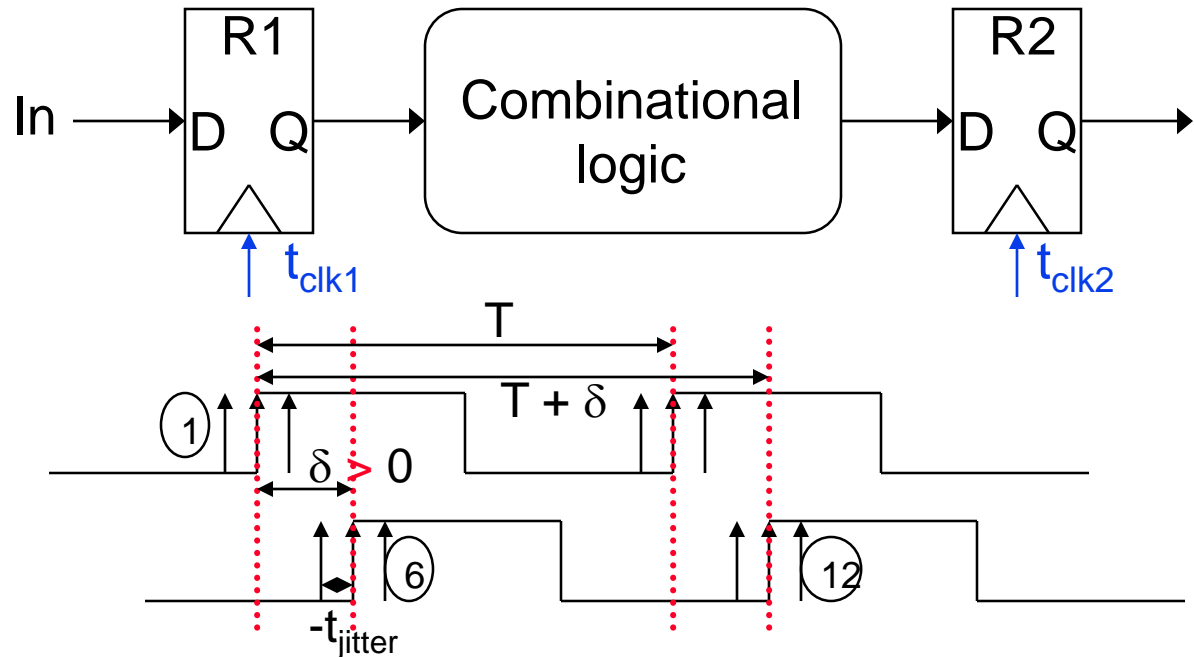


$$T : T - 2t_{jitter} \geq t_{c-q} + t_{plogic} + t_{su} \quad \text{so} \quad T \geq t_{c-q} + t_{plogic} + t_{su} + 2t_{jitter}$$

- ❑ Jitter directly reduces the performance of a sequential circuit

# Combined Impact of Skew and Jitter

- Constraints on the minimum clock period ( $\delta > 0$ )



$$T \geq t_{c-q} + t_{plogic} + t_{su} - \delta + 2t_{jitter}$$

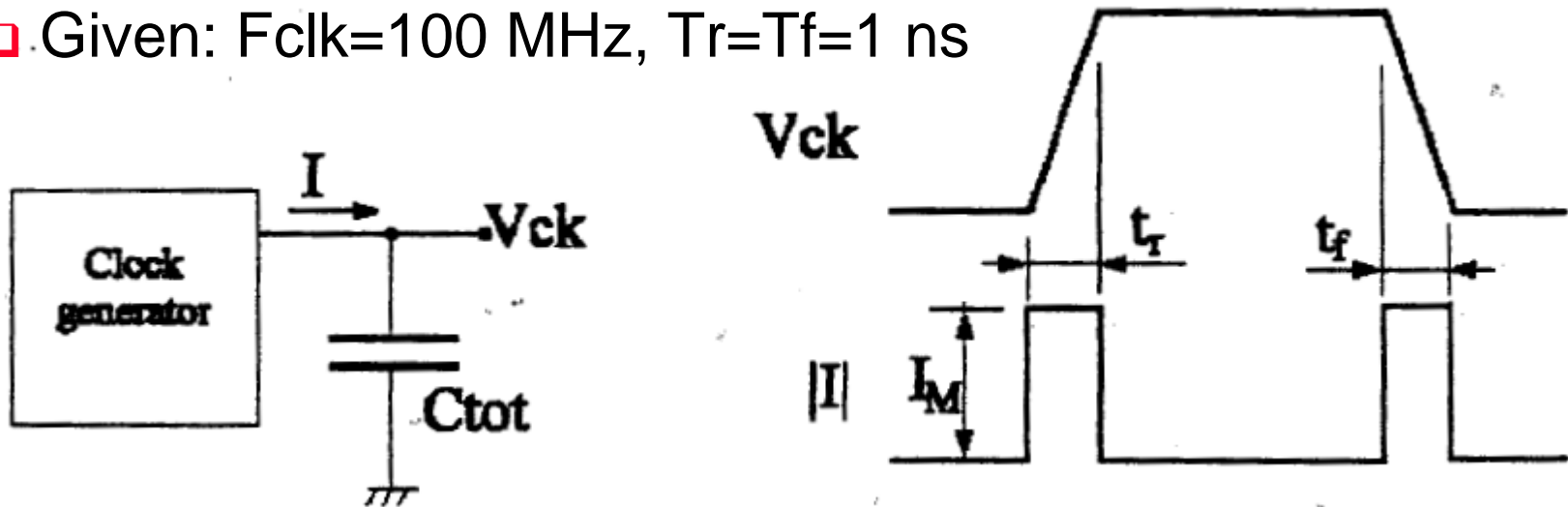
$$t_{hold} \leq t_{cdlogic} + t_{cdreg} - \delta - 2t_{jitter}$$

- $\delta > 0$  with jitter: Degrades performance, and makes  $t_{hold}$  even *harder* to meet. (The acceptable skew is reduced by jitter.)



# Example of a Clock Generator

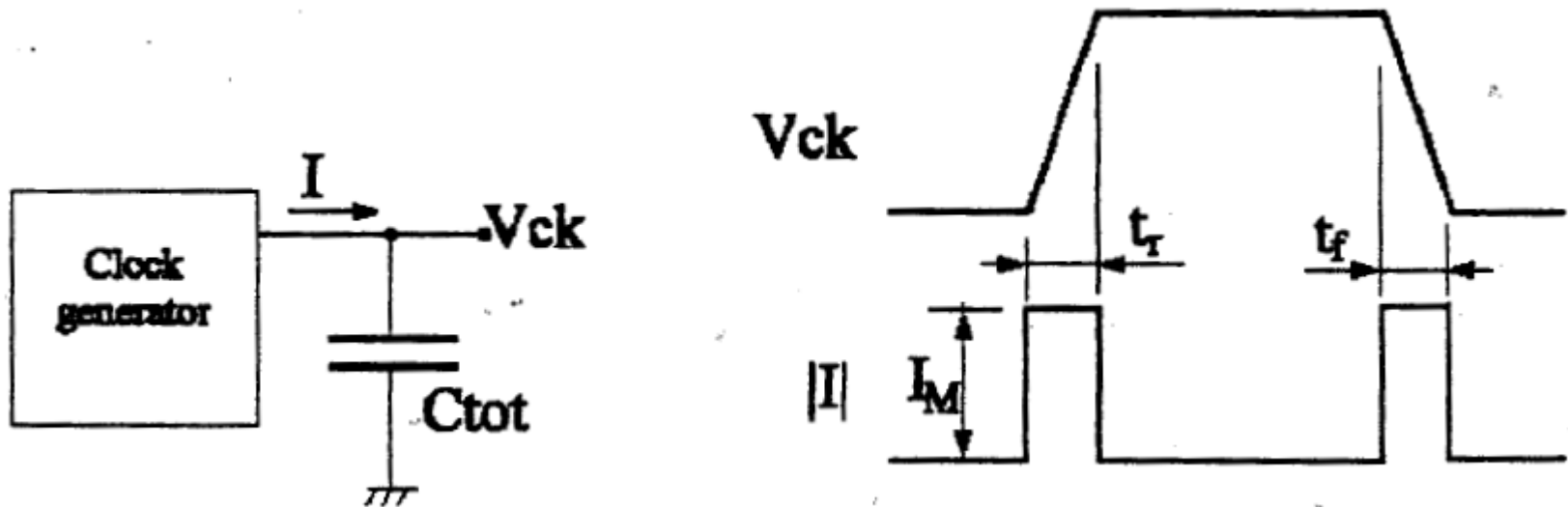
- ❑ Chip with 10.000 FF
- ❑ Each FF has a  $C_{in}=50$  fF for the clock input
- ❑ Clock Generator needs to drive  $C_{tot}= 500$ pF
- ❑ Given:  $F_{clk}=100$  MHz,  $T_r=T_f=1$  ns



$$I_M = C_{tot} \cdot \left. \frac{\partial V_{ck}}{\partial t} \right|_{MAX} = C \frac{V_{dd}}{t_r} \cong 2.5 A$$

# Example of a Clock Generator

- ❑ High Power consumption
- ❑ Supply line noise
- ❑ High current density in interconnection (elettromigration !)



$$I_M = C_{tot} \cdot \left. \frac{\partial V_{ck}}{\partial t} \right|_{MAX} = C \frac{V_{dd}}{t_r} \cong 2.5 A$$

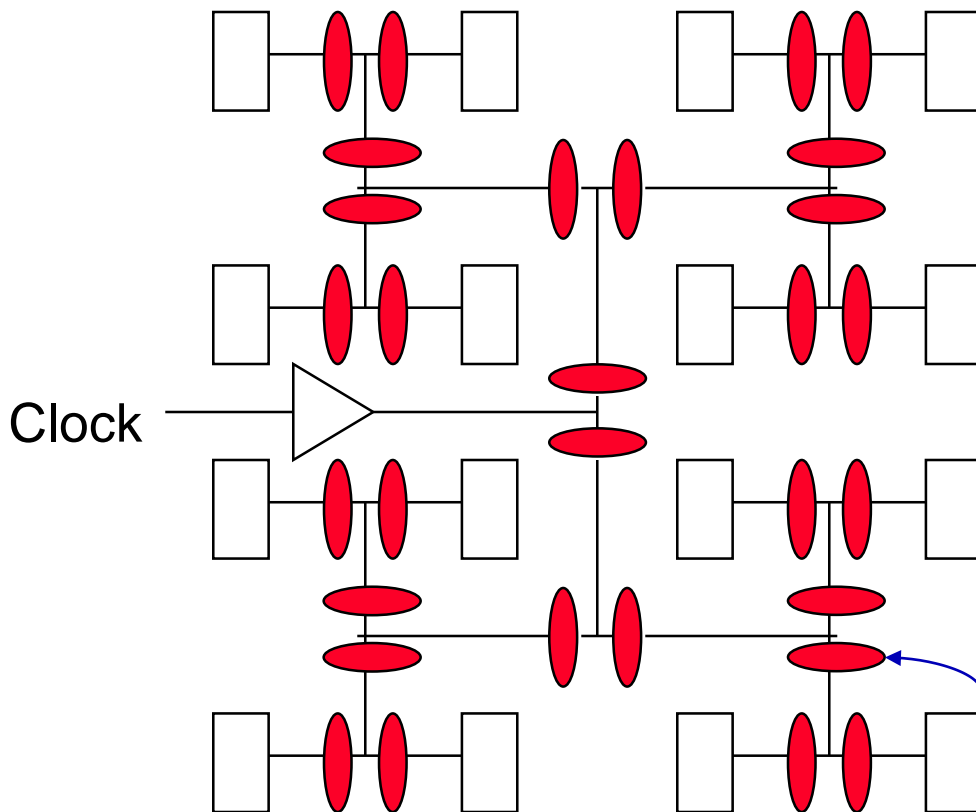
# Clock Distribution Networks

---

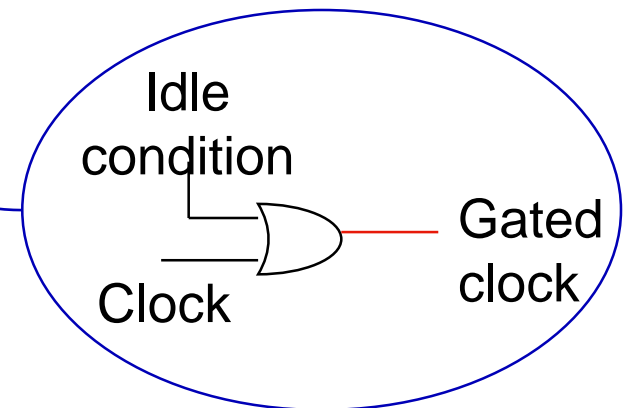
- ❑ Clock skew and jitter can ultimately limit the performance of a digital system, so designing a **clock network** that minimizes both is important
  - In many high-speed processors, a majority of the dynamic power is dissipated in the clock network.
  - To reduce dynamic power, the clock network must support clock gating (shutting down (disabling the clock) units)
- ❑ **Clock distribution techniques**
  - **Balanced paths** (H-tree network, matched RC trees)
    - In the ideal case, can eliminate skew
    - Could take multiple cycles for the clock signal to propagate to the leaves of the tree
  - **Clock grids**
    - Typically used in the final stage of the clock distribution network
    - Minimizes absolute delay (not relative delay)

# H-Tree Clock Network

- ❑ If the paths are perfectly balanced, clock skew is zero

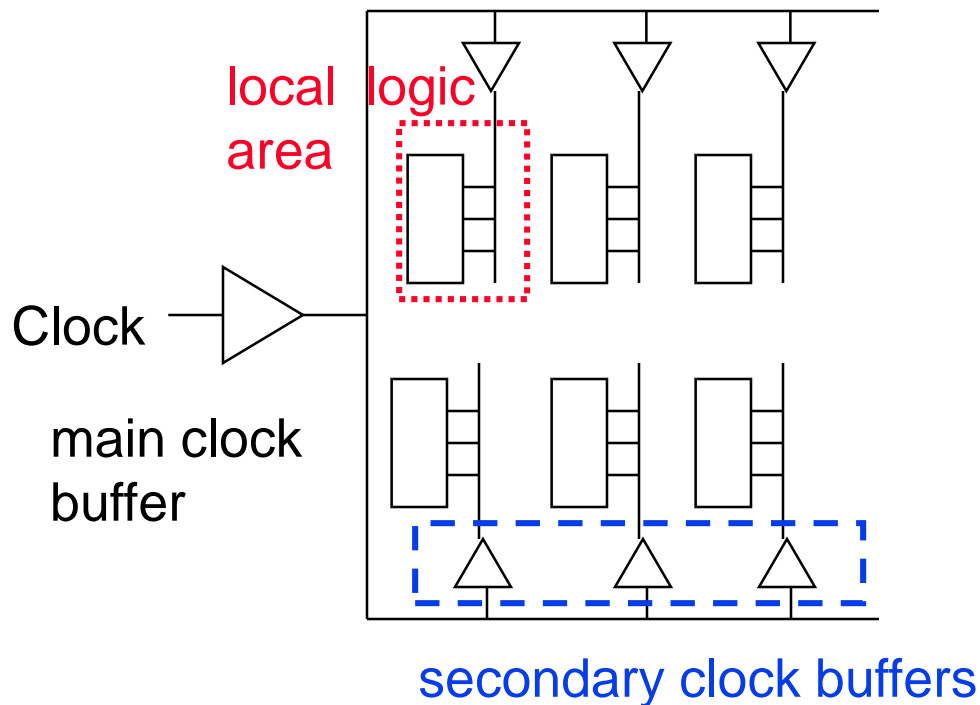


Can insert clock gating at multiple levels in clock tree  
Can shut off entire subtree if all gating conditions are satisfied



# Clock Grid Network

- ❑ Distributed buffering reduces absolute delay and makes clock gating easier, but is sensitive to variations in the buffer delay

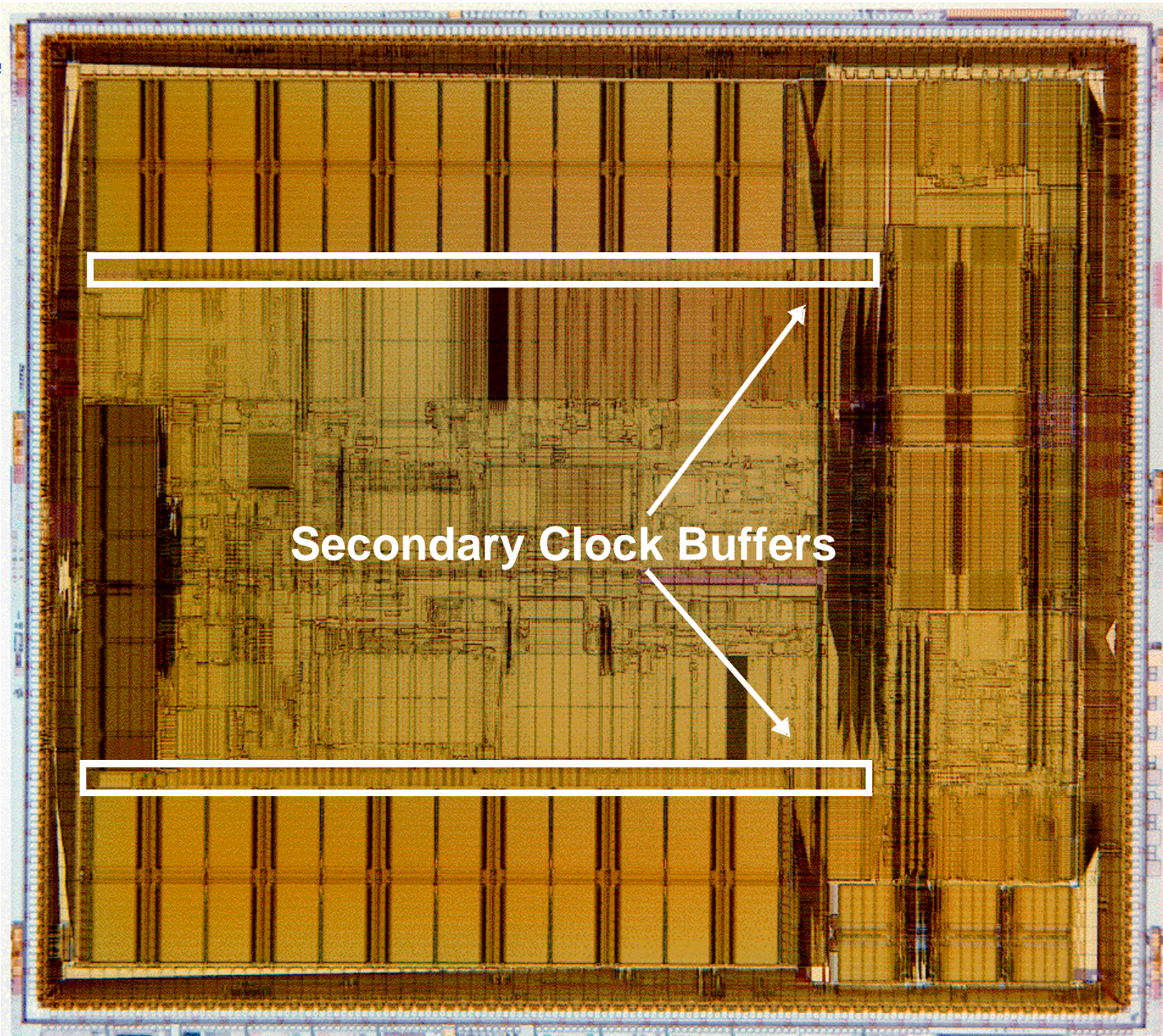


- ❑ The **secondary buffers** isolate the local clock nets from the upstream load and amplify the clock signals degraded by the RC network
  - decreases absolute skew
  - gives steeper clocks
- ❑ Only have to bound the skew within the **local logic area**

# DEC Alpha 21164 (EV5) Example

- ❑ 300 MHz clock (9.3 million transistors on a 16.5x18.1 mm die in 0.5 micron CMOS technology - 4 layer metal)
  - single phase clock
- ❑ 3.75 nF total clock load
  - Extensive use of dynamic logic
- ❑ 20 W (out of 50) in clock distribution network
- ❑ Two level clock distribution
  - Single 6 inverter stage main clock buffer at the center of the chip
  - Secondary clock buffers drive the left and right sides of the clock grid in m3 and m4
- ❑ Total equivalent driver size of 58 cm !!

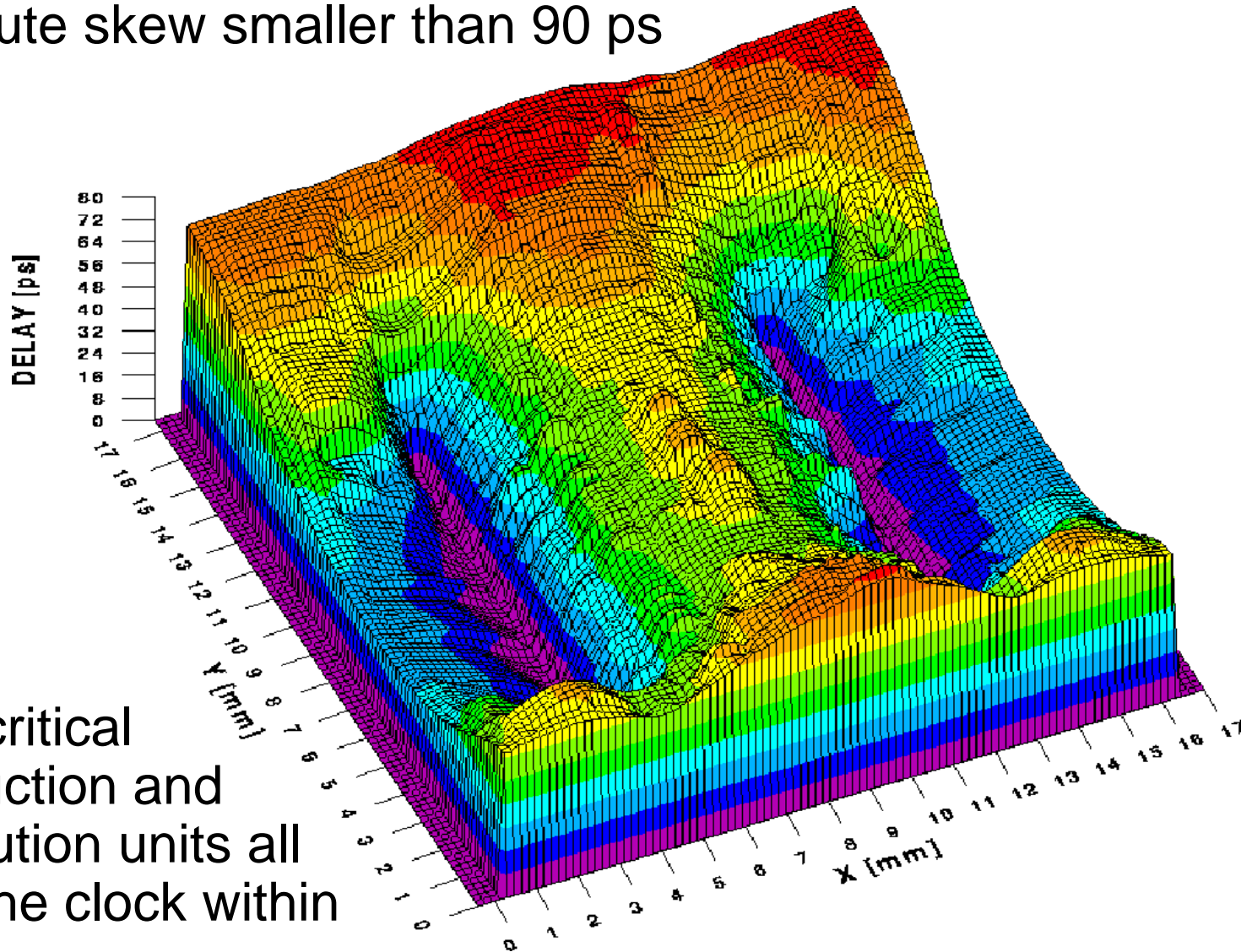






# Clock Skew in Alpha Processor

❑ Absolute skew smaller than 90 ps



❑ The critical instruction and execution units all see the clock within 65 ps



# Dealing with Clock Skew and Jitter

---

- ❑ To minimize skew, balance clock paths using *H-tree* or *matched-tree* clock distribution structures.
- ❑ If possible, route data and clock in opposite directions; eliminates races at the cost of performance.
- ❑ The use of gated clocks to help with dynamic power consumption make jitter worse.
- ❑ Shield clock wires (route power lines –  $V_{DD}$  or GND – next to clock lines) to minimize/eliminate coupling with neighboring signal nets.
- ❑ Use dummy fills to reduce skew by reducing variations in interconnect capacitances due to interlayer dielectric thickness variations.
- ❑ Beware of temperature and supply rail variations and their effects on skew and jitter. *Power supply noise fundamentally limits the performance of clock networks.*

# What is a PLL?

---

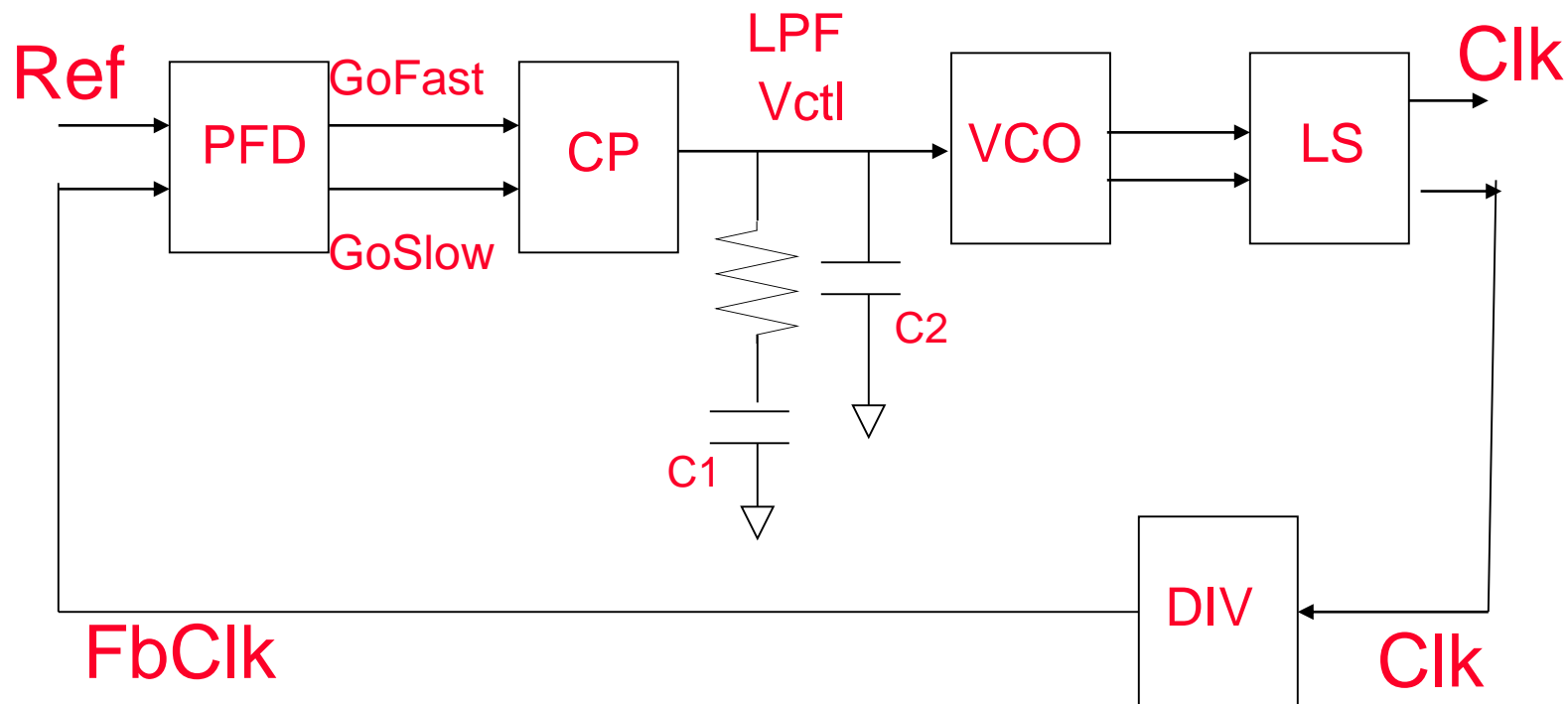
- ❑ A PLL is a negative feedback system where an oscillator-generated signal is phase and frequency locked to a reference signal.

# How are PLL's Used?

---

- ❑ Frequency Synthesis (e.g. generating a 1 GHz clock from a 100 MHz reference)
- ❑ Skew Cancellation (e.g. phase-aligning an internal clock to the IO clock) (May use a DLL instead)
- ❑ Extracting a clock from a random data stream (e.g. serial-link receiver)

# Charge-Pump PLL Block Diagram



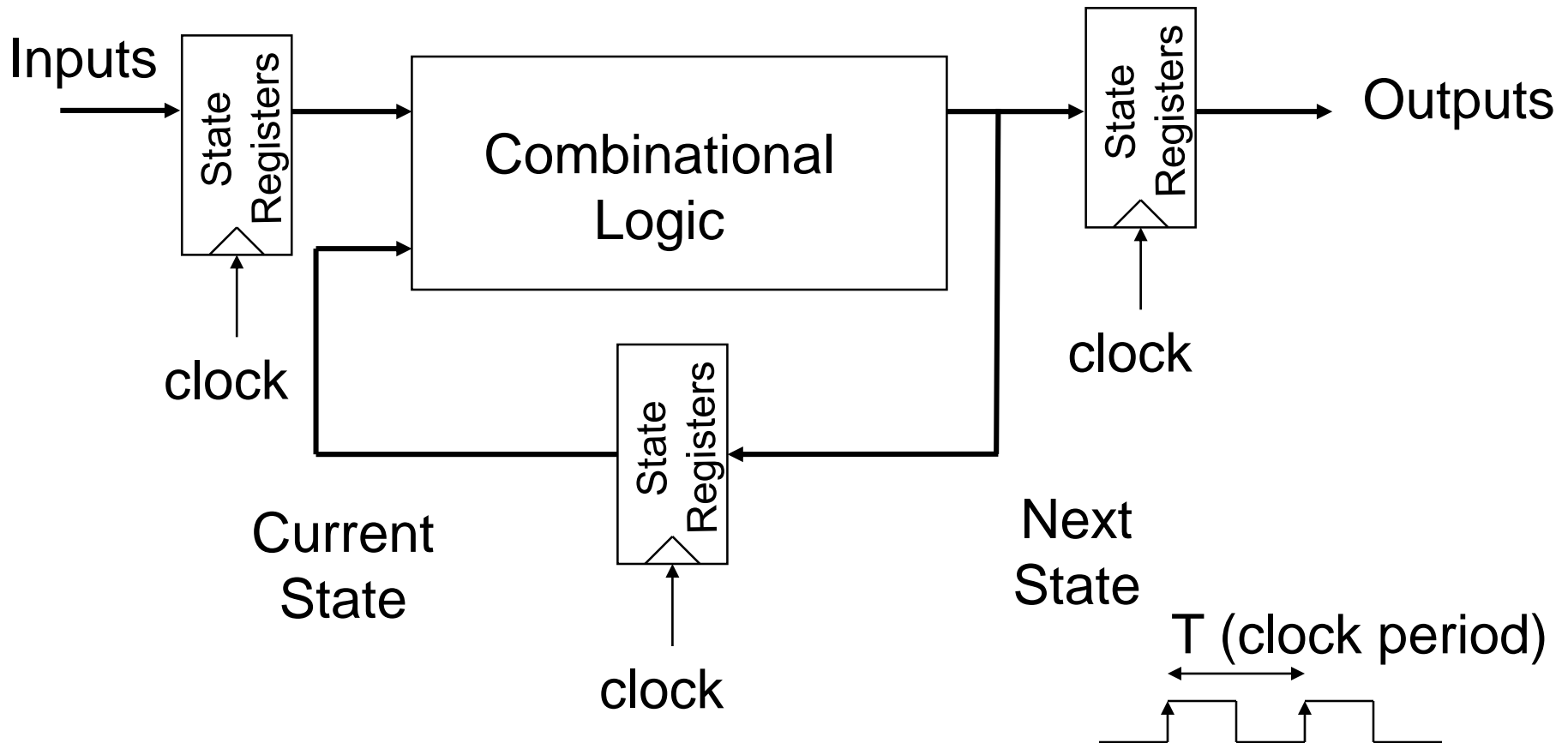
- ❑ Phase-Frequency Detector (*PFD*)
- ❑ Charge-Pump (*CP*)
- ❑ Low-Pass Filter (*LPF*)
- ❑ Voltage-Controlled Oscillator (*VCO*)
- ❑ VCO Level-Shifter (*LS*)
- ❑ Feedback Divider (*FBDIV*)

# Components in a Nutshell

---

- ❑ PFD: outputs digital pulse whose width is proportional to phase error
- ❑ CP: converts digital error pulse to analog error current
- ❑ LPF: integrates (and low-pass filters) error current to generate VCO control voltage
- ❑ VCO: low-swing oscillator with frequency proportional to control voltage
- ❑ LS: amplifies VCO levels to full-swing
- ❑ DIV: divides VCO clock to generate FBCLK clock

# System Timing Constraints



$$t_{\text{cdreg}} + t_{\text{cdlogic}} \geq t_{\text{hold}}$$

$$T \geq t_{\text{c-q}} + t_{\text{plogic}} + t_{\text{su}}$$