Course Outline

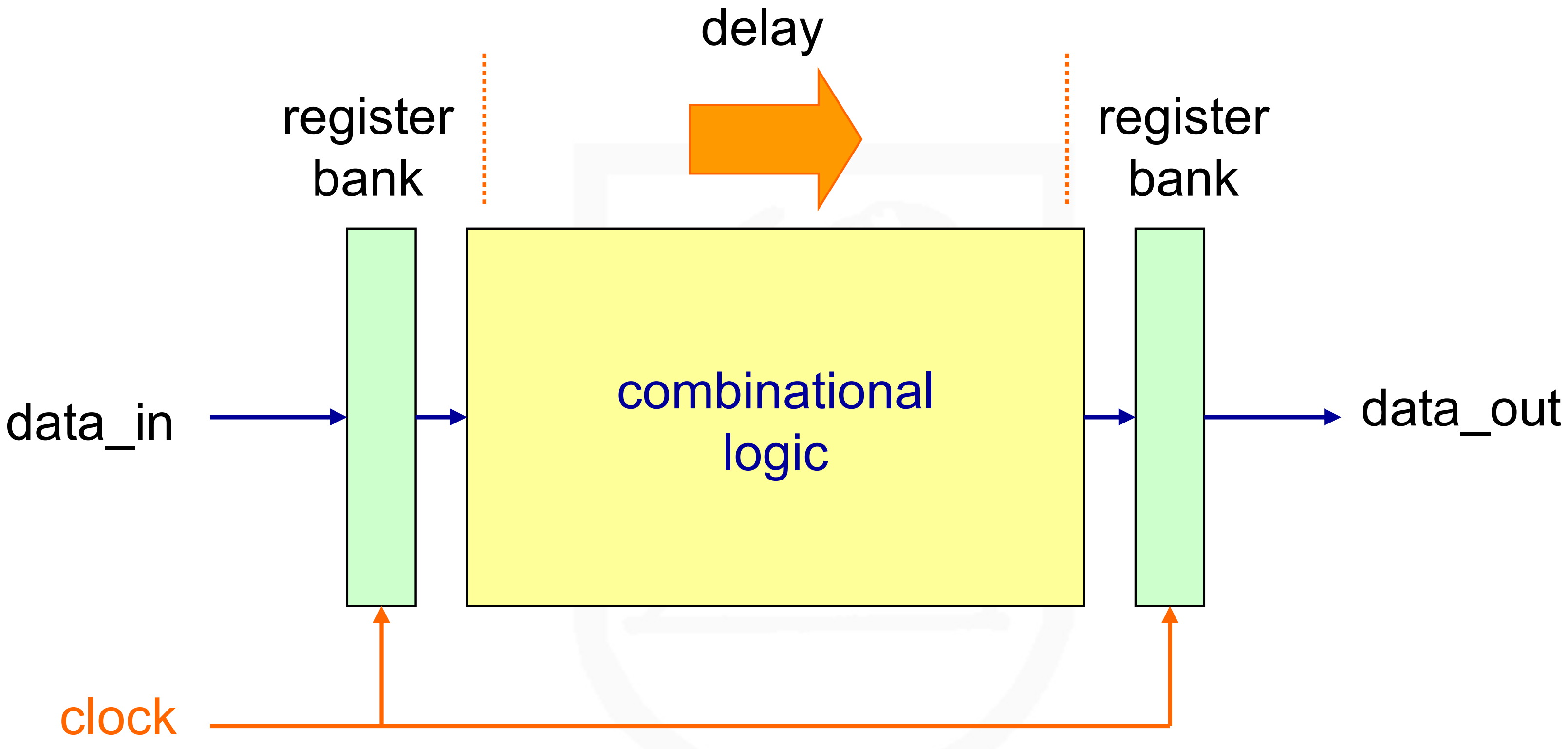
- Semiconductor Industry and Technology Overview
- IC Design Flows
- **Timing in Digital Systems**
- Front-end Design Flow
- Back-end Design Flow
- Interconnection and Signal Integrity
- Low-Power Design
- Design-for-Testability (DFT)

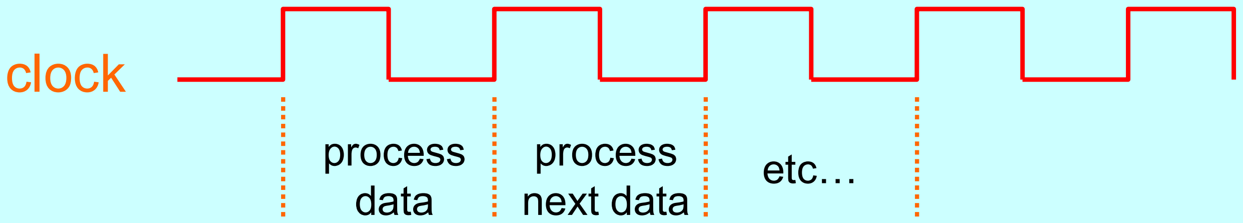
- Based on “Introduction to VLSI Design” slides by Professor Yusuf Leblebici of the Microelectronic Systems Laboratory (LSM) and others.

Synchronous vs. Asynchronous Systems

- **Synchronous Systems**
  - All memory elements are simultaneously updated using a global clock signal
  - Functionality is ensured by strict constraints on the clock generation and distribution
- **Asynchronous Systems**
  - Self-timed systems with no global clock but have asynchronous circuit overhead (e.g., handshaking logic)
- **GALS**
  - Hybrid systems that are global asynchronous with no common clock between blocks, while each block is locally synchronous.

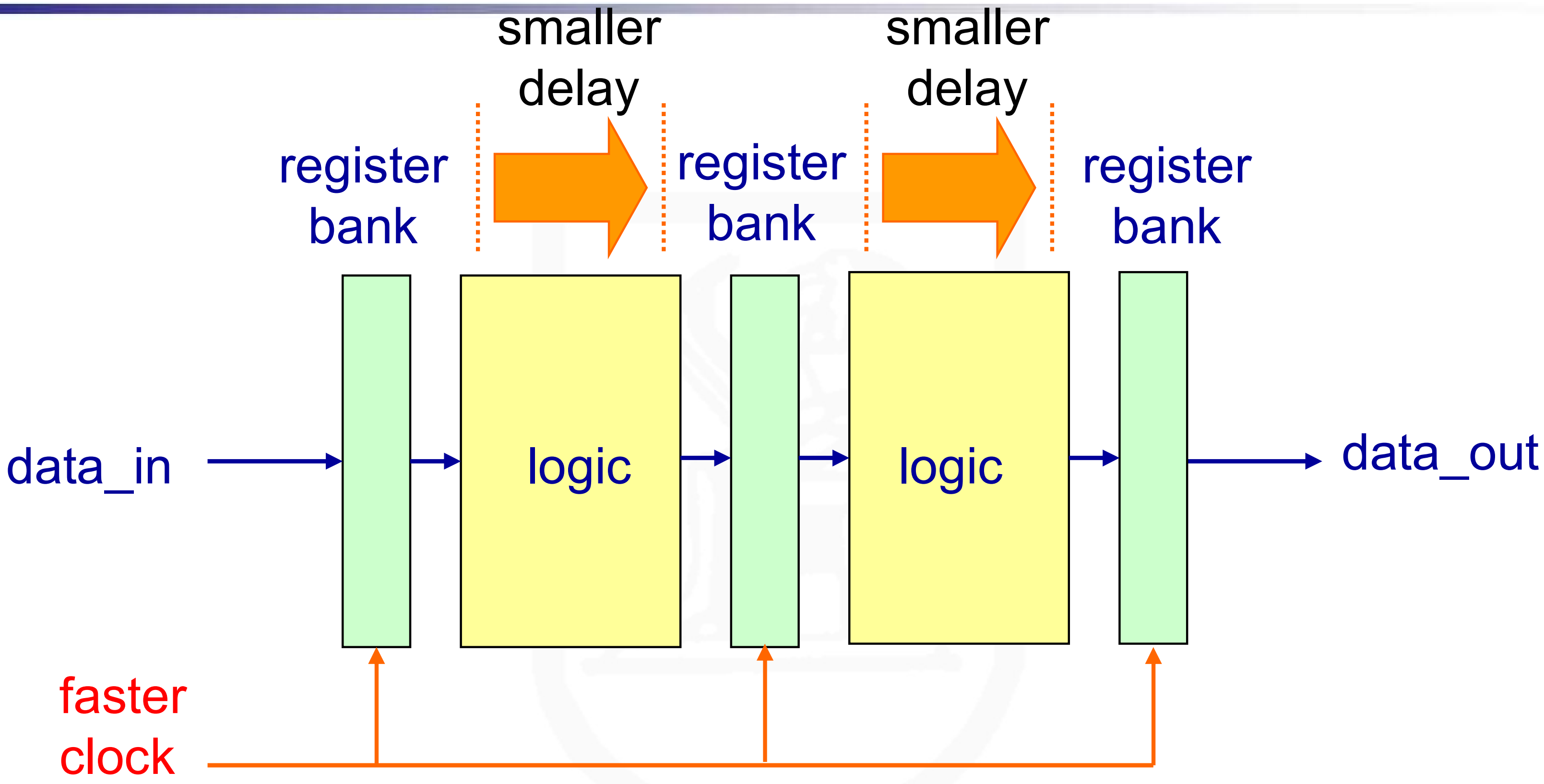
Synchronous Digital Logic





The clock period cannot be shorter than the delay of the combinational block !

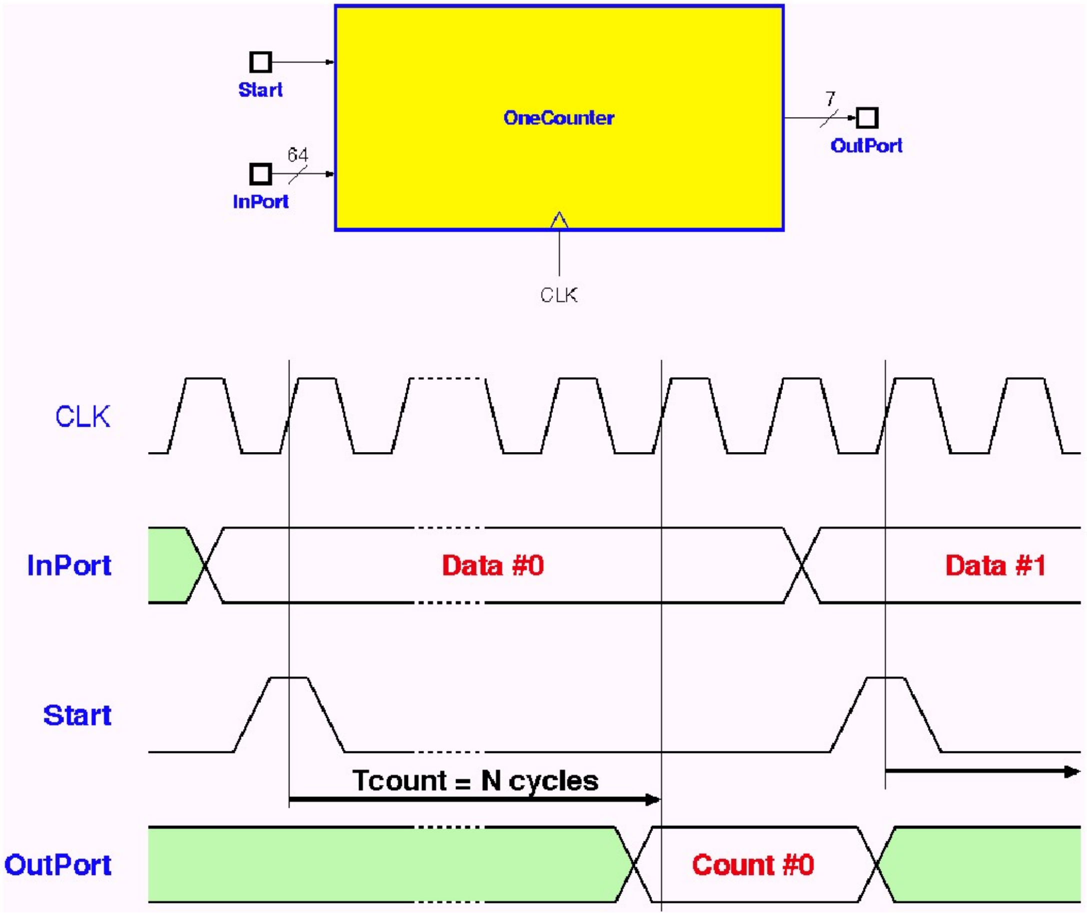
Pipelined Digital Logic



Pipelining results in smaller stage delays between register banks, and consequently, faster clock frequencies → System throughput increases !

However, latency also increases slightly and hardware overhead grows as a result of additional registers and clock distribution network.

Clocked Logic Example



Parallel Counter

- Example:

  if **InPort** is

  $1248c6391248c639_{16}$

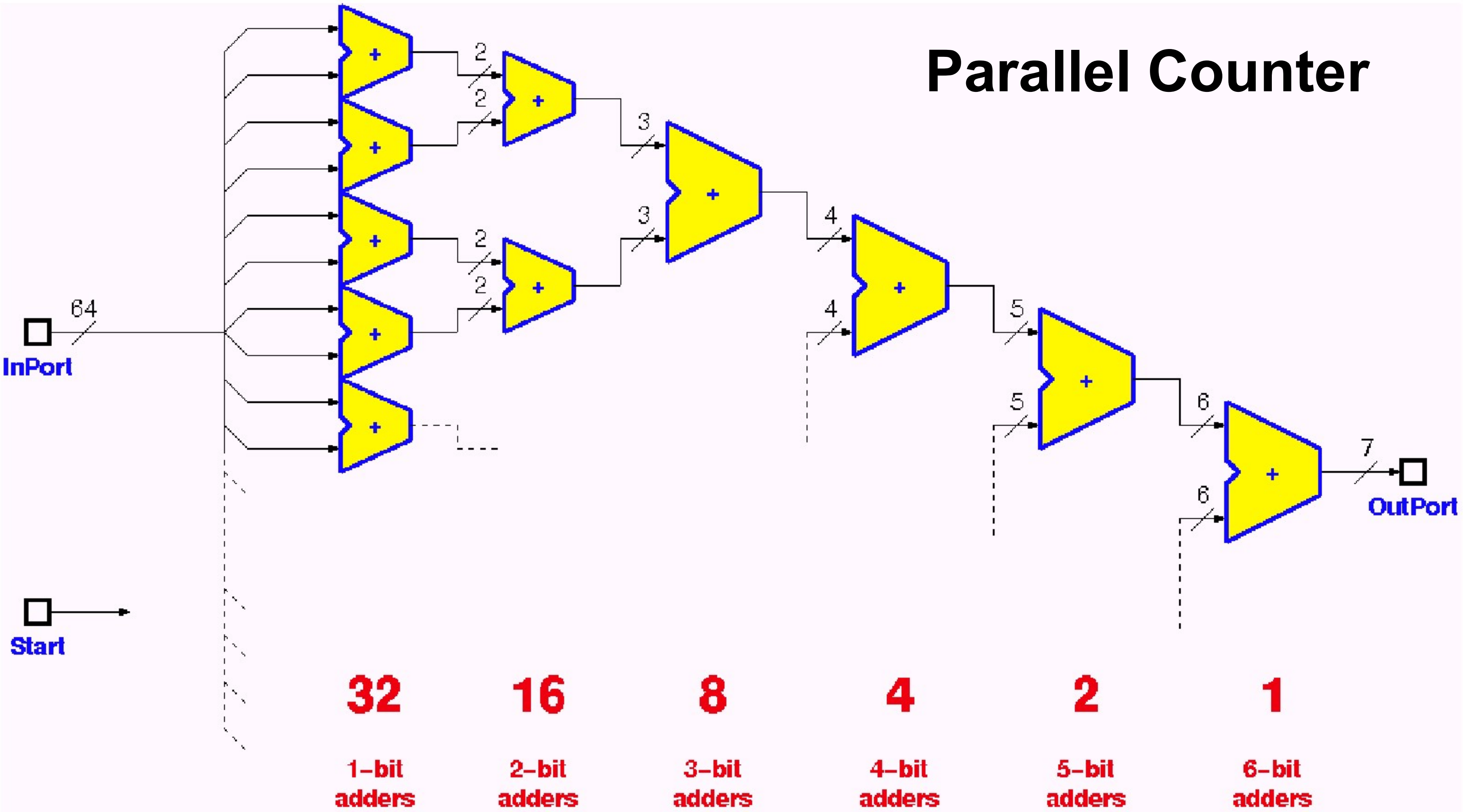
  then **OutPort** must be

  $0011000_2 = 24$

- Number of cycles (N) is unspecified

There is always more than one way of solving a problem !

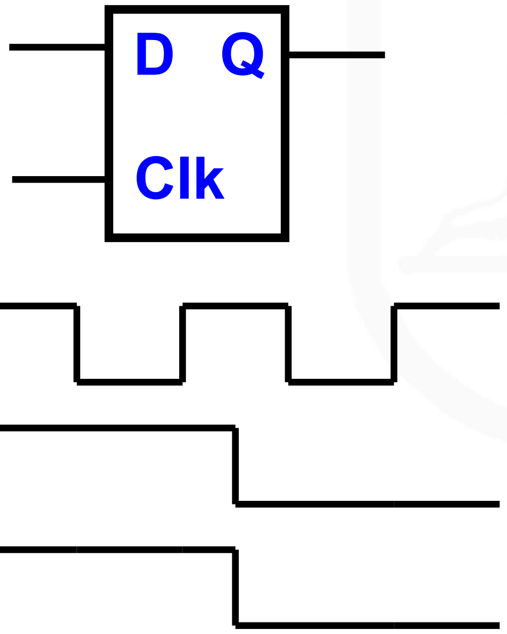
Clocked Logic Example



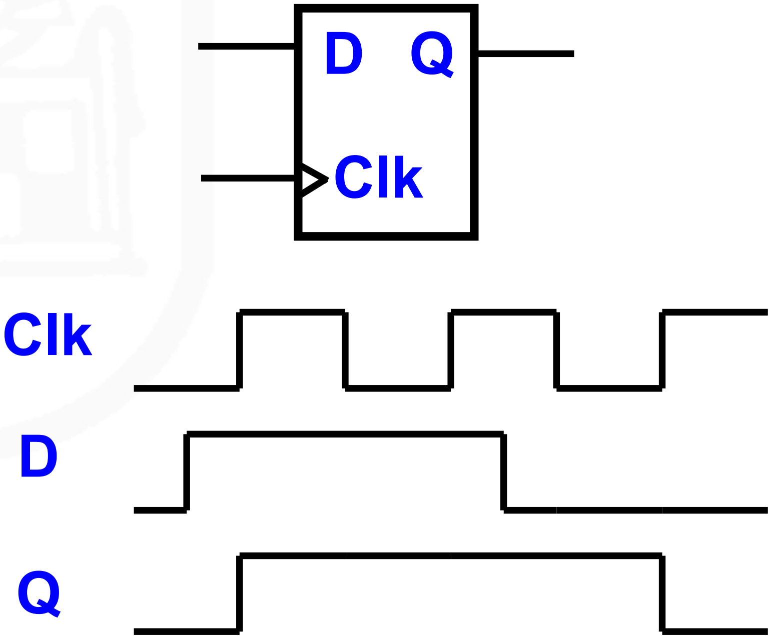
Much faster but also complicated hardware !

Latch versus Register

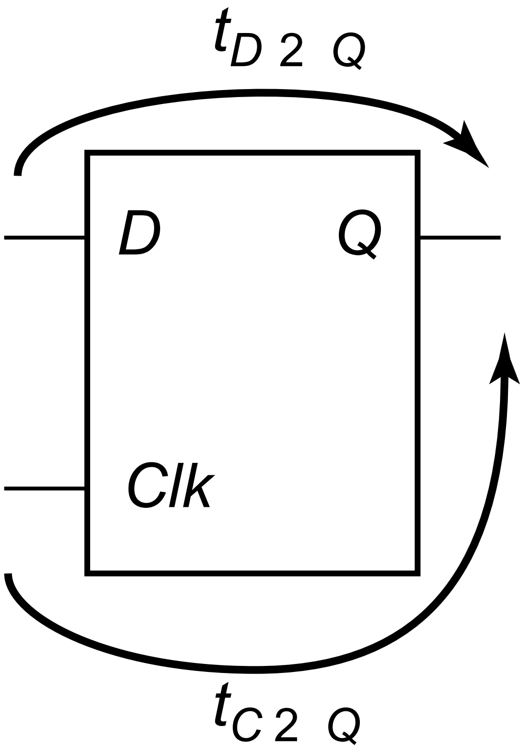
- **Latch**
  stores data when clock is low



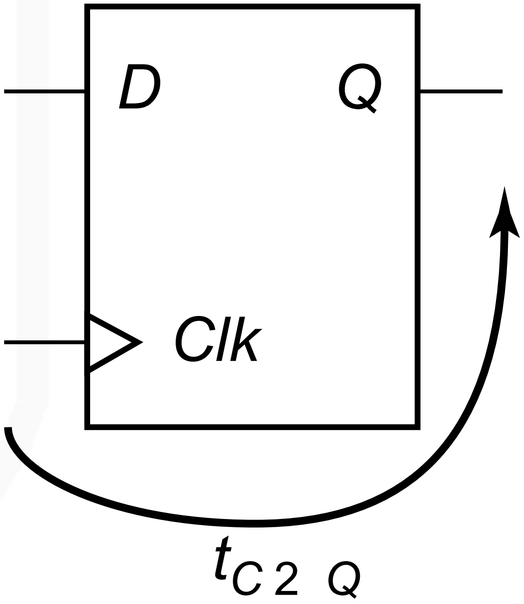
- **Register**
  stores data when clock rises



Characterizing Timing

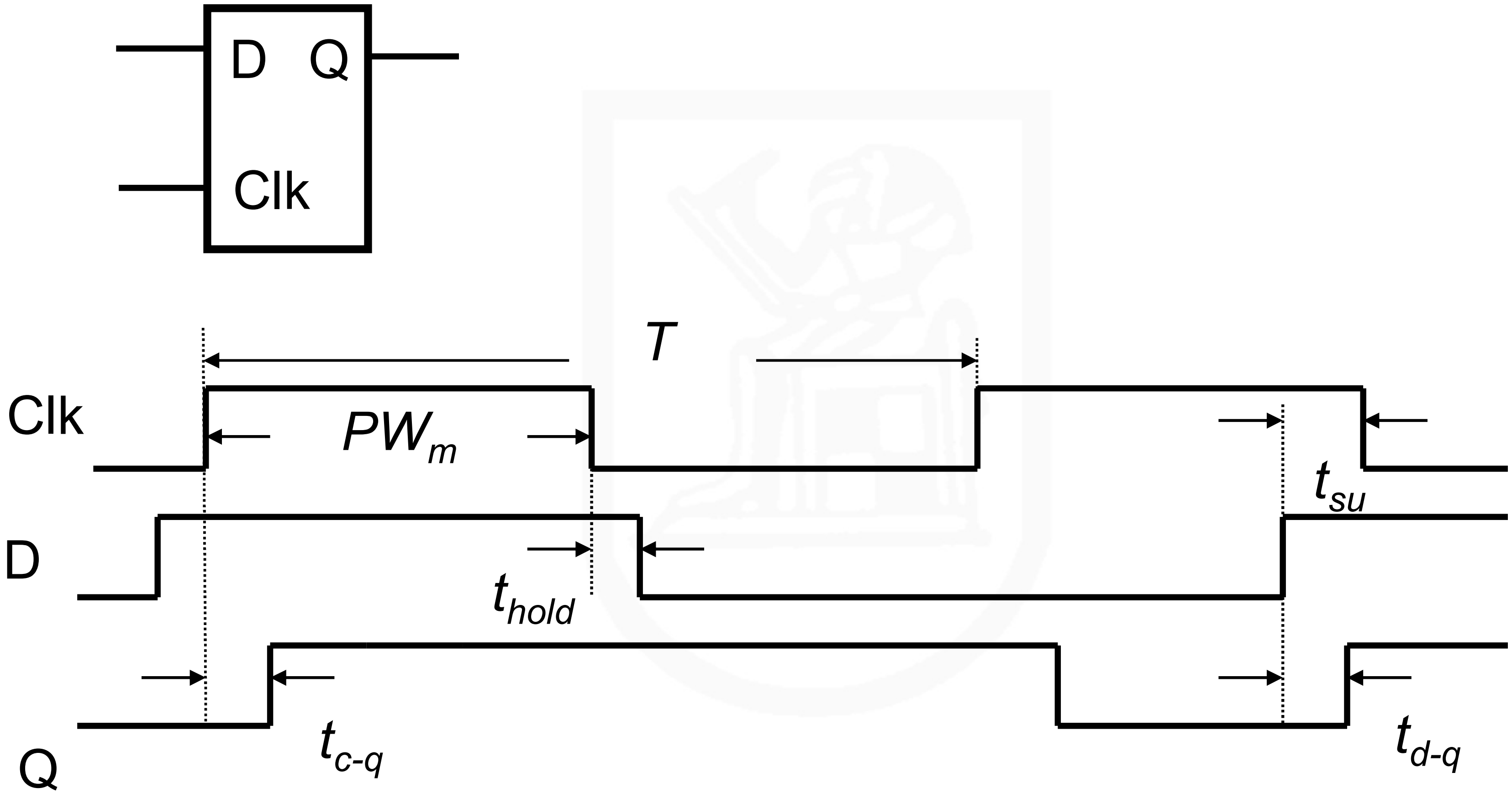


Latch



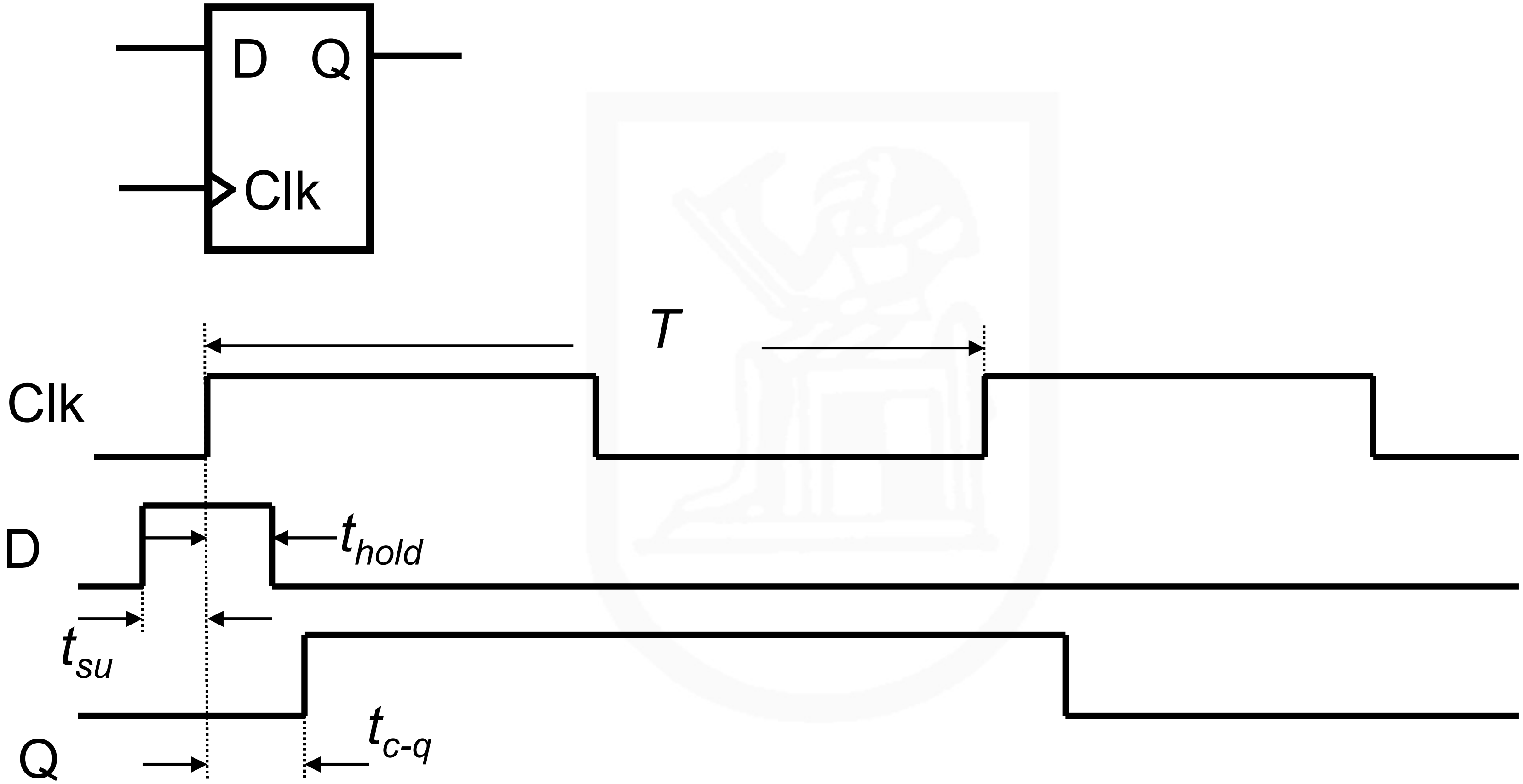
Register

Latch Parameters



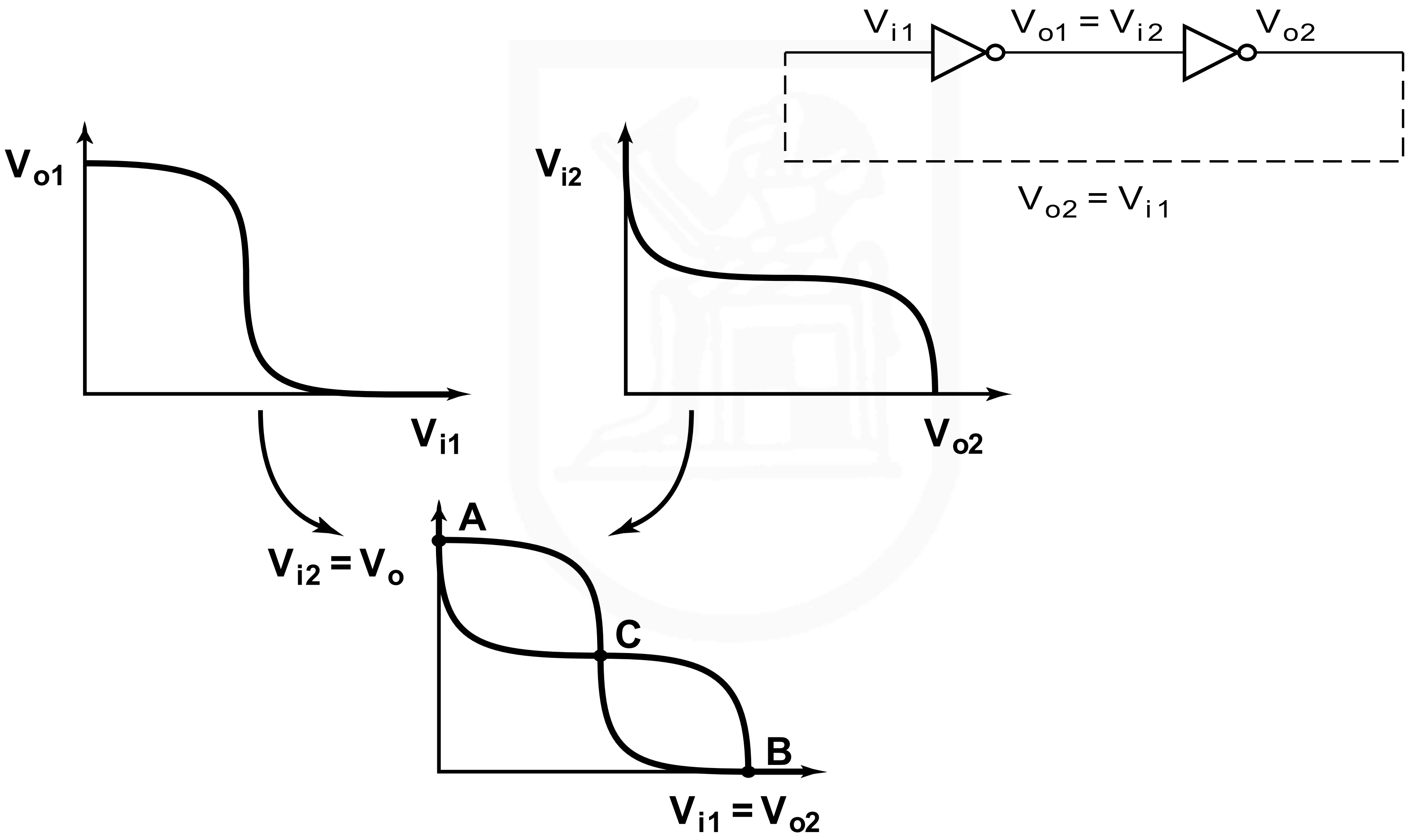
Delays can be different for rising and falling data transitions

Register Parameters

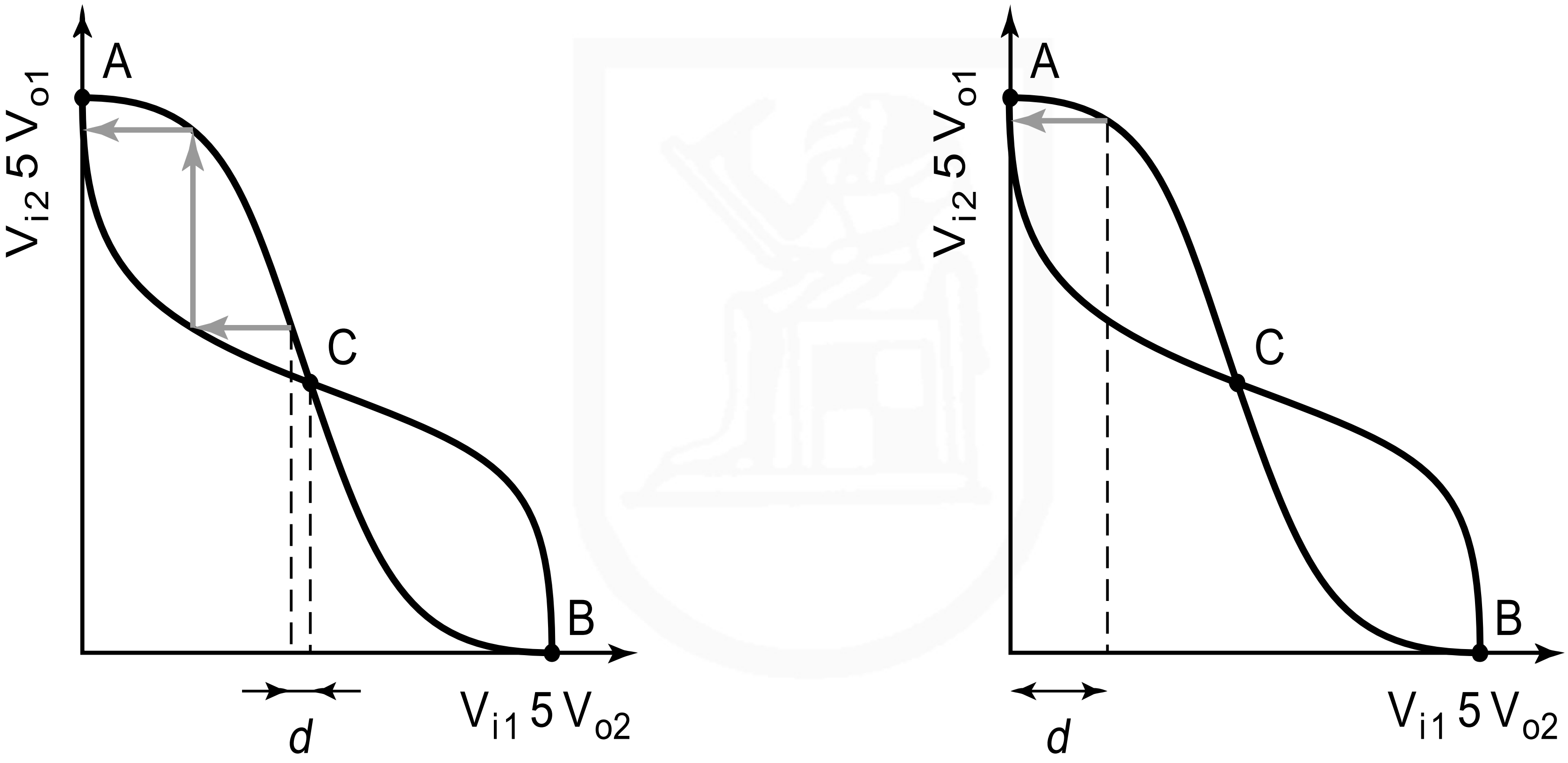


Delays can be different for rising and falling data transitions

Positive Feedback: Bi-Stability



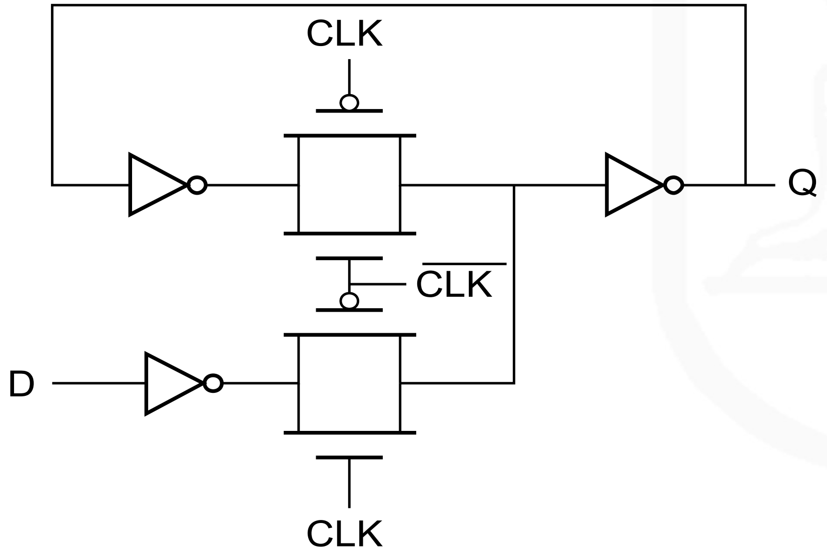
Meta-Stability

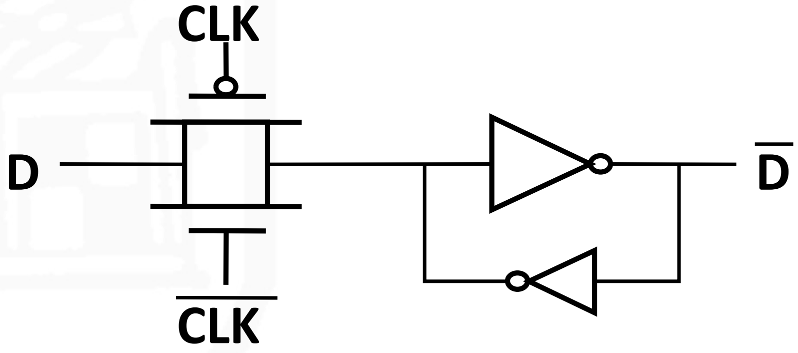


Gain should be larger than 1 in the transition region

Writing into a Static Latch

Use the clock as a decoupling signal, that distinguishes between the transparent and opaque states



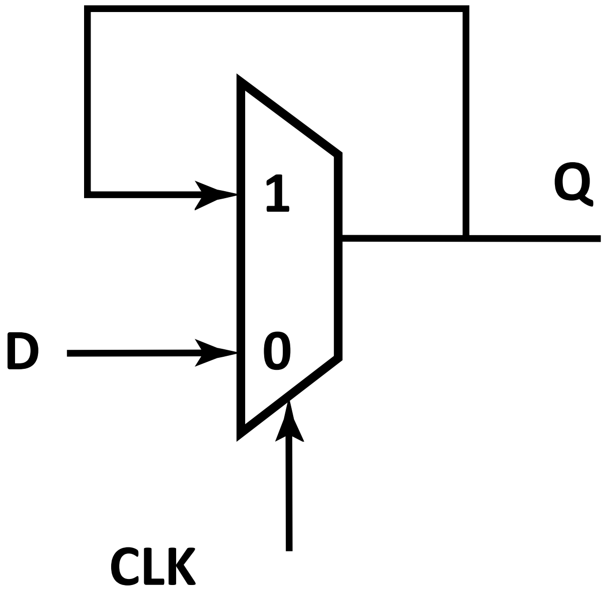


Converting into a MUX

Forcing the state (can implement as NMOS-only)

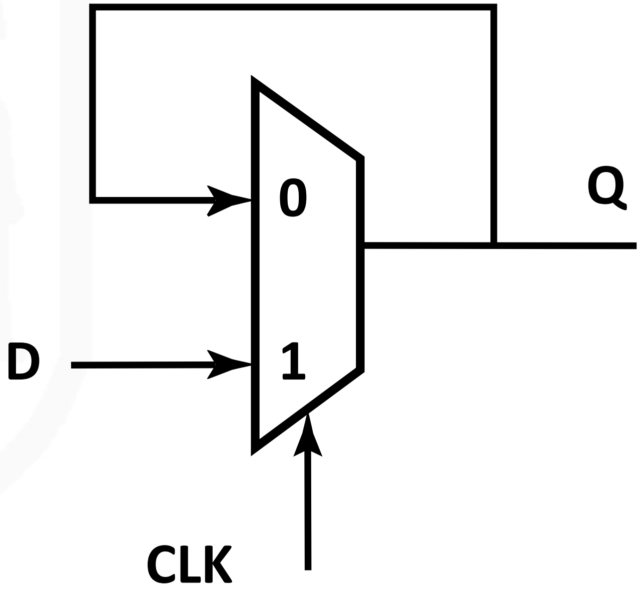
Mux-Based Latches

Negative latch
(transparent when CLK= 0)



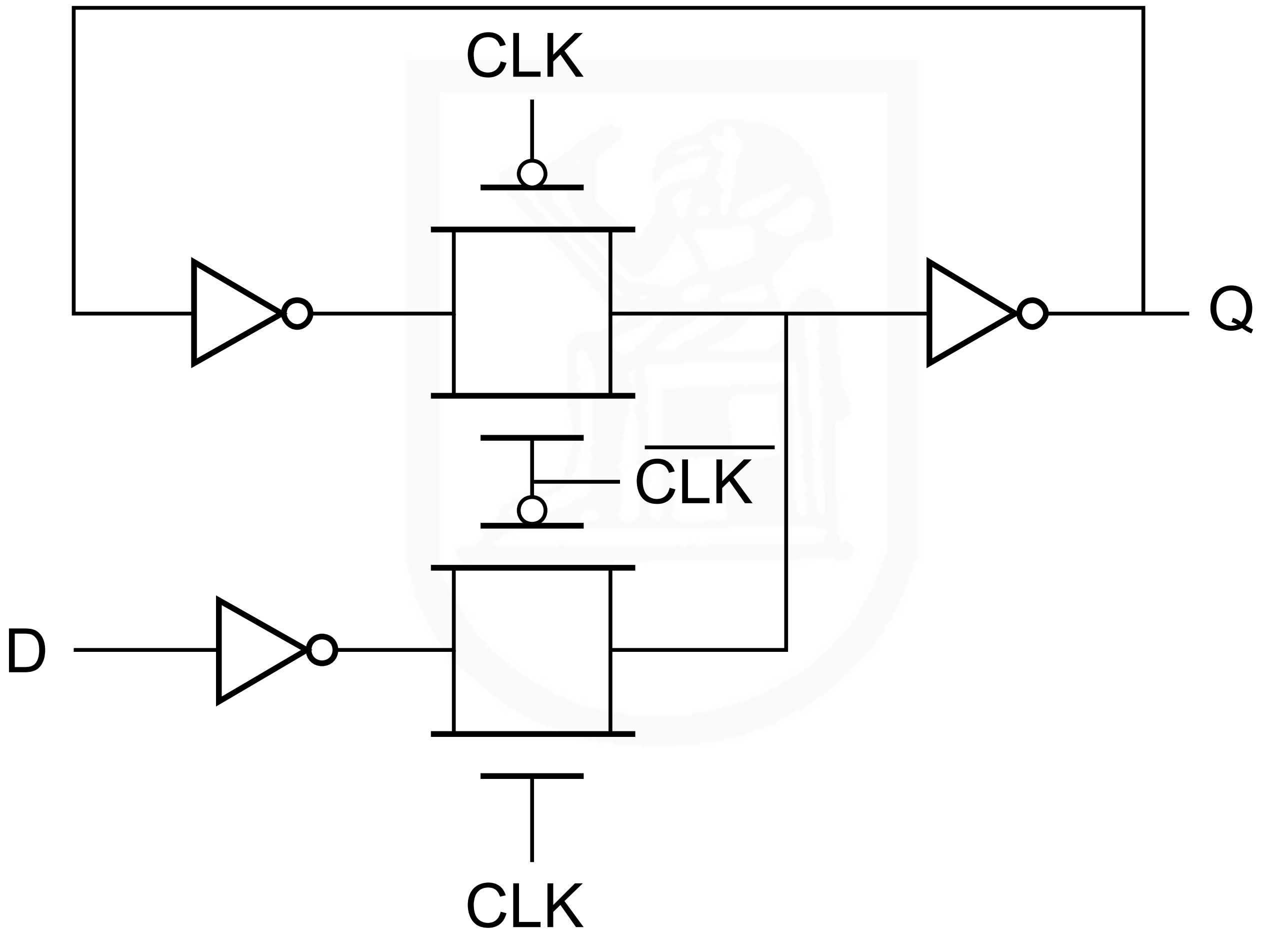
$$Q = \overline{Clk} \cdot Q + Clk \cdot In$$

Positive latch
(transparent when CLK= 1)

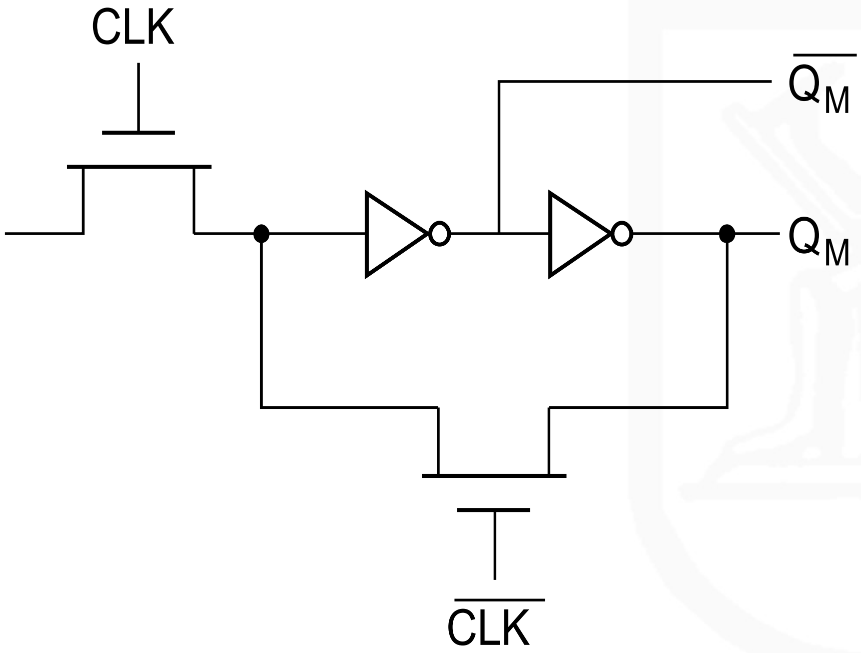


$$Q = Clk \cdot Q + \overline{Clk} \cdot In$$

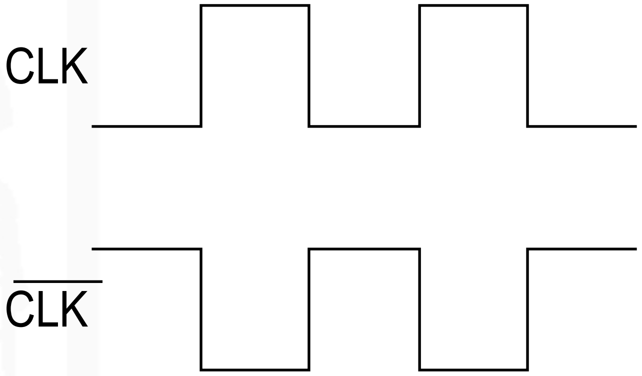
Mux-Based Latch



Mux-Based Latch

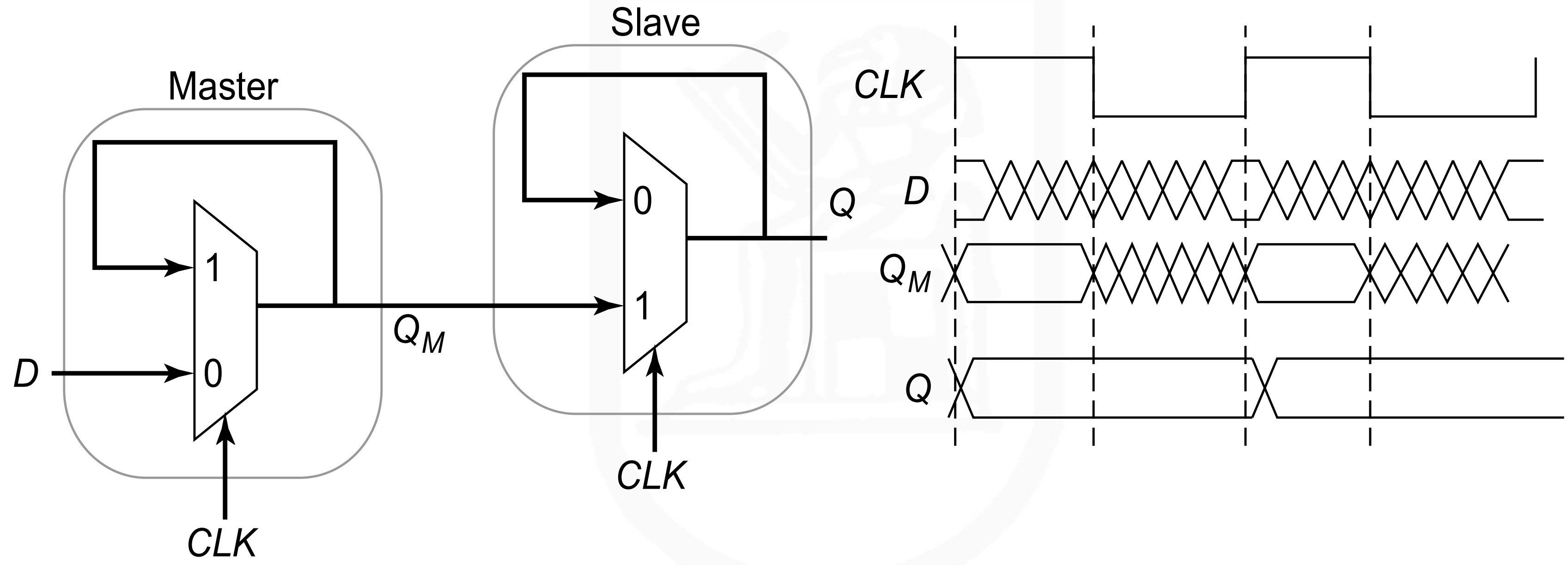


NMOS only



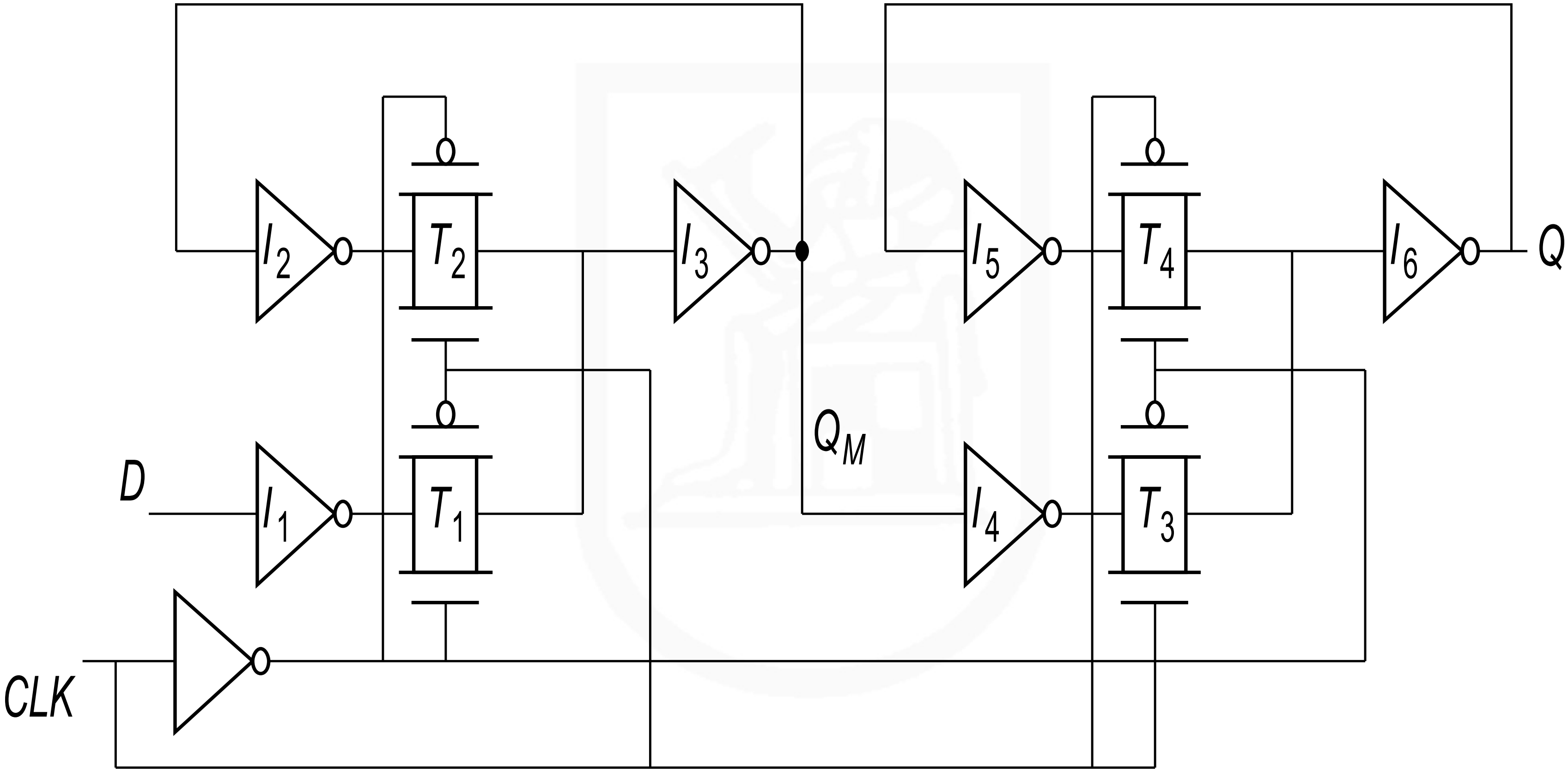
Non-overlapping clocks

Master-Slave (Edge-Triggered) Register



Two opposite latches trigger on edge also called as master-slave latch pair

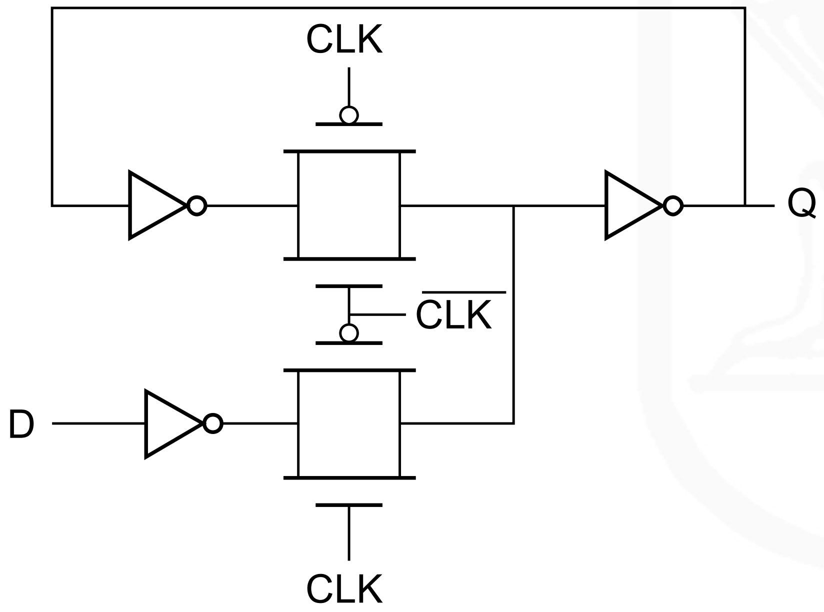
Master-Slave Register



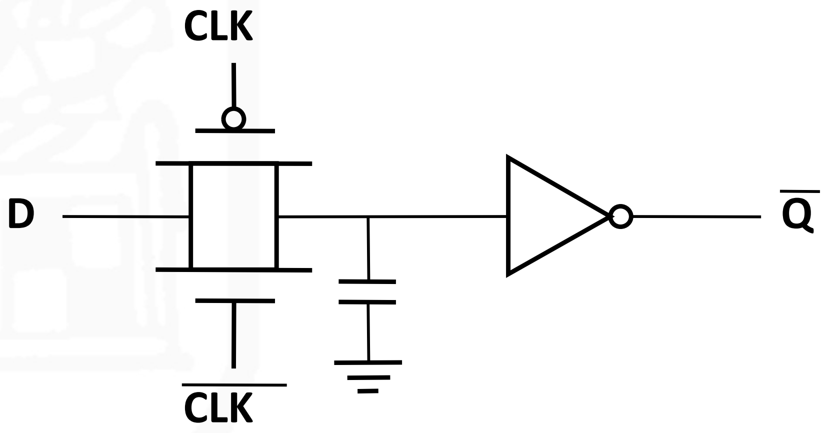
Multiplexer-based latch pair

Storage Mechanisms

Static

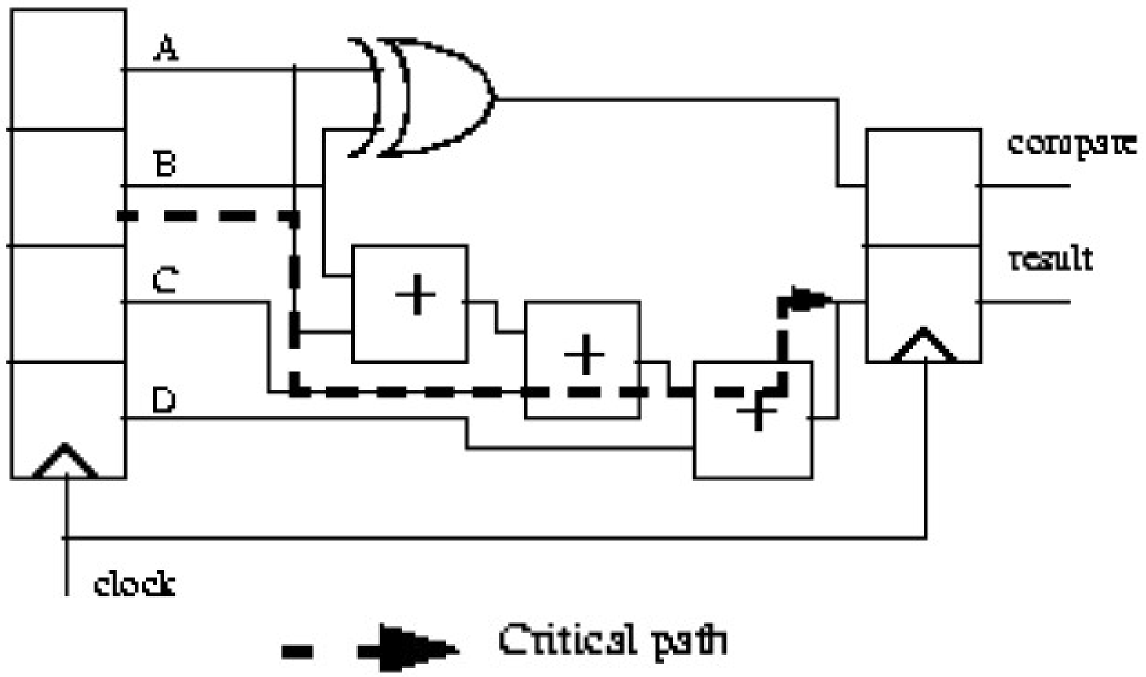


Dynamic (charge-based)



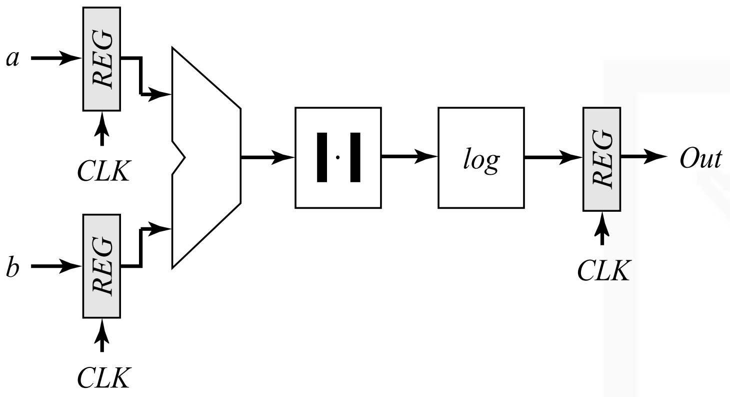
Critical Path

The clock speed is determined by the slowest feasible path between registers in the design, often referred to as *the critical path*.

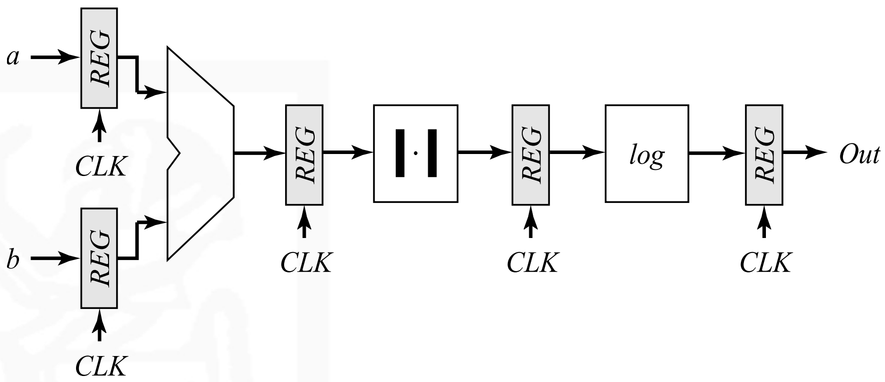


Critical path is longer with increased *logic depth* (# gates in series)

Pipelining



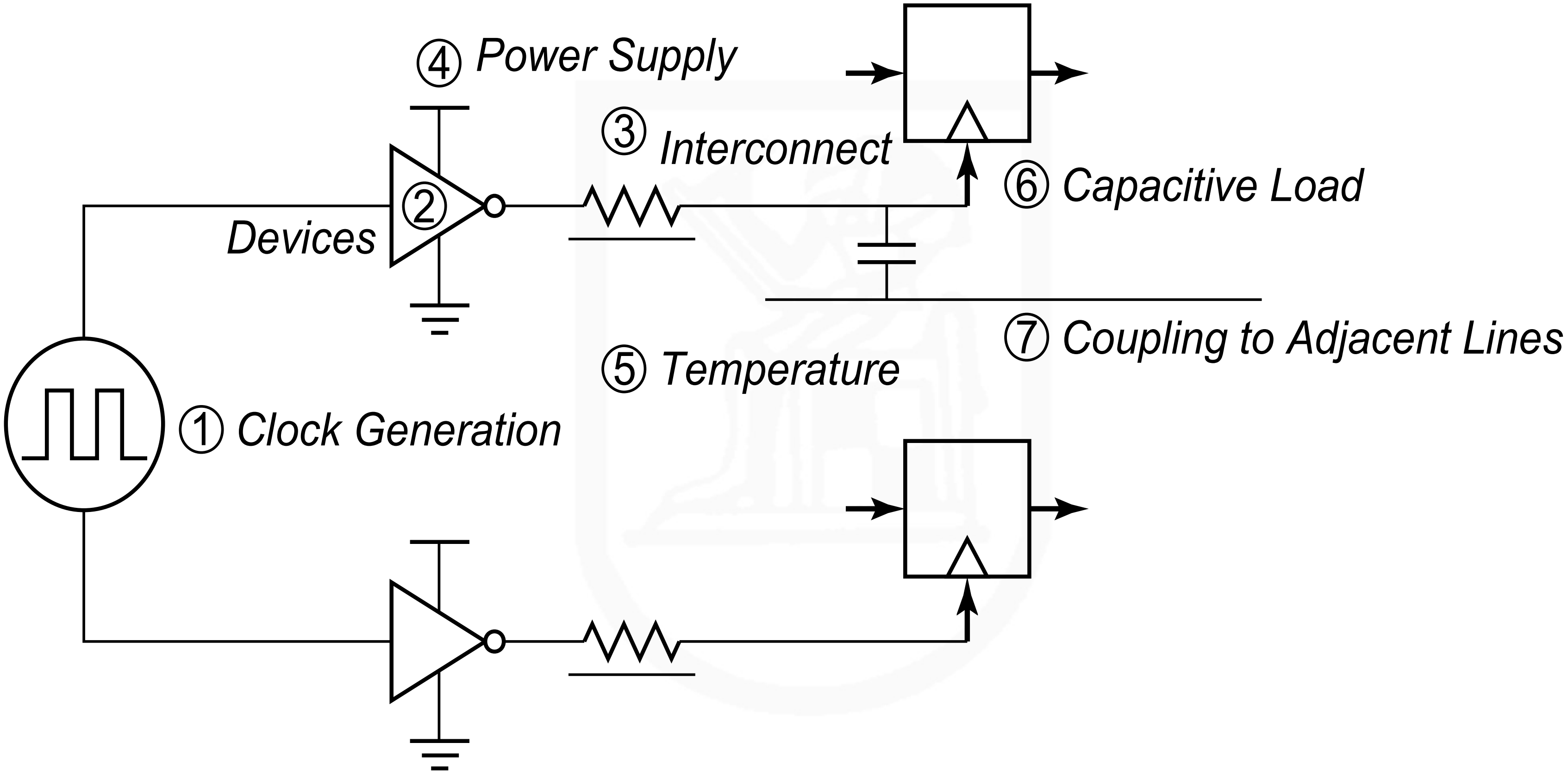
Reference



Pipelined

Clock Period	Adder	Absolute Value	Logarithm
1	$a_1 + b_1$		
2	$a_2 + b_2$	$\lvert a_1 + b_1\rvert$	
3	$a_3 + b_3$	$\lvert a_2 + b_2\rvert$	$\log(\lvert a_1 + b_1\rvert)$
4	$a_4 + b_4$	$\lvert a_3 + b_3\rvert$	$\log(\lvert a_2 + b_2\rvert)$
5	$a_5 + b_5$	$\lvert a_4 + b_4\rvert$	$\log(\lvert a_3 + b_3\rvert)$

Clock Uncertainties



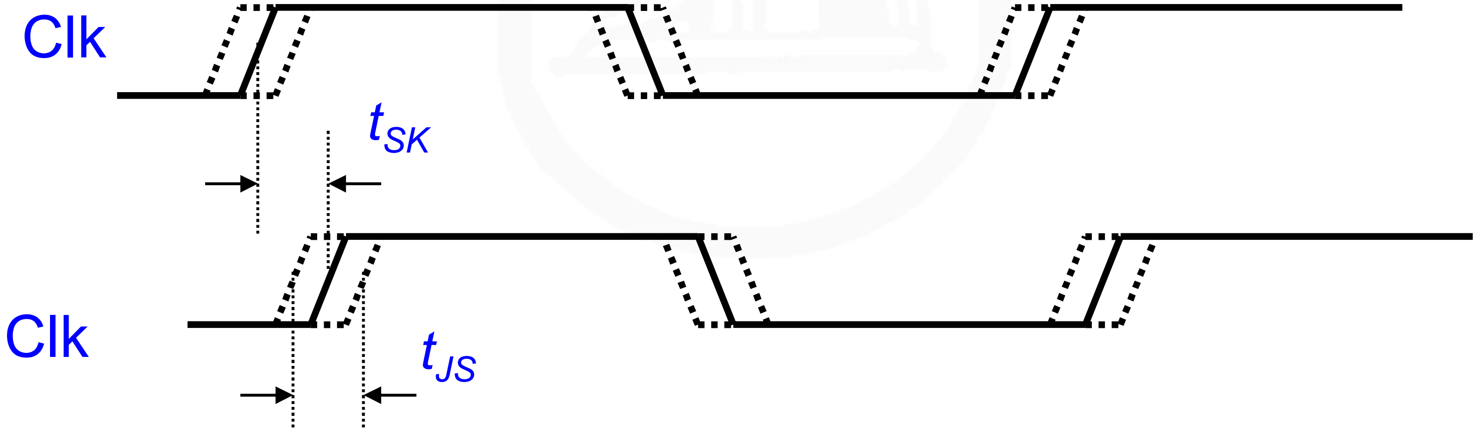
Sources of clock uncertainty

Clock Nonidealities

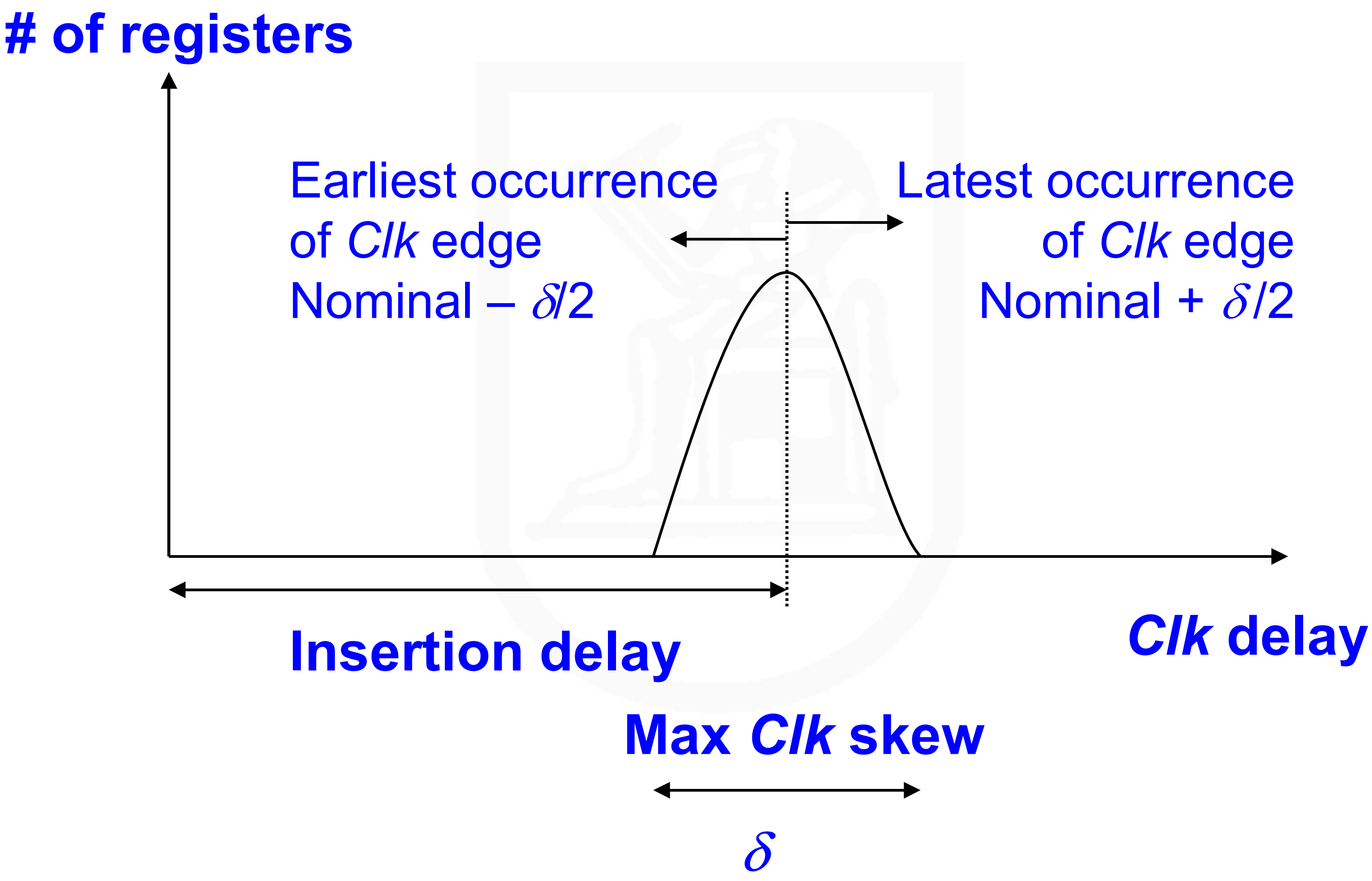
- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}
  - Skew is constant from cycle to cycle by definition.
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- **Variation of the pulse width**
  - Important for level sensitive and mixed-edge clocking

Clock Skew and Jitter

- Both skew and jitter affect the effective cycle time
- Only skew affects the race margin



Clock Skew



# of registers

Earliest occurrence of *Clk* edge
Nominal – $\delta/2$

Latest occurrence of *Clk* edge
Nominal + $\delta/2$

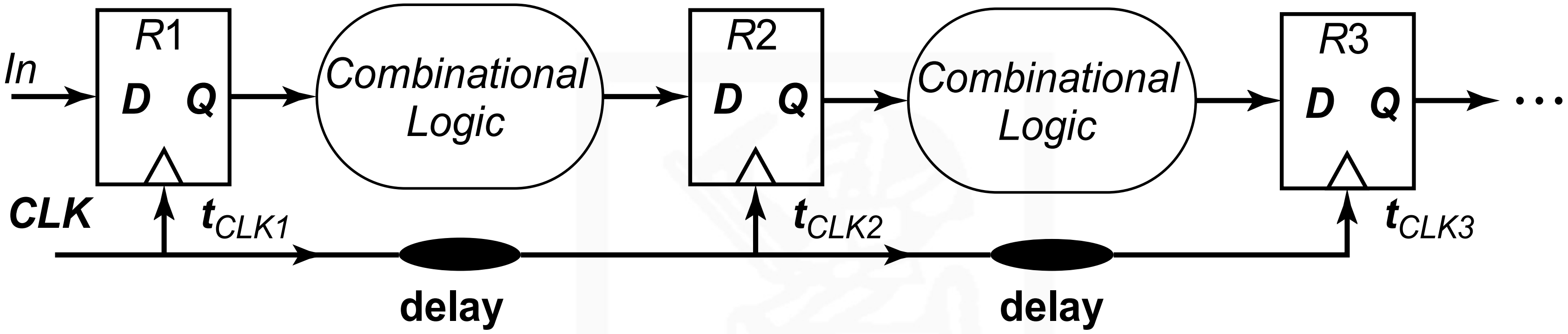
Insertion delay

***Clk* delay**

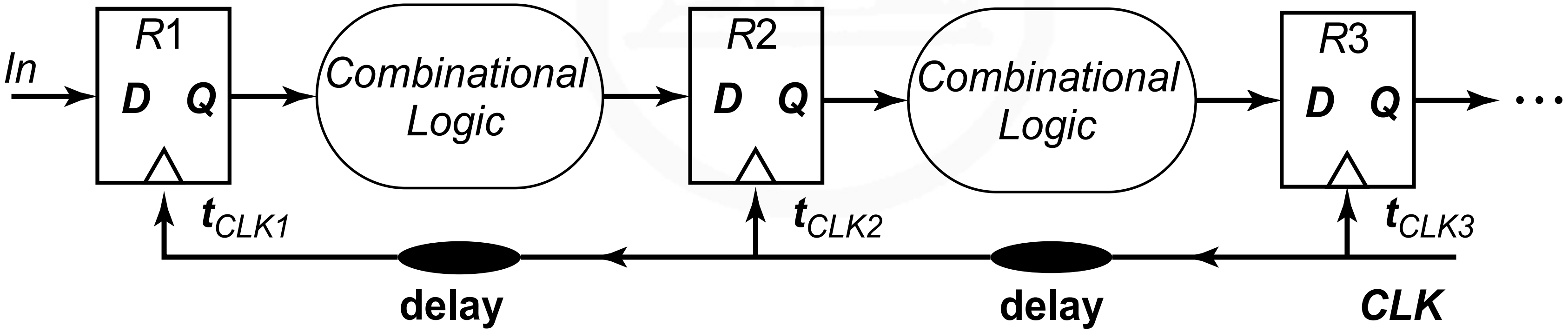
Max *Clk* skew

δ

Positive and Negative Skew

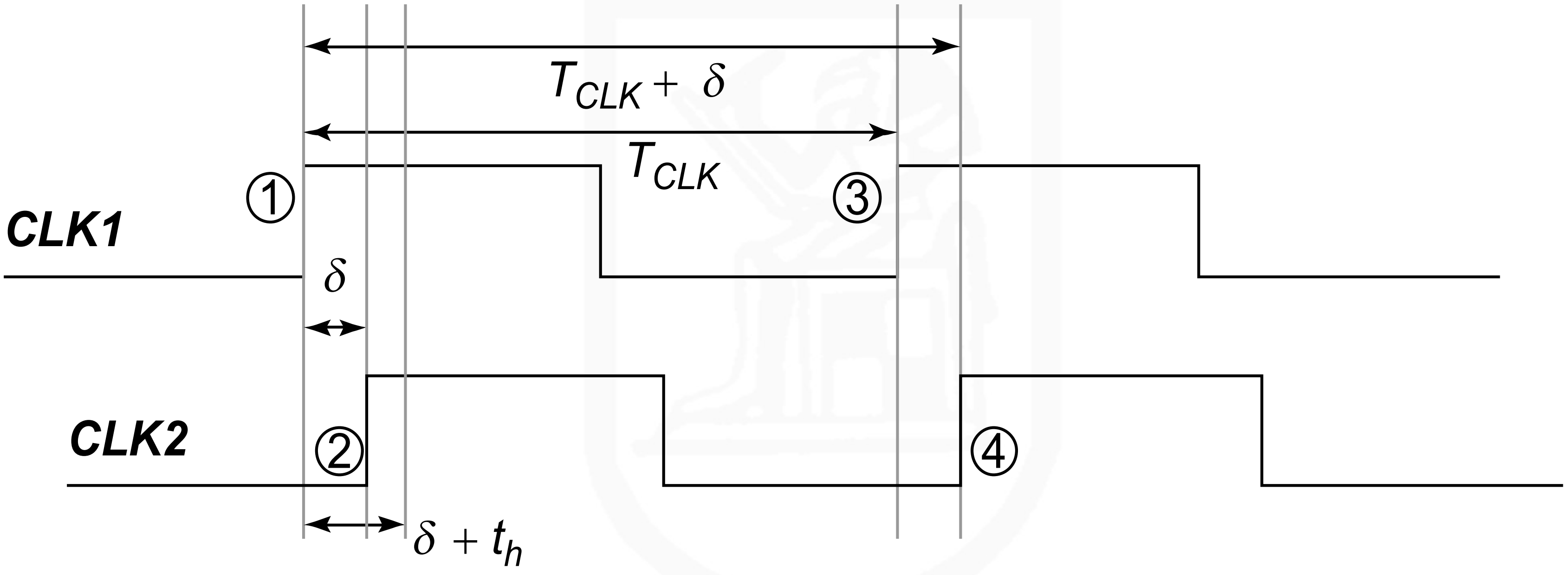


(a) Positive skew



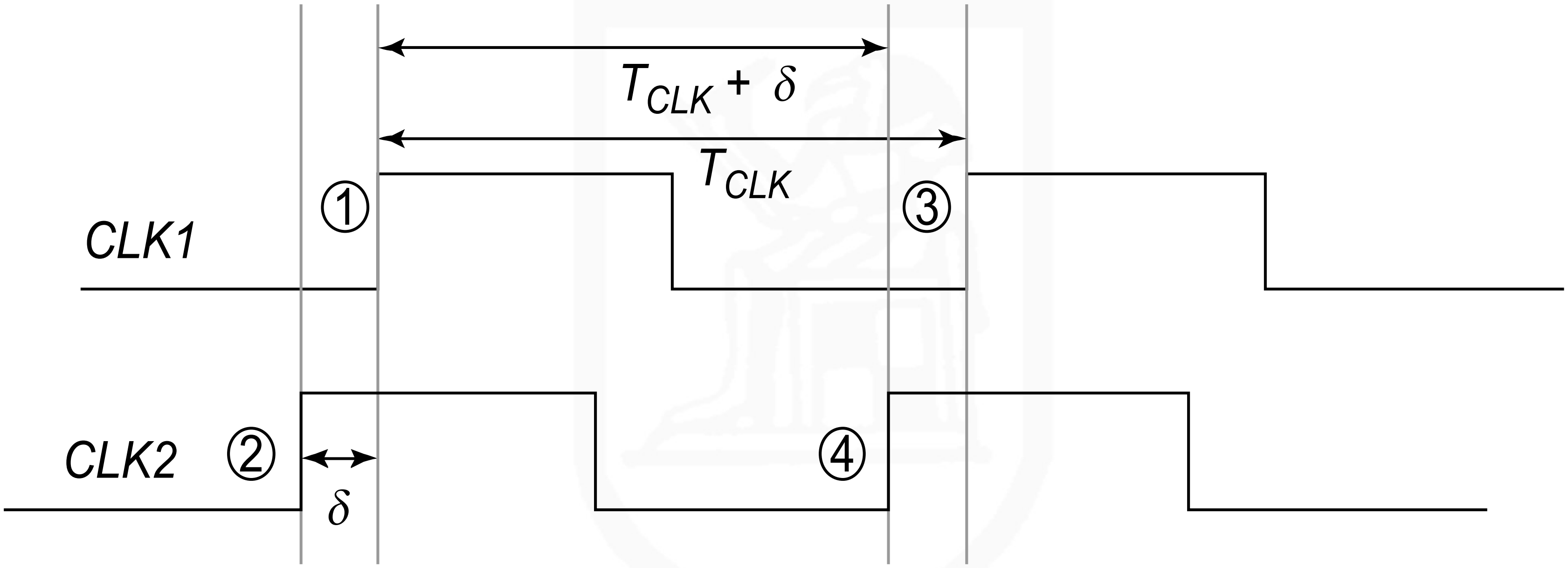
(b) Negative skew

Positive Skew



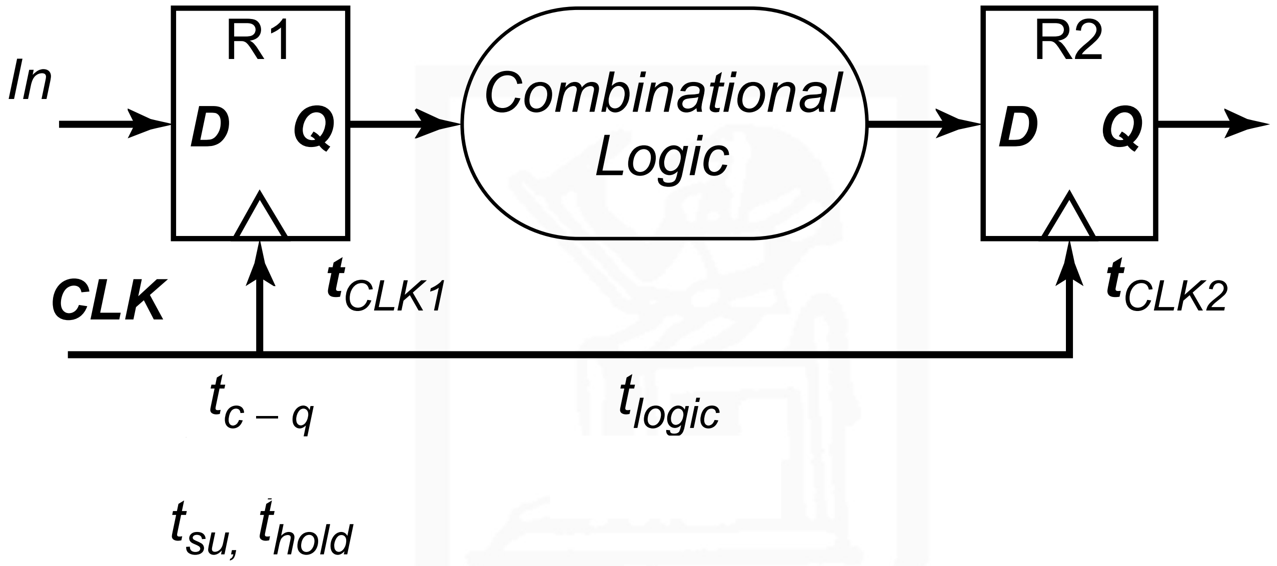
Launching edge arrives before the receiving edge

Negative Skew



Receiving edge arrives before the launching edge

Timing Constraints



Minimum cycle time:

$$T_{CLK} + \delta > t_{c\text{-}q\text{-}max} + t_{logic\text{-}max} + t_{su}$$

Worst case is when receiving edge arrives early (negative δ)

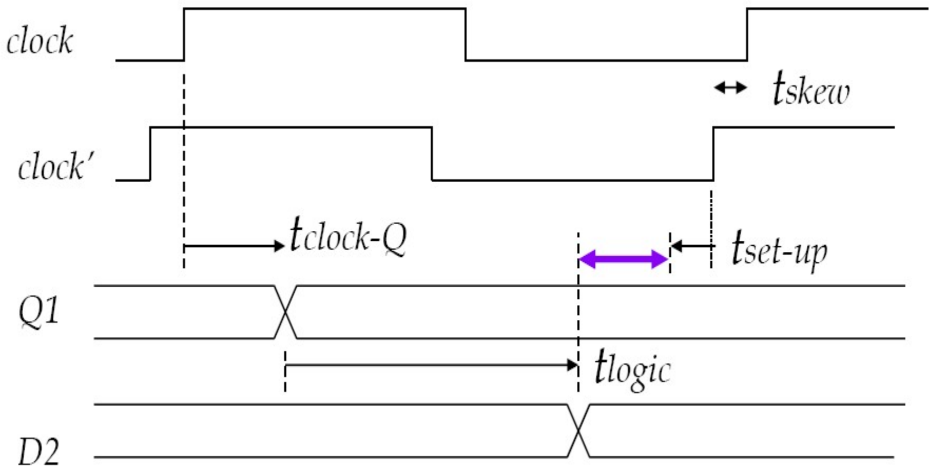
Preventing Set-Up Violations

- **Set-up violation:**

  Logic is too slow for the correct logic value to arrive at the inputs to the register on the right before one set-up time before the clock edge

- **Constraint to prevent this:**

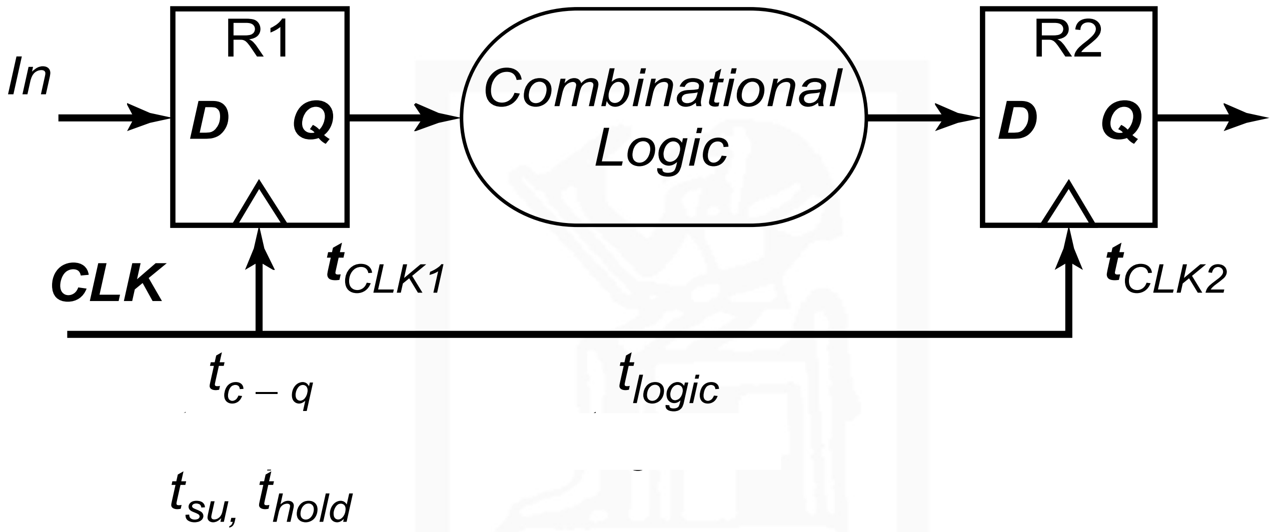
$$t_{clock} \geq t_{clock-Q-\max} + t_{\log ic-\max} + t_{set-up} - t_{skew}$$



The amount of time required to turn ‘>‘ into ‘=‘ is referred to as timing slack

⟷

Timing Constraints

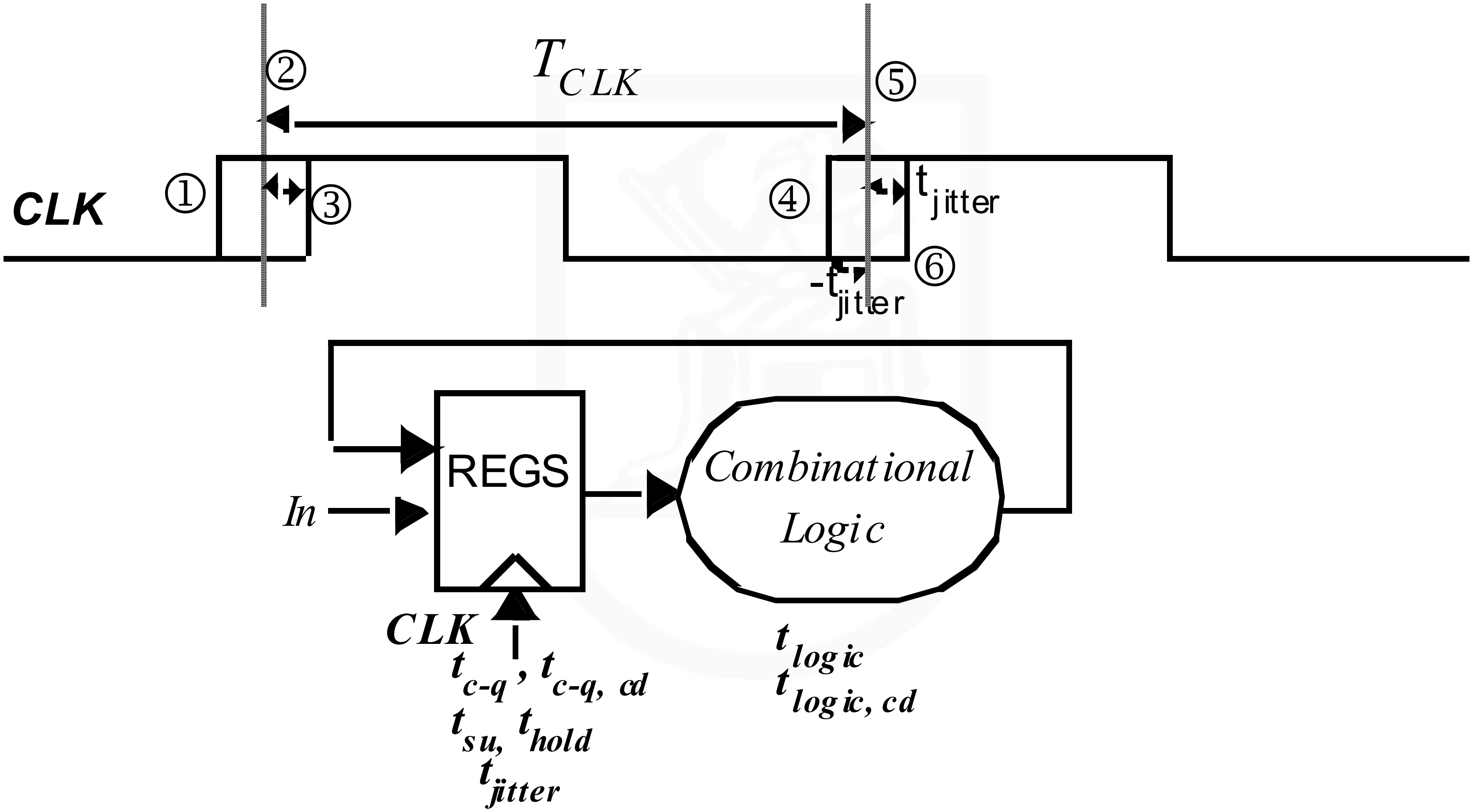


Hold time constraint:

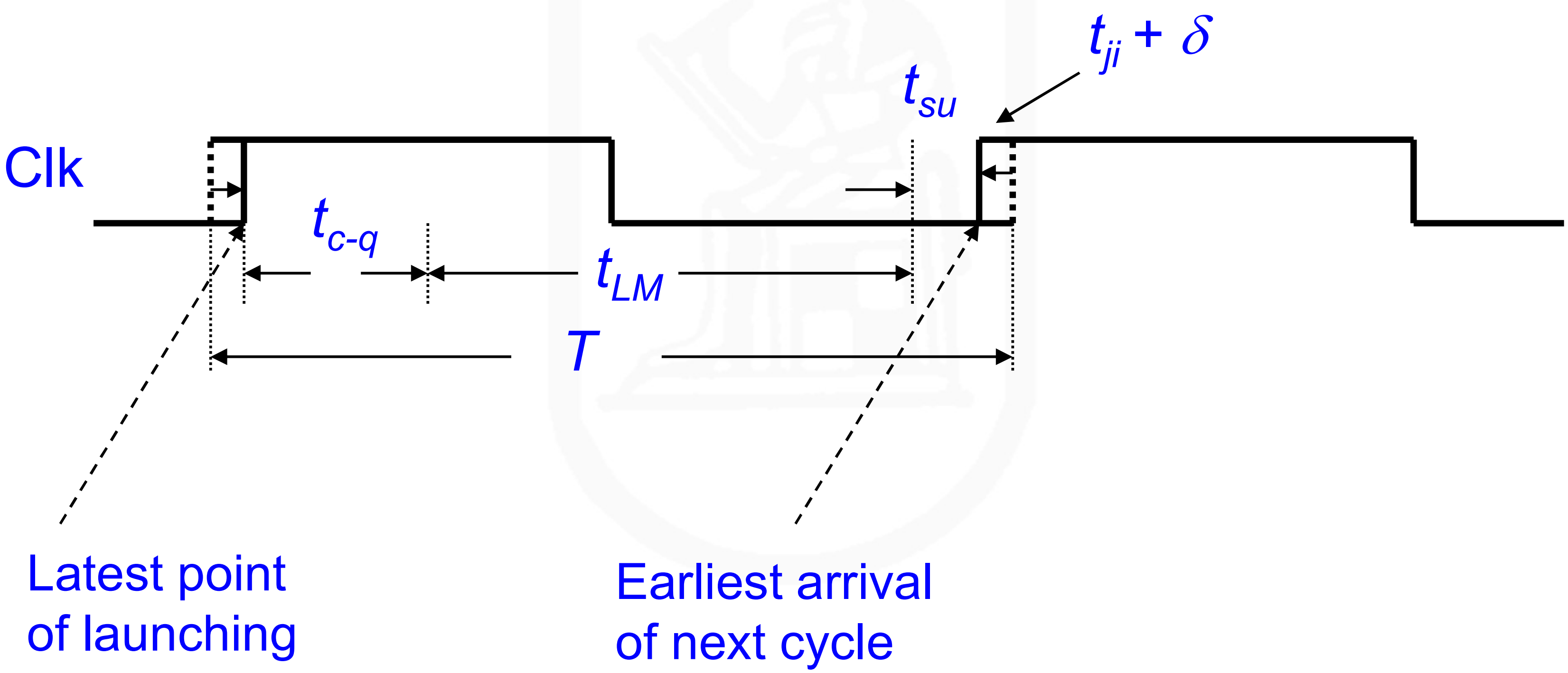
$$t_{c\text{-}q\text{-}min} + t_{logic\text{-}min} > t_{hold} + \delta$$

Worst case is when receiving edge arrives late
Race between data and clock

Impact of Jitter



Impact of Jitter: Longest Logic Path



Clock Constraints in Edge-Triggered Systems

If launching edge is late and receiving edge is early, the data will not be too late if:

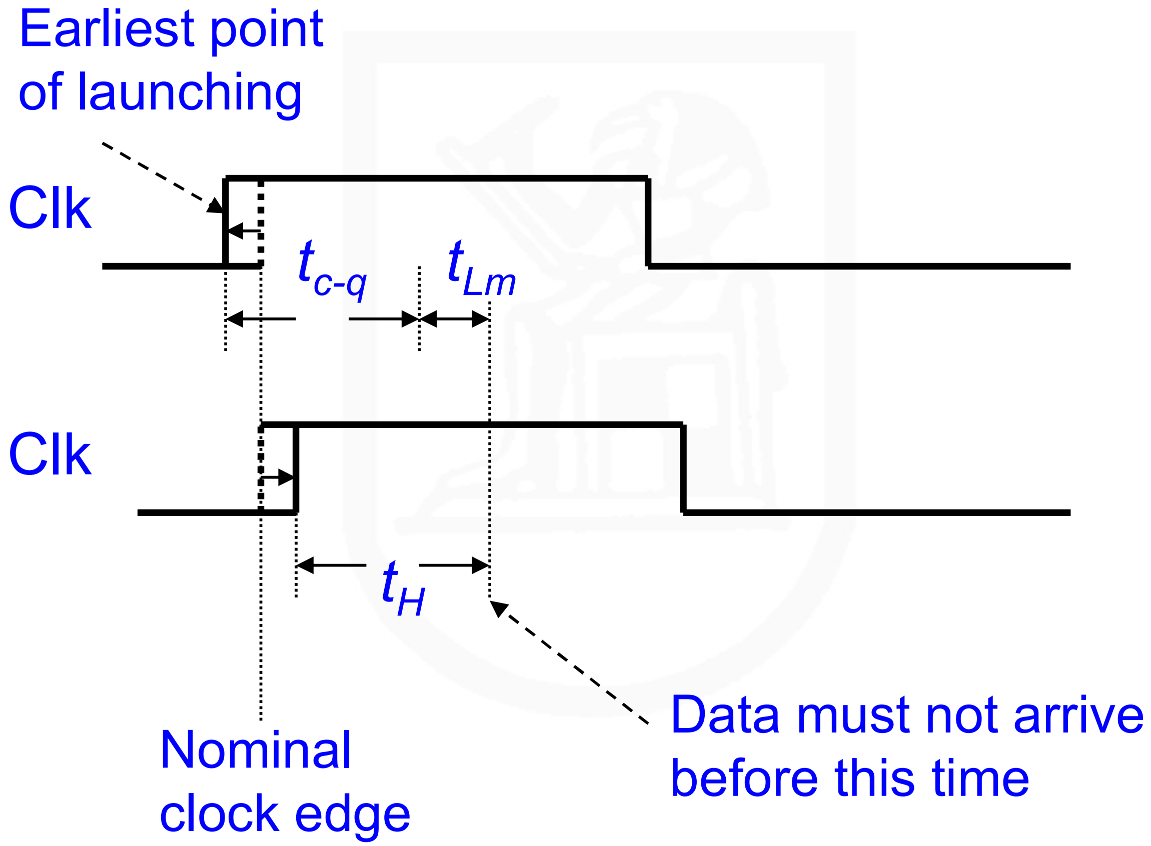
$$T_{CLK} + \delta > t_{c\text{-}q} + t_{LM} + t_{su} + t_{ji,1} + t_{ji,2}$$

Minimum cycle time is determined by the maximum delays through the logic

$$T_{CLK} > t_{c\text{-}q} + t_{LM} + t_{su} - \delta + 2\, t_{ji}$$

Skew can be either positive or negative

Shortest Path

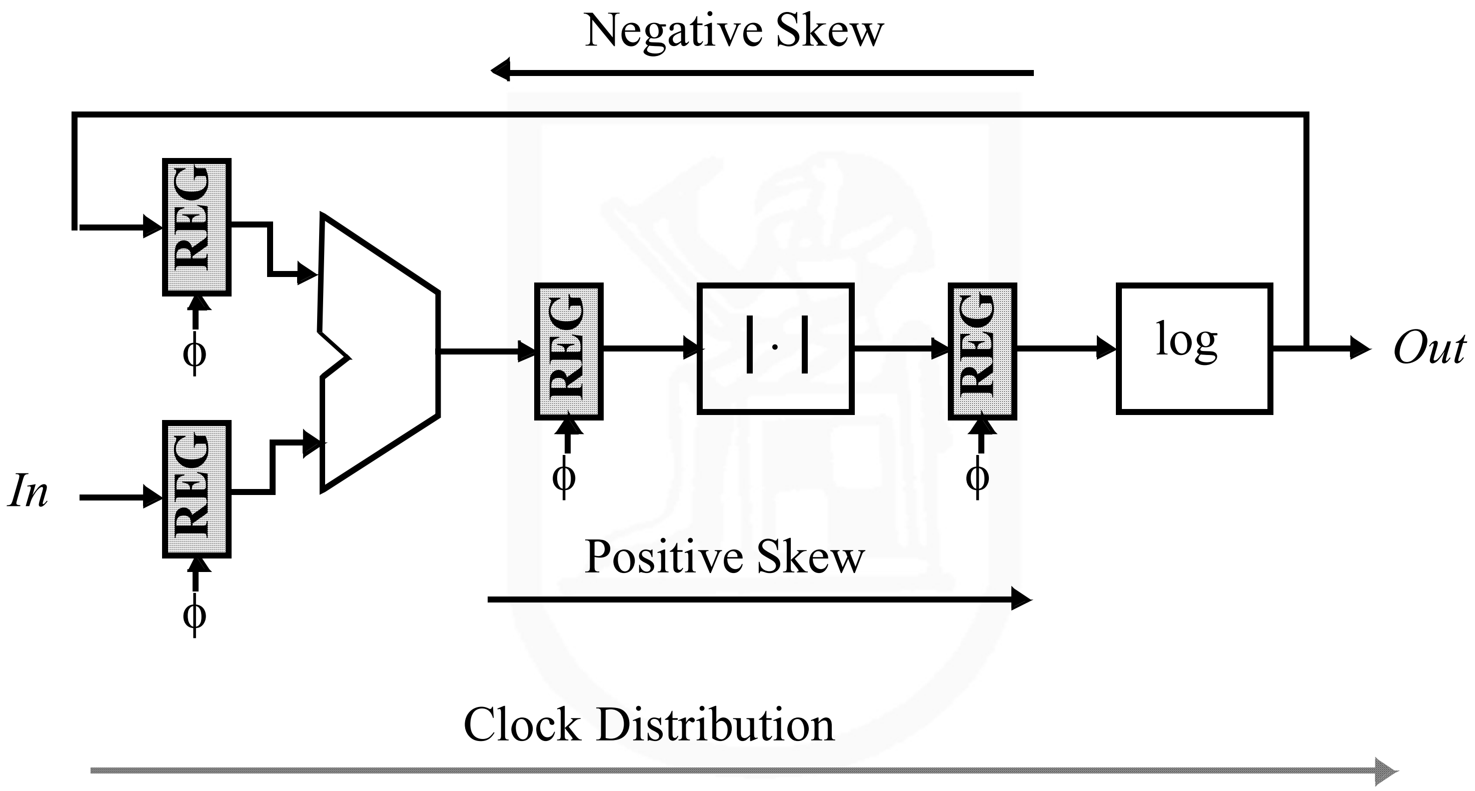

Earliest point of launching
Clk
$t_{c\text{-}q}$
t_{Lm}
Clk
t_H
Nominal clock edge
Data must not arrive before this time

Clock Constraints in Edge-Triggered Systems

Minimum logic delay

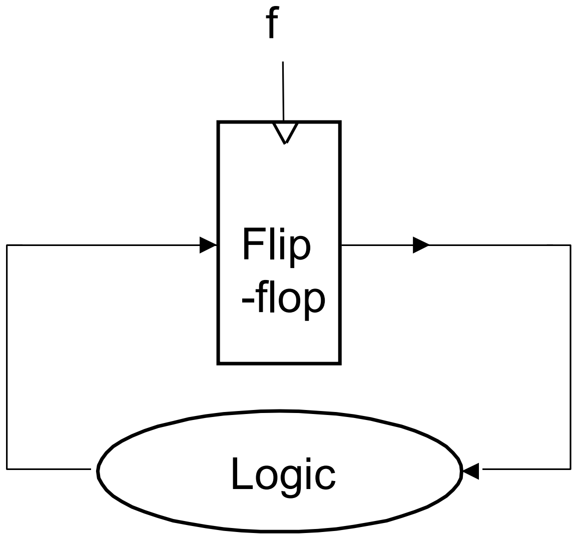
$$t_{c\text{-}q} + t_{Lm} > t_{hold} + \delta$$

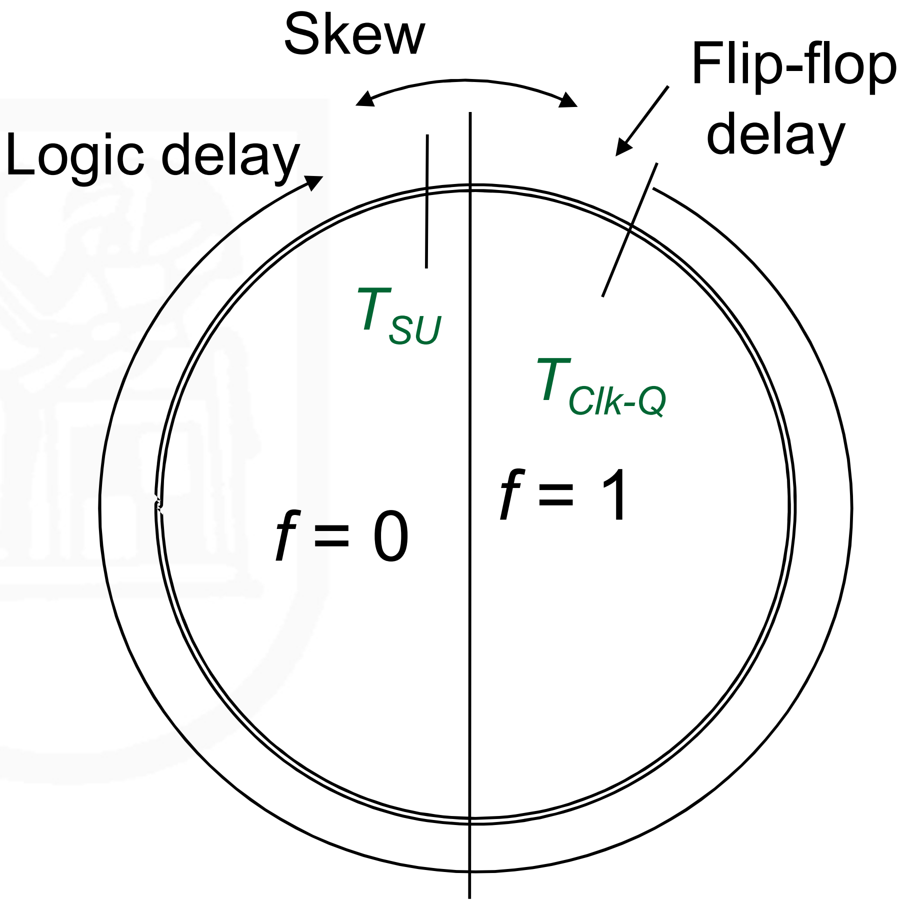
How to counter Clock Skew?



Data and Clock Routing

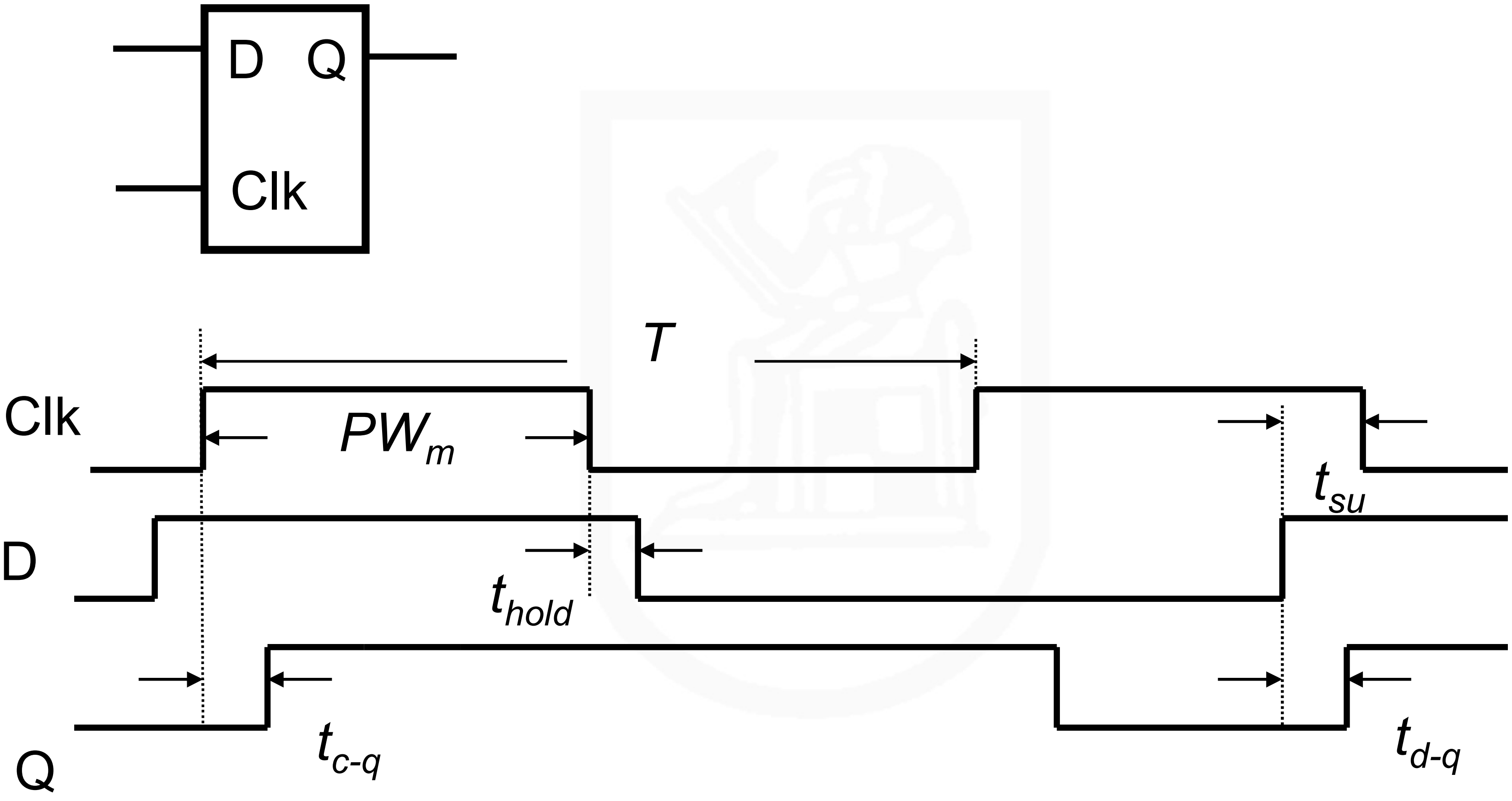
Flip-Flop – Based Timing





Representation after
M. Horowitz, VLSI Circuits 1996.

Latch Parameters

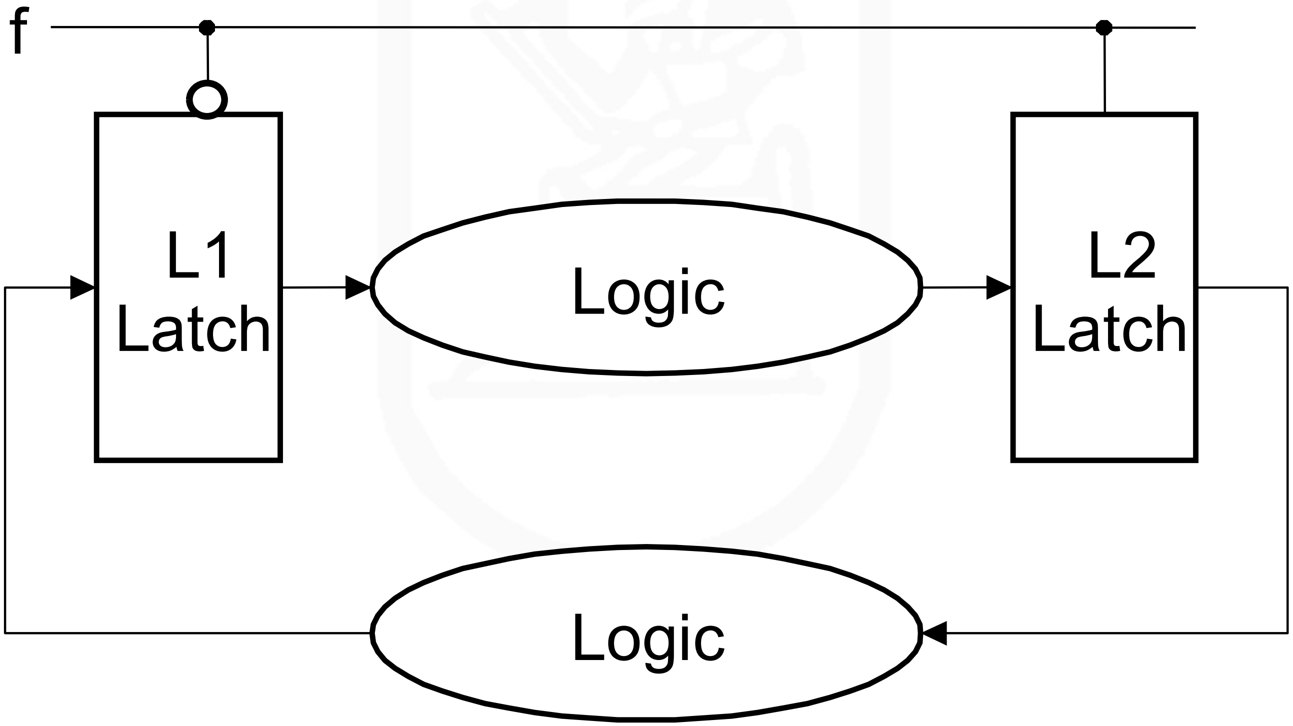


Delays can be different for rising and falling data transitions

Latch-Based Design

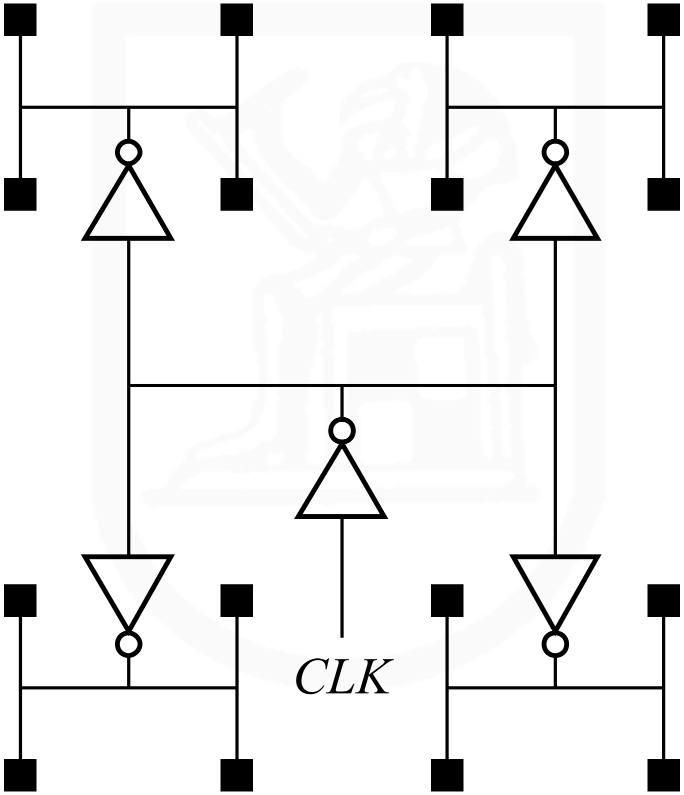
L1 latch is transparent
when f = 0

L2 latch is transparent
when f = 1



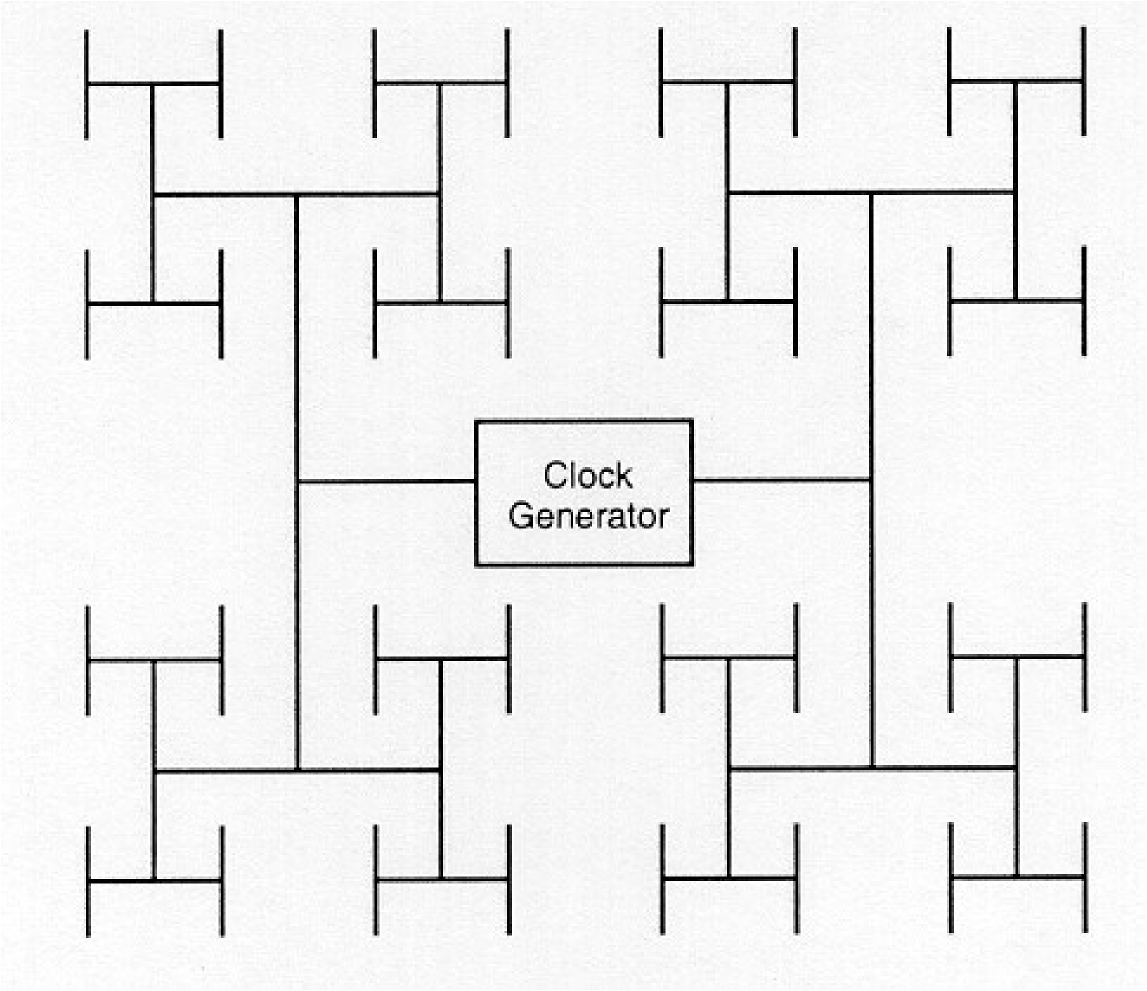
Clock Distribution

H-tree



Clock is distributed in a tree-like fashion

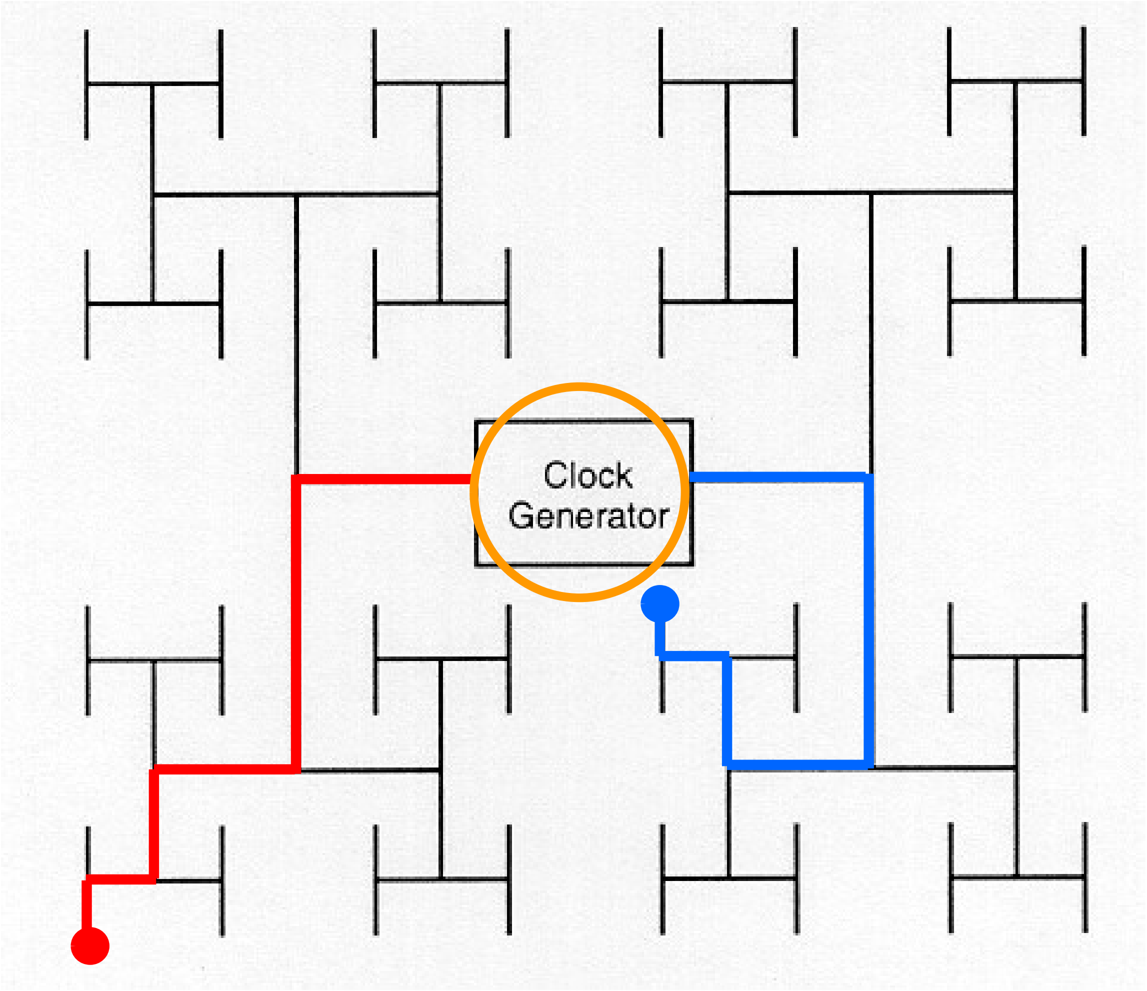
H-Tree



Goal:
Try to equalize clock distribution path lengths to minimize the skew !

H-tree clock distribution network

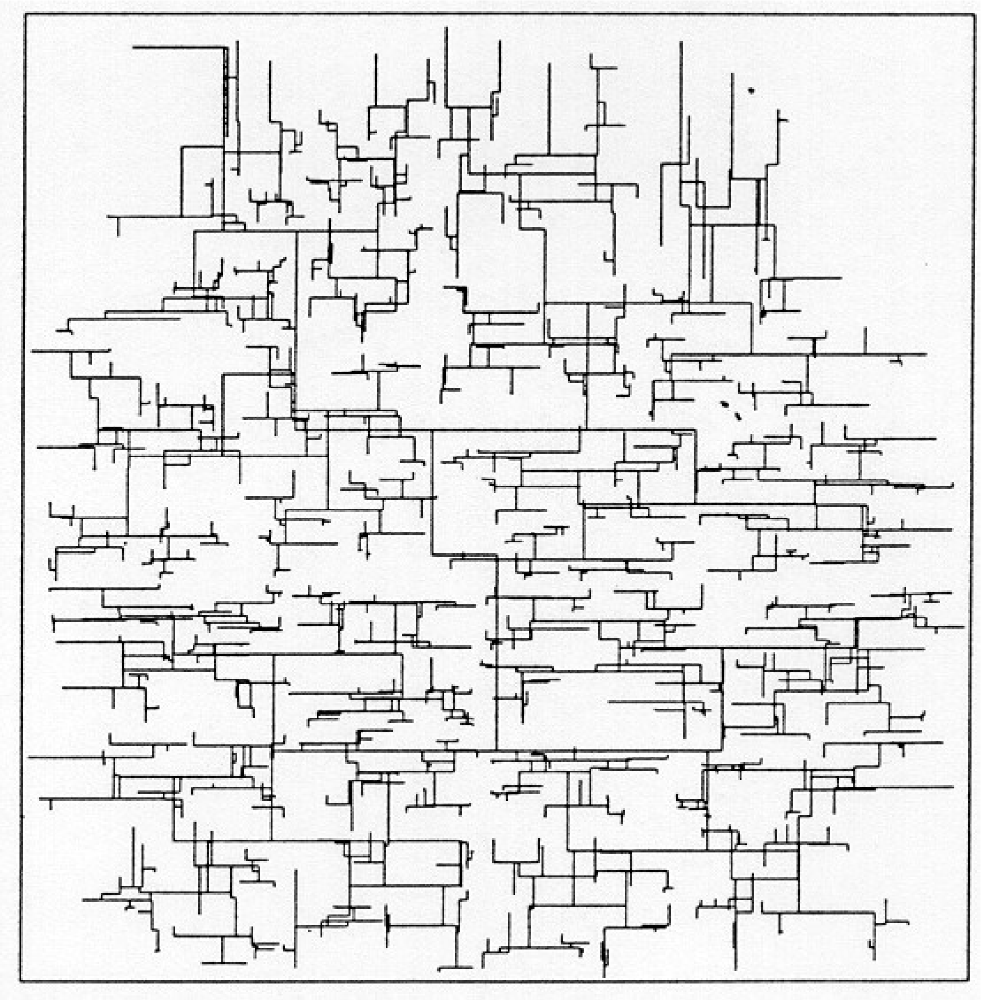
H-Tree



H-tree clock distribution network

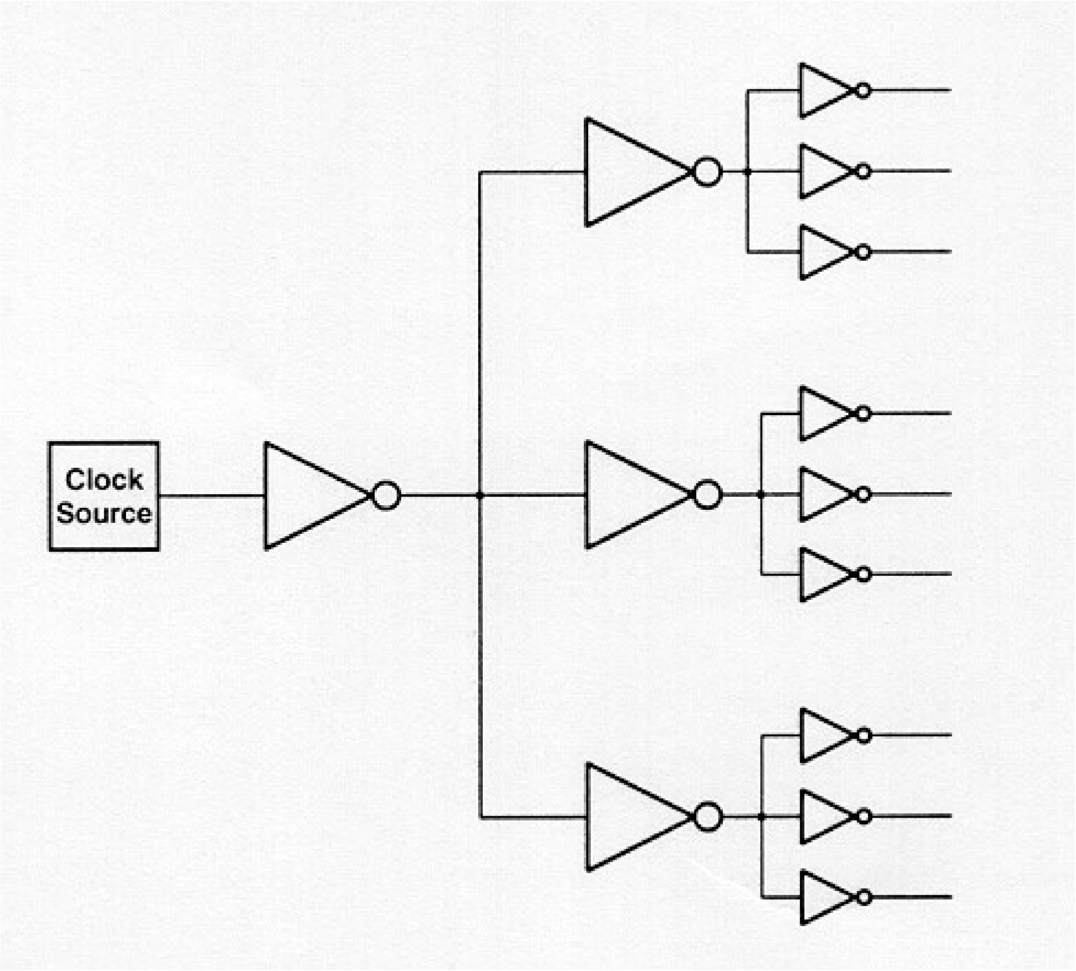
Goal:
Try to equalize clock distribution path lengths to minimize the skew !

H-Tree



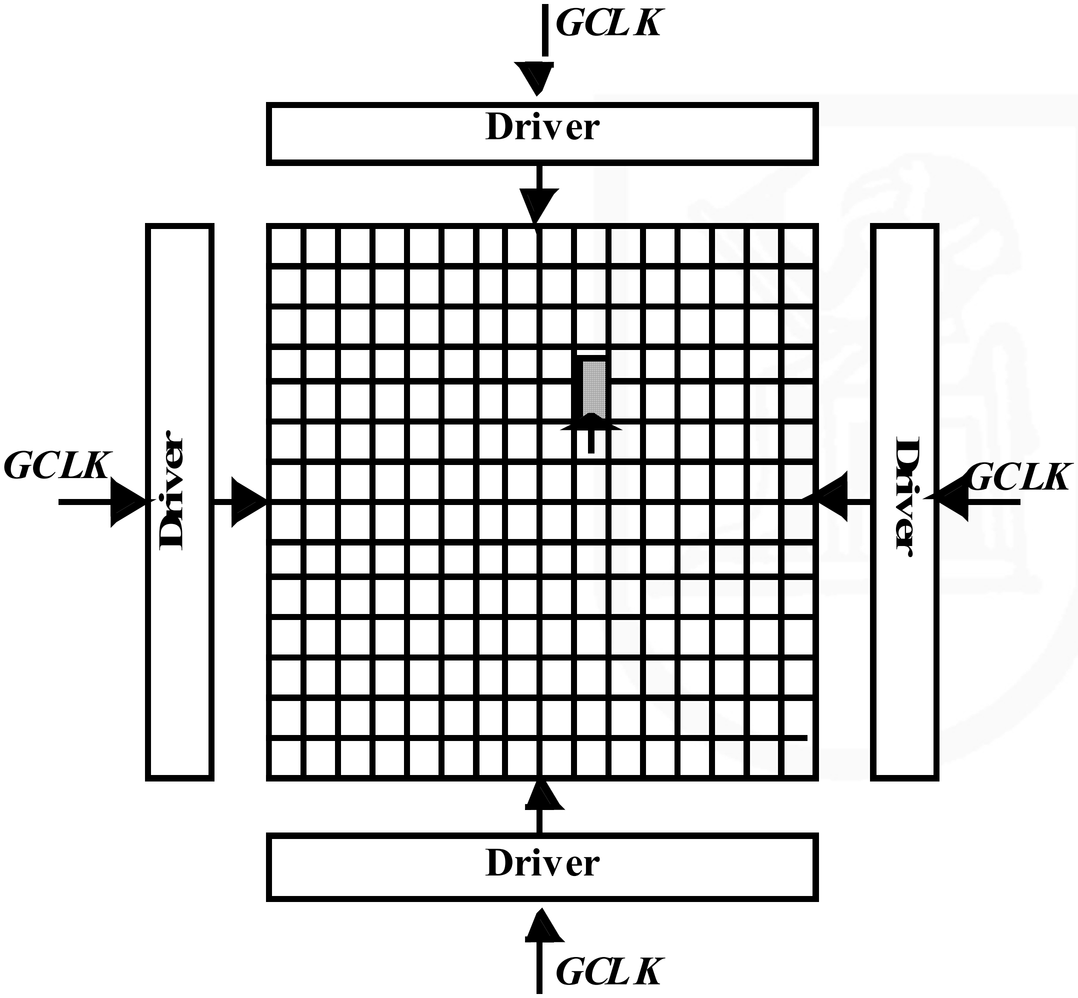
Typical clock distribution network of a real chip

Clock Driver Network



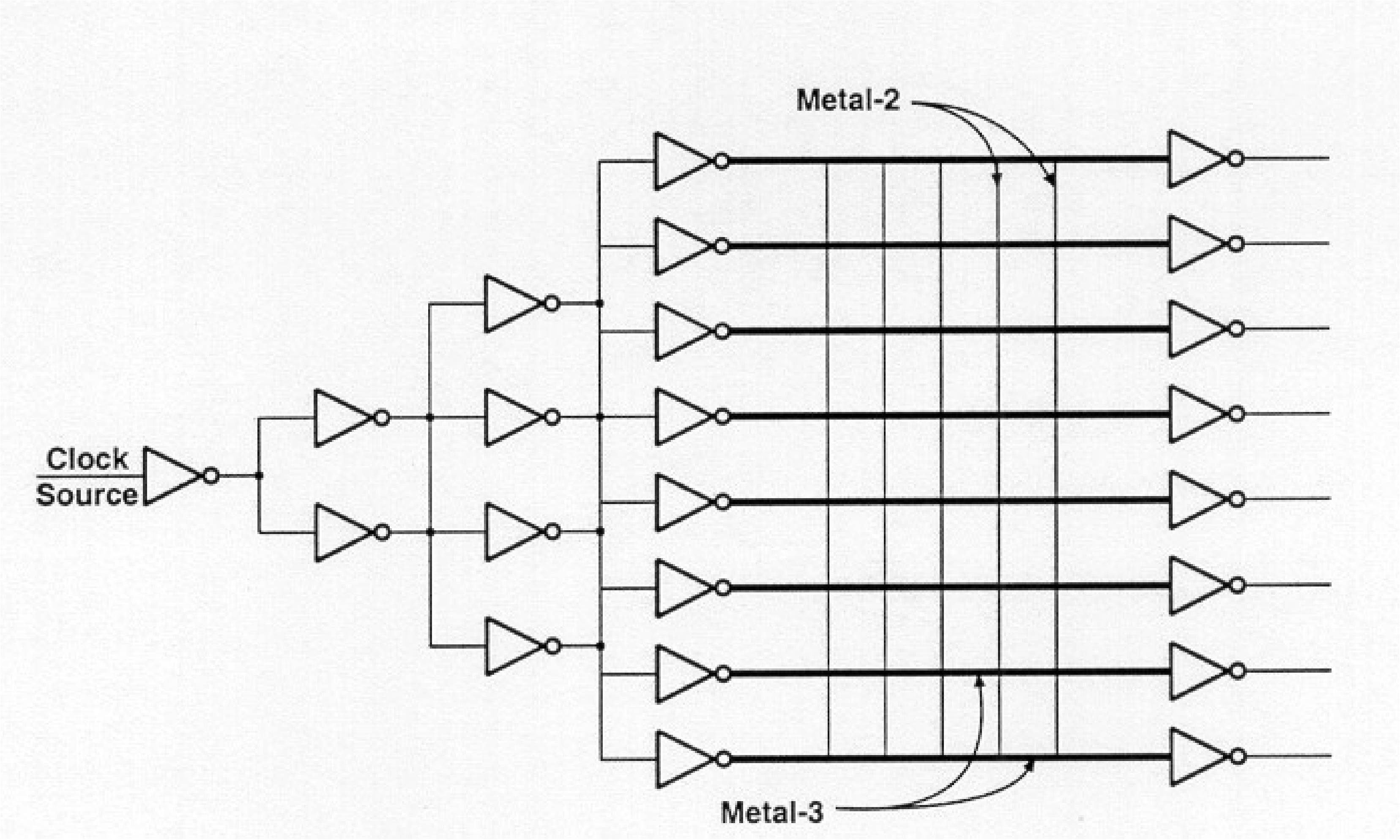
Balanced clock driver network

The Grid System



- *No RC-matching*
- *Large power*
- *Mostly in FPGAs*

Balanced Grid

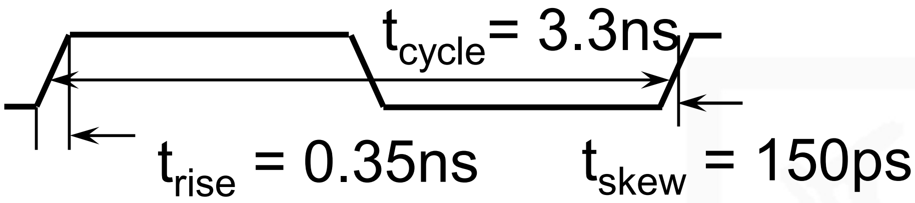


Balanced clock grid (Alpha processor)

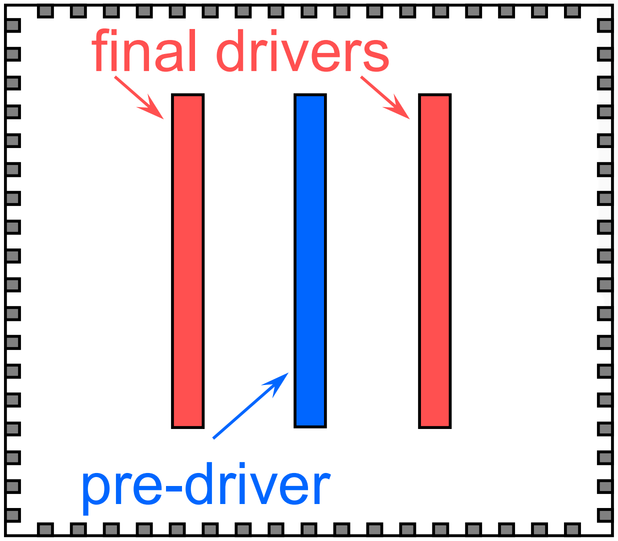
Example: DEC Alpha 21164

- Clock Frequency: 300 MHz - 9.3 Million Transistors
  - Total Clock Load: 3.75 nF
- Power in Clock Distribution network : 20W out of 50
- Uses Two Level Clock Distribution:
  - Single 6-stage driver at center of chip
  - Secondary buffers drive left and right side
  - Clock grid in Metal3 and Metal4

21164 Clocking



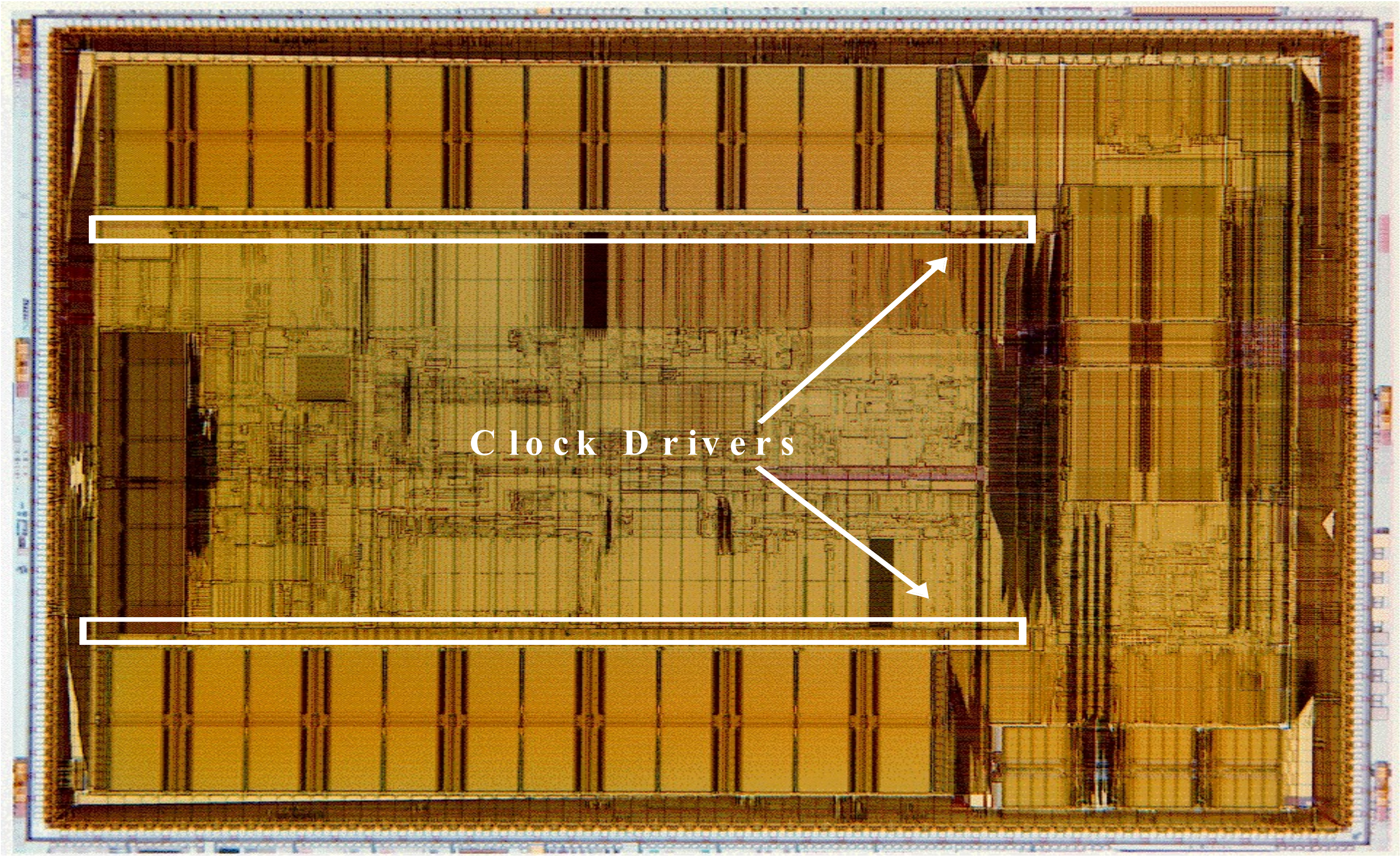
Clock waveform



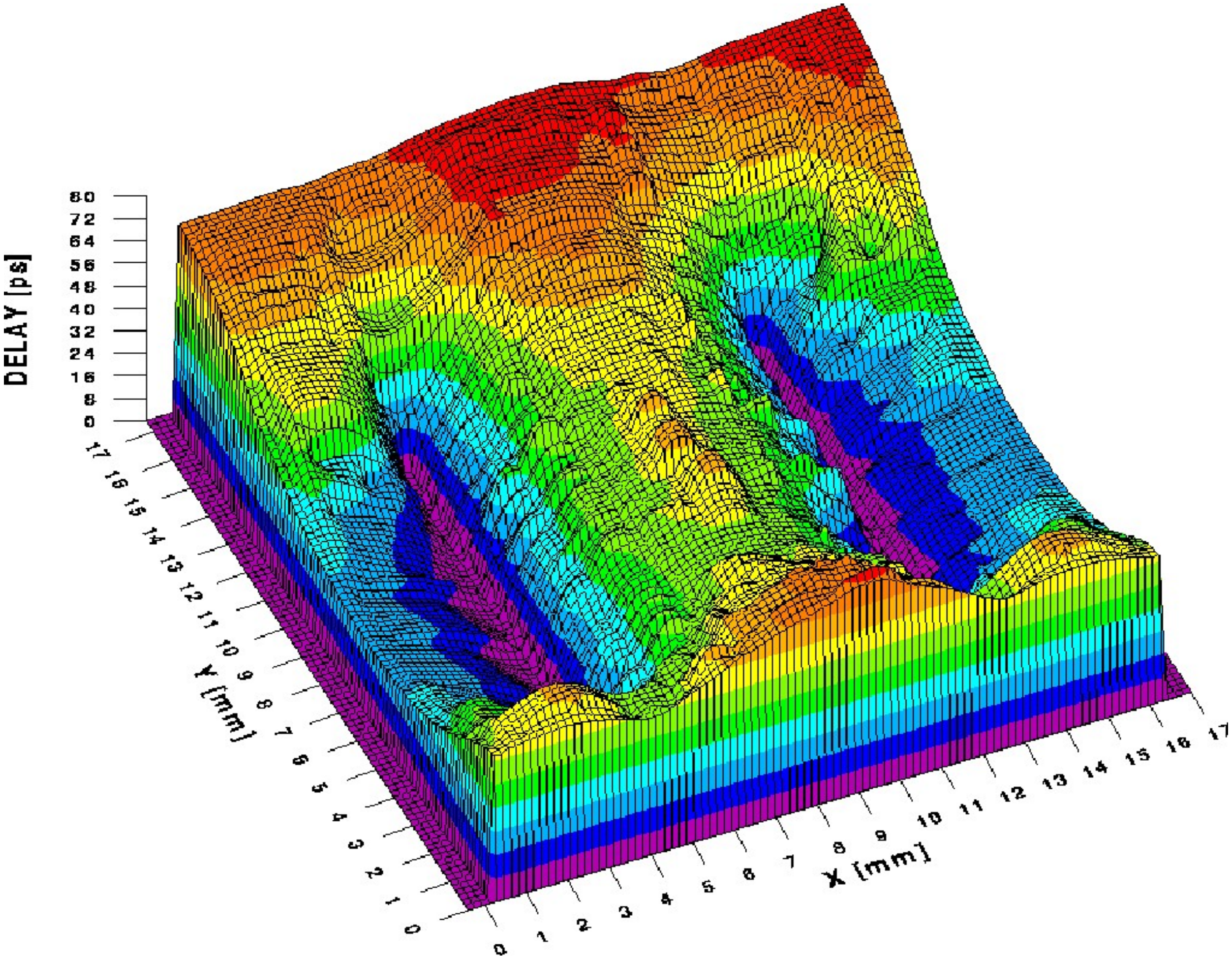
Location of clock driver on die

- **2 phase single wire clock, distributed globally**
- **2 distributed driver channels**
  - Reduced RC delay/skew
  - Improved thermal distribution
  - 3.75nF clock load
- **Local inverters for latching**
- **Conditional clocks in caches to reduce power**
- **More complex race checking**
- **Device variation**

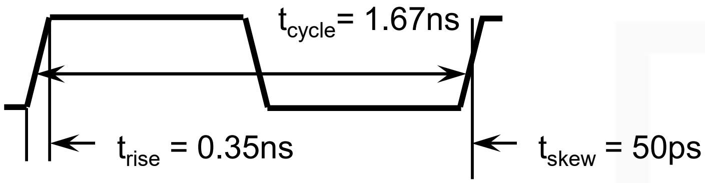
21164 Clocking



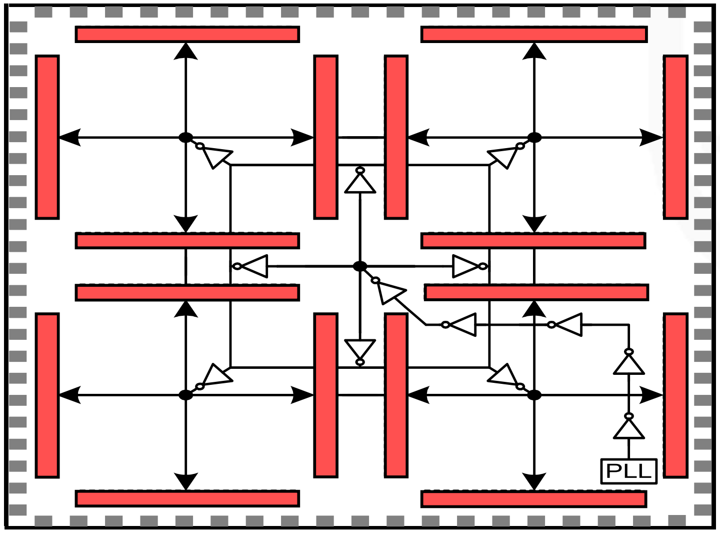
Clock Skew in Alpha Processor



EV6 (Alpha 21264)
Clocking 600 MHz – 0.35um CMOS

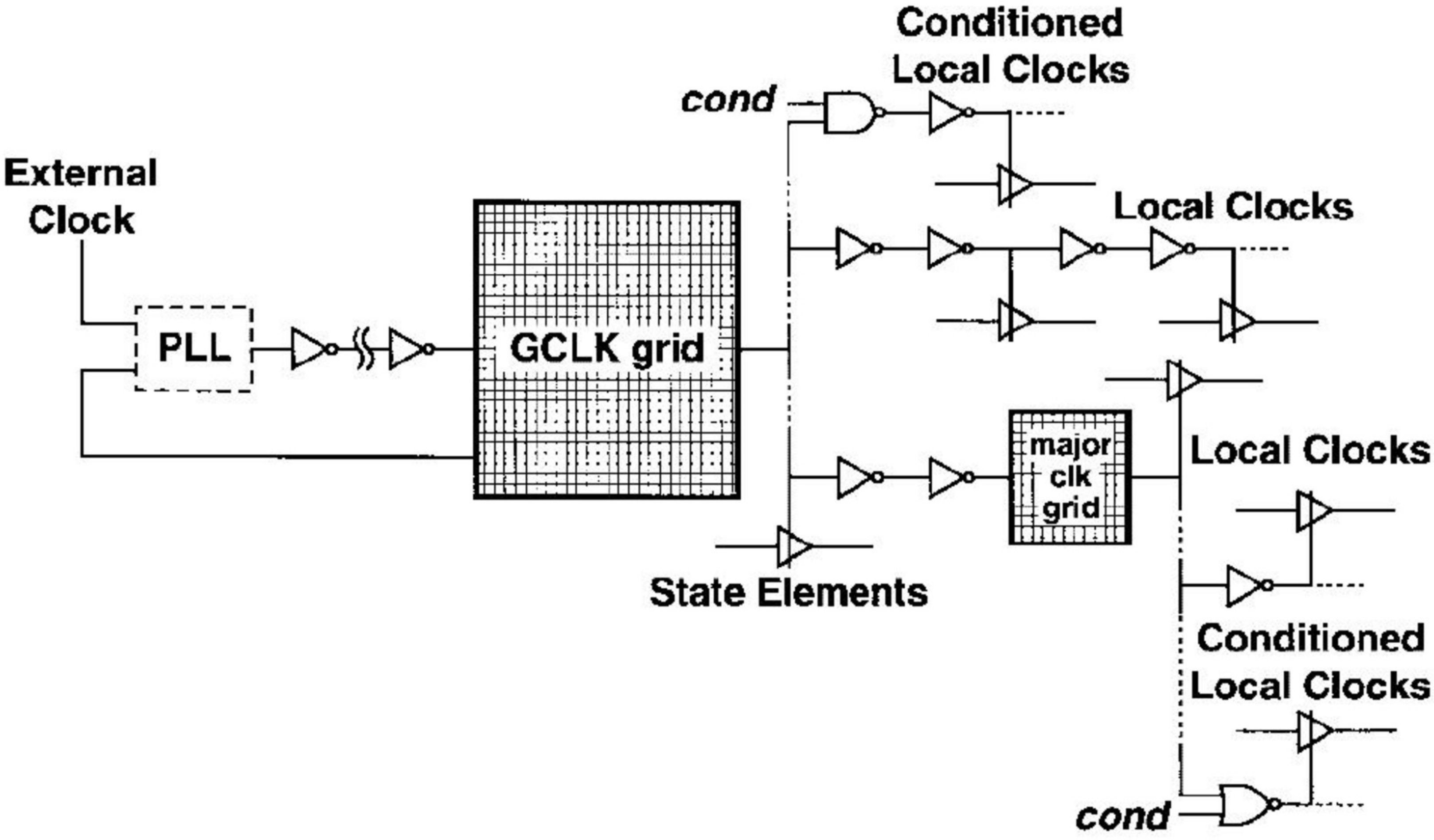


Global clock waveform

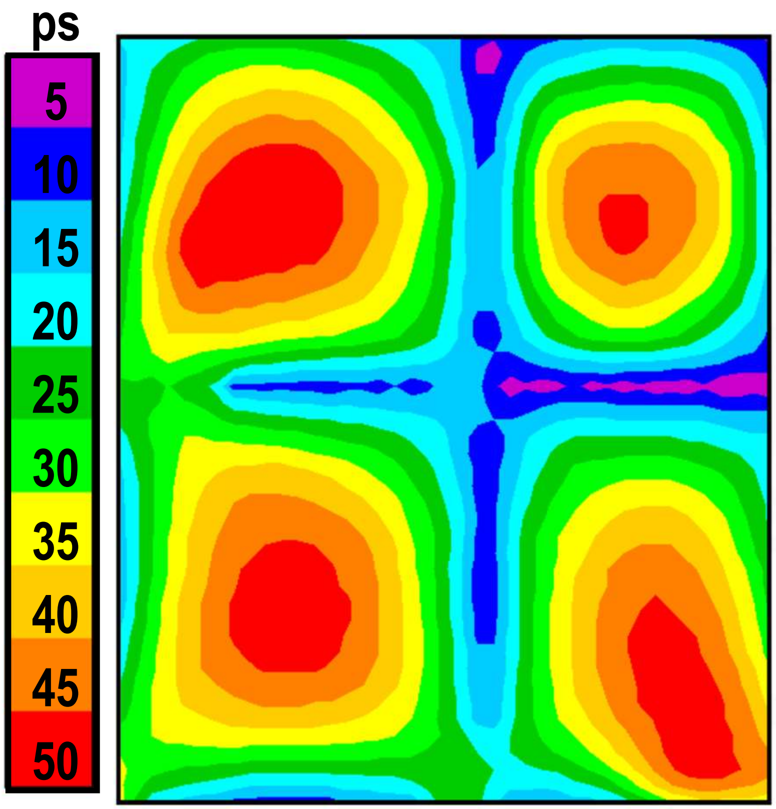


- **2 Phase, with multiple conditional buffered clocks**
  - 2.8 nF clock load
  - 40 cm final driver width
- **Local clocks can be gated "off" to save power**
- **Reduced load/skew**
- **Reduced thermal issues**
- **Multiple clocks complicate race checking**

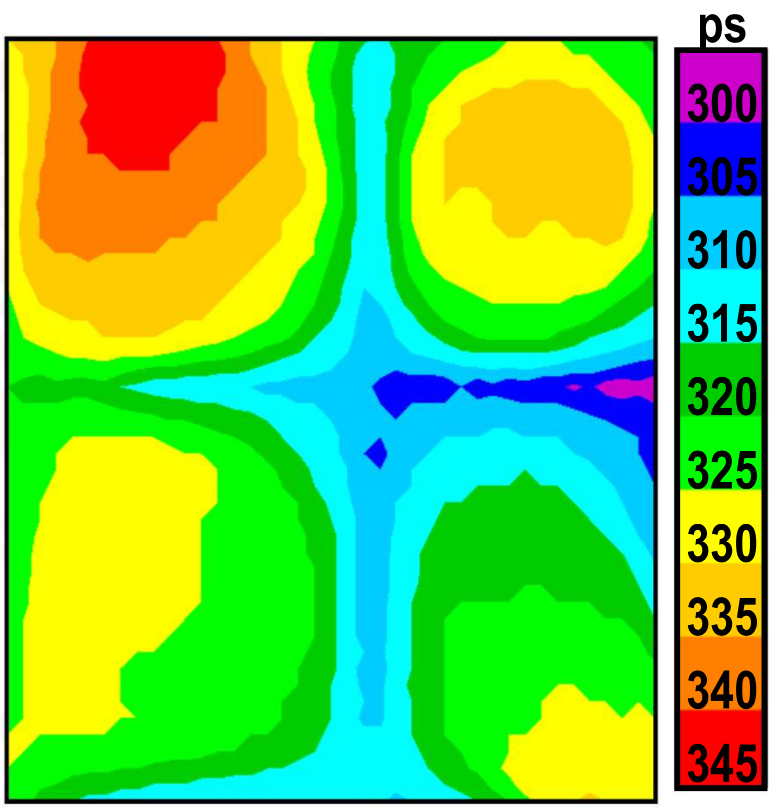
21264 Clocking



EV6 Clock Results



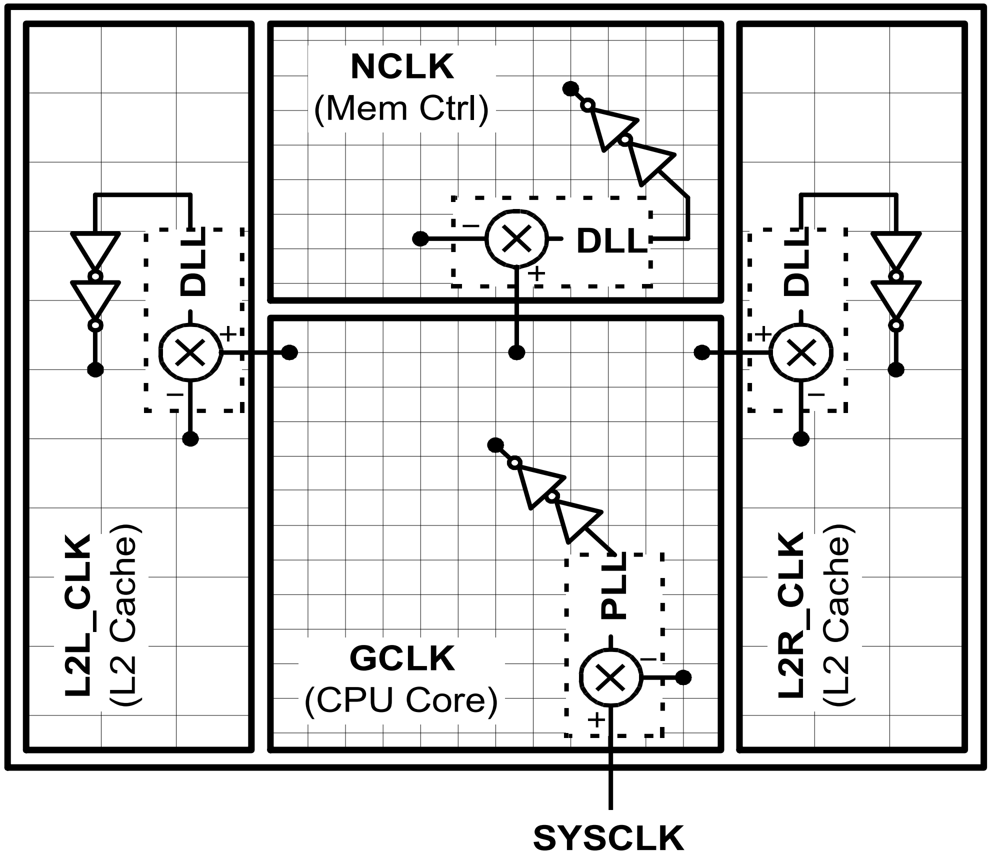
GCLK Skew
(at Vdd/2 Crossings)



GCLK Rise Times
(20% to 80% Extrapolated to 0% to 100%)

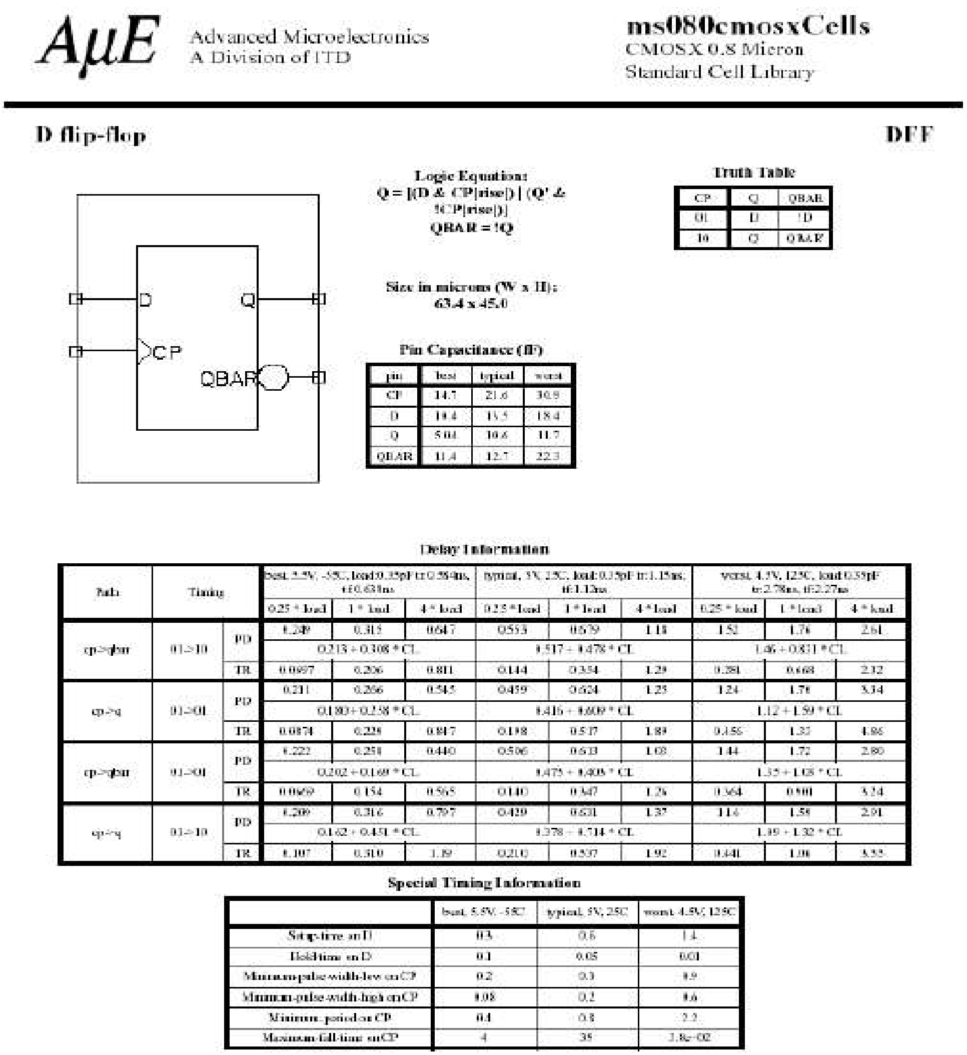
EV7 Clock Hierarchy

Active Skew Management and Multiple Clock Domains



- + Widely dispersed drivers
- + DLLs compensate static and low-frequency variation
- + Divides design and verification effort
- - DLL design and verification is added work
- + Tailored clocks

Cell Library Example



AμE Advanced Microelectronics
A Division of ITD

ms080cmosxCells
CMOSX 0.8 Micron
Standard Cell Library

D flip-flop **DFF**

Logic Equation:
Q = [(D & CP[rise]) | (Q' & !CP[rise])]
QBAR = !Q

Truth Table

CP	Q	QBAR
01	D	!D
10	Q	QBAR

Size in microns (W x H):
63.4 x 45.0

Pin Capacitance (fF)

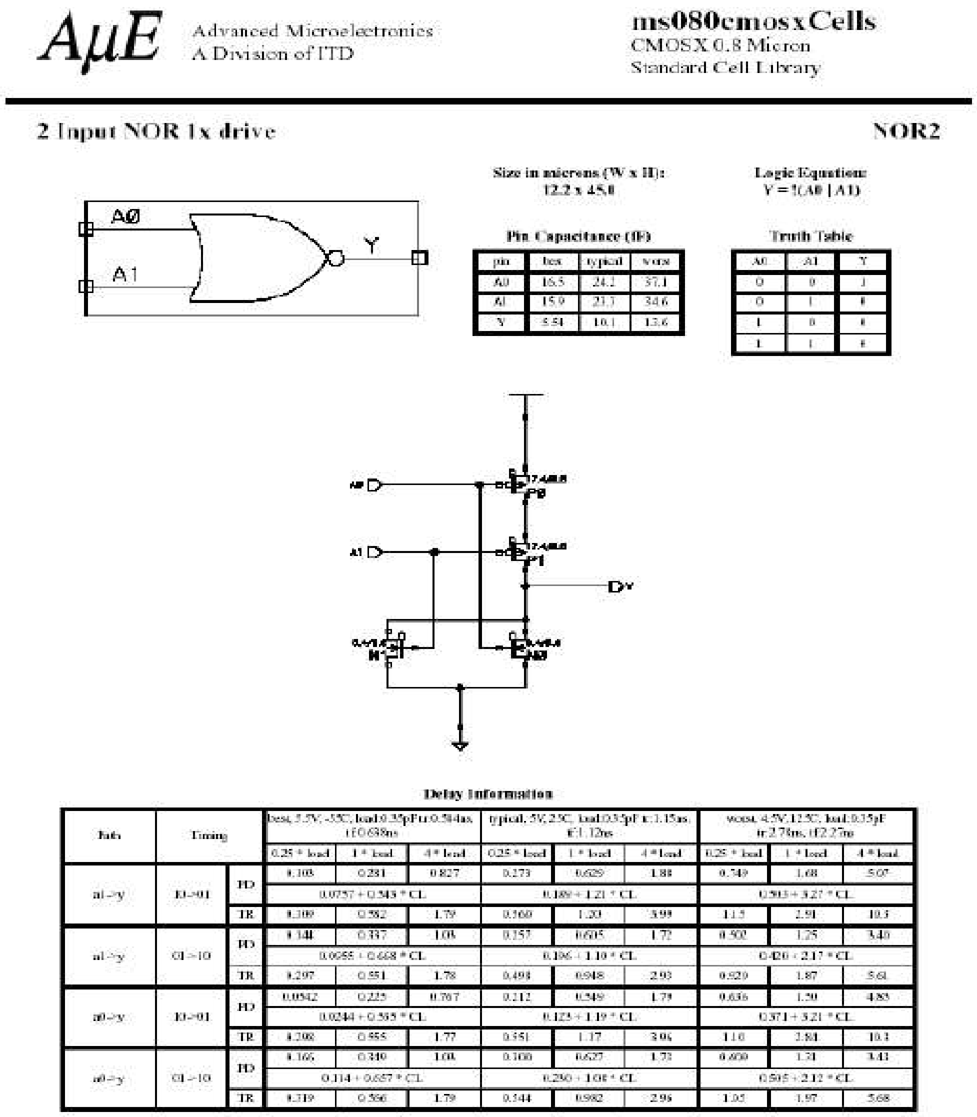
pin	best	typical	worst
CP	14.7	21.6	30.9
D	10.4	15.5	18.4
Q	5.04	10.6	11.7
QBAR	11.4	12.7	22.3

Delay Information

Path	Timing		best, 5.5V, -55C, load 0.35pF tr:0.584ns, tf:0.638ns			typical, 5V, 25C, load 0.35pF tr:1.15ns, tf:1.12ns			worst, 4.5V, 125C, load 0.35pF tr:2.78ns, tf:2.27ns		
			0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load
cp->qbar	01->10	PD	0.249	0.315	0.647	0.553	0.679	1.18	1.52	1.76	2.61
			0.213 + 0.308 * CL			0.517 + 0.478 * CL			1.46 + 0.831 * CL		
		TR	0.0997	0.206	0.811	0.144	0.334	1.29	0.281	0.668	2.12
cp->q	01->01	PD	0.211	0.266	0.545	0.459	0.624	1.25	1.24	1.78	3.34
			0.180 + 0.258 * CL			0.416 + 0.609 * CL			1.12 + 1.59 * CL		
		TR	0.0874	0.228	0.847	0.198	0.517	1.89	0.456	1.33	4.86
cp->qbar	01->01	PD	0.222	0.251	0.440	0.506	0.613	1.08	1.44	1.72	2.80
			0.202 + 0.169 * CL			0.475 + 0.408 * CL			1.35 + 1.08 * CL		
		TR	0.0669	0.154	0.565	0.140	0.347	1.26	0.564	0.901	3.24
cp->q	01->10	PD	0.209	0.316	0.797	0.429	0.631	1.37	1.16	1.58	2.91
			0.162 + 0.451 * CL			0.378 + 0.714 * CL			1.19 + 1.32 * CL		
		TR	0.107	0.310	1.19	0.210	0.537	1.92	0.441	1.08	3.55

Special Timing Information

	best, 5.5V, -55C	typical, 5V, 25C	worst, 4.5V, 125C
Setup-time on D	0.3	0.6	1.4
Hold-time on D	0.1	0.05	0.01
Minimum-pulse-width-low on CP	0.2	0.3	0.9
Minimum-pulse-width-high on CP	0.08	0.2	0.6
Minimum-period on CP	0.4	0.8	2.2
Maximum-fall-time on CP	4	35	3.8e+02



AμE Advanced Microelectronics
A Division of ITD

ms080cmosxCells
CMOSX 0.8 Micron
Standard Cell Library

2 Input NOR 1x drive **NOR2**

AØ
A1
Y

Size in microns (W x H):
12.2 x 45.0

Logic Equation:
Y = !(A0 | A1)

Pin Capacitance (fF)

pin	best	typical	worst
A0	16.5	24.2	37.1
A1	15.9	21.3	34.6
Y	5.54	10.1	13.6

Truth Table

A0	A1	Y
0	0	1
0	1	0
1	0	0
1	1	0

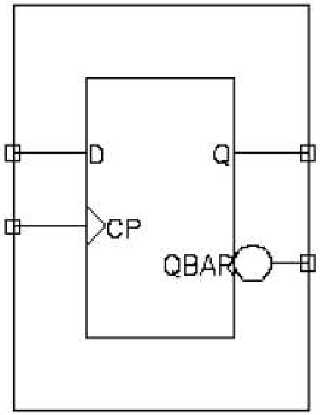
A0
A1
17.4/0.8
P0
17.4/0.8
P1
Y
9.4/0.8
N1
9.4/0.8
N0

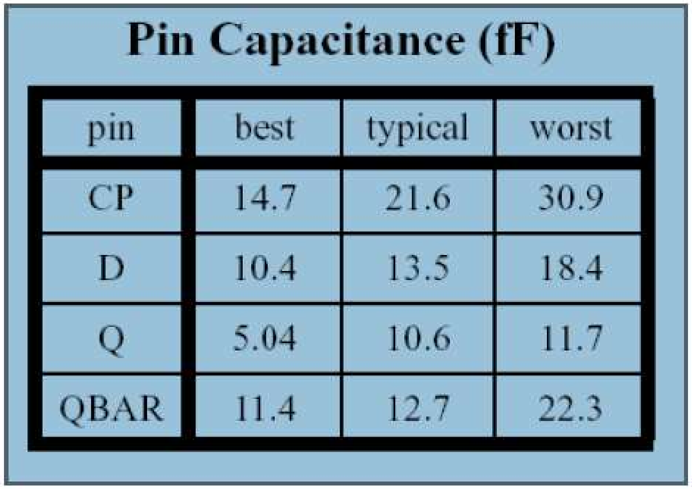
Delay Information

Path	Timing		best, 5.5V, -55C, load 0.35pF tr:0.584ns, tf:0.638ns			typical, 5V, 25C, load 0.35pF tr:1.15ns, tf:1.12ns			worst, 4.5V, 125C, load 0.35pF tr:2.78ns, tf:2.27ns		
			0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load
a1->y	10->01	PD	0.103	0.281	0.827	0.273	0.629	1.88	0.749	1.68	5.07
			0.0757 + 0.543 * CL			0.189 + 1.21 * CL			0.515 + 3.27 * CL		
		TR	0.109	0.582	1.79	0.360	1.20	3.99	1.15	2.91	10.3
a1->y	01->10	PD	0.141	0.337	1.03	0.157	0.605	1.72	0.502	1.25	3.40
			0.0955 + 0.668 * CL			0.106 + 1.10 * CL			0.420 + 2.17 * CL		
		TR	0.297	0.551	1.78	0.498	0.948	2.93	0.920	1.87	5.61
a0->y	10->01	PD	0.0542	0.225	0.767	0.212	0.549	1.79	0.636	1.50	4.85
			0.0244 + 0.585 * CL			0.123 + 1.19 * CL			0.371 + 3.21 * CL		
		TR	0.202	0.555	1.77	0.551	1.17	3.96	1.10	2.81	10.3
a0->y	01->10	PD	0.166	0.349	1.03	0.100	0.627	1.73	0.600	1.31	3.43
			0.114 + 0.657 * CL			0.250 + 1.08 * CL			0.505 + 2.12 * CL		
		TR	0.319	0.566	1.79	0.344	0.982	2.96	1.05	1.97	5.68

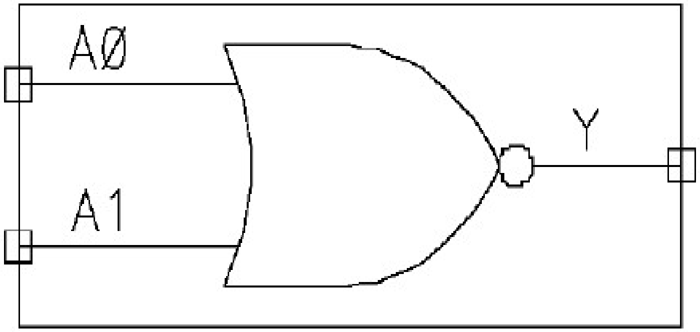
C_{load} on Flip-Flop

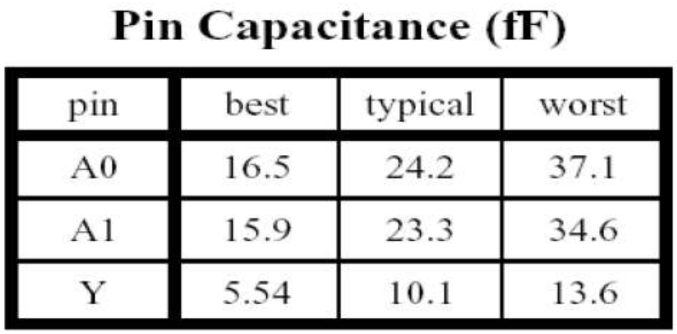
- C_{load} = output capacitance of flip-flop + input capacitance of NOR gate




Pin Capacitance (fF)

pin	best	typical	worst
CP	14.7	21.6	30.9
D	10.4	13.5	18.4
Q	5.04	10.6	11.7
QBAR	11.4	12.7	22.3

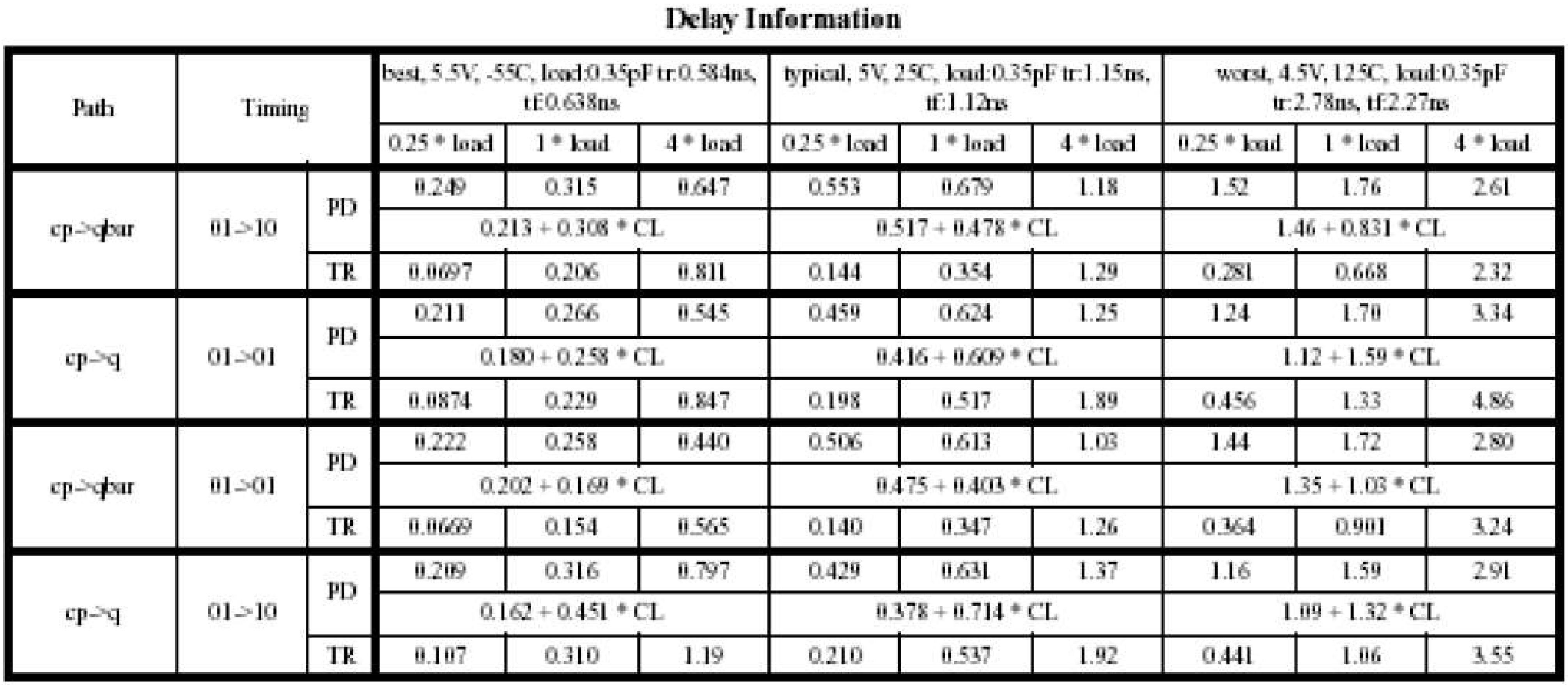



Pin Capacitance (fF)

pin	best	typical	worst
A0	16.5	24.2	37.1
A1	15.9	23.3	34.6
Y	5.54	10.1	13.6

Timing Tables

- **Predict delay**
  - **Note delay different for rising and falling edges**

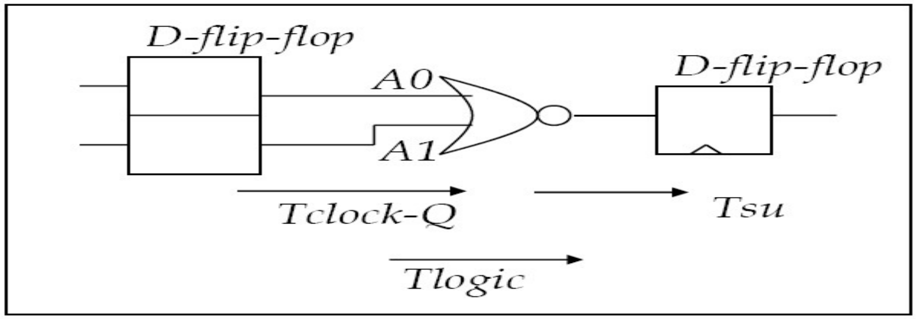


Delay Information

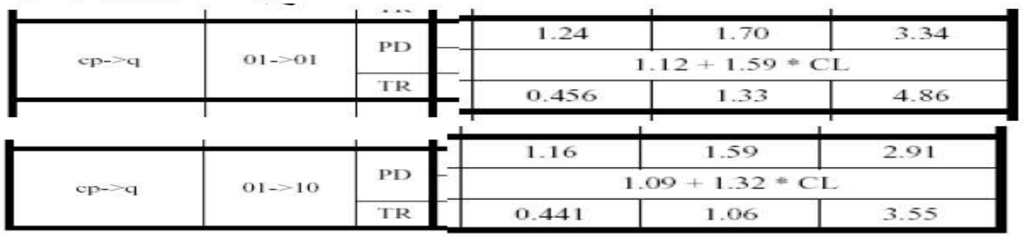
Path	Timing		best, 5.5V, -55C, load:0.35pF tr:0.584ns, tf:0.638ns			typical, 5V, 25C, load:0.35pF tr:1.15ns, tf:1.12ns			worst, 4.5V, 125C, load:0.35pF tr:2.78ns, tf:2.27ns		
			0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load	0.25 * load	1 * load	4 * load
cp->qbar	01->10	PD	0.249	0.315	0.647	0.553	0.679	1.18	1.52	1.76	2.61
			0.213 + 0.308 * CL			0.517 + 0.478 * CL			1.46 + 0.831 * CL		
		TR	0.0697	0.206	0.811	0.144	0.354	1.29	0.281	0.668	2.32
cp->q	01->01	PD	0.211	0.266	0.545	0.459	0.624	1.25	1.24	1.70	3.34
			0.180 + 0.258 * CL			0.416 + 0.609 * CL			1.12 + 1.59 * CL		
		TR	0.0874	0.229	0.847	0.198	0.517	1.89	0.456	1.33	4.86
cp->qbar	01->01	PD	0.222	0.258	0.440	0.506	0.613	1.03	1.44	1.72	2.80
			0.202 + 0.169 * CL			0.475 + 0.403 * CL			1.35 + 1.03 * CL		
		TR	0.0669	0.154	0.565	0.140	0.347	1.26	0.364	0.901	3.24
cp->q	01->10	PD	0.209	0.316	0.797	0.429	0.631	1.37	1.16	1.59	2.91
			0.162 + 0.451 * CL			0.378 + 0.714 * CL			1.09 + 1.32 * CL		
		TR	0.107	0.310	1.19	0.210	0.537	1.92	0.441	1.06	3.55

Data Sheet Example

- **Using timing approximations in the datasheet, what is the maximum clock frequency for this circuit (ignore wire load, Tskew)?**



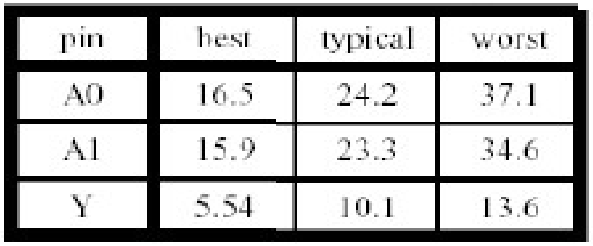
Tclock-Q:



cp->q	01->01	PD	1.24	1.70	3.34
			1.12 + 1.59 * CL		
		TR	0.456	1.33	4.86
cp->q	01->10	PD	1.16	1.59	2.91
			1.09 + 1.32 * CL		
		TR	0.441	1.06	3.55

CL:

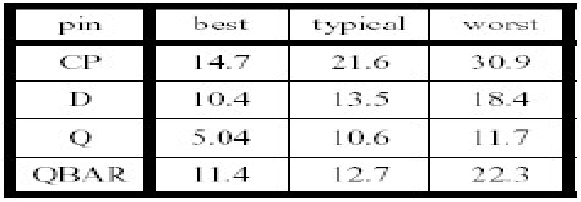
Pin Capacitance (fF)



pin	best	typical	worst
A0	16.5	24.2	37.1
A1	15.9	23.3	34.6
Y	5.54	10.1	13.6

DFF

Pin Capacitance (fF)



pin	best	typical	worst
CP	14.7	21.6	30.9
D	10.4	13.5	18.4
Q	5.04	10.6	11.7
QBAR	11.4	12.7	22.3

Q -> A0:

C_{L_max} = 0.0371 + 0.0117 pF = 0.0488 pF

$T_{cp\text{-}Q\_max}$ = max(1.12+1.59*CL, 1.09 + 1.32*CL)

= max(1.12+1.59*0.0488,

1.09 + 1.32*0.0488)

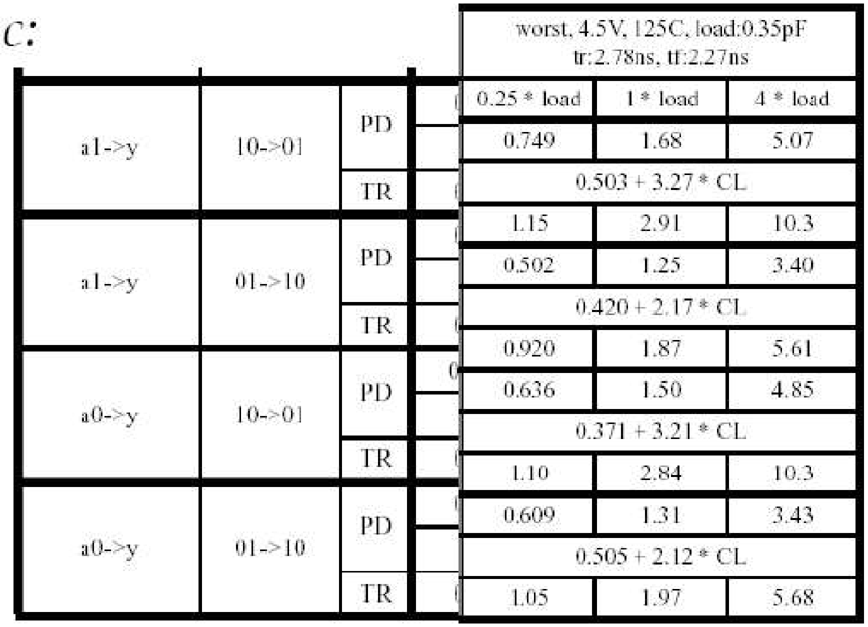
= max(1.2, 1.15) = 1.2 ns

Q->A1:

$T_{cp\text{-}Q\_max}$ = 1.2 ns

Data Sheet Example

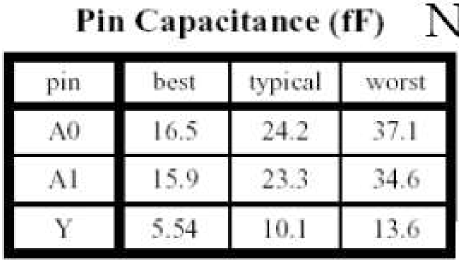
Tlogic:



			worst, 4.5V, 125C, load:0.35pF tr:2.78ns, tf:2.27ns		
			0.25 * load	1 * load	4 * load
a1->y	10->01	PD	0.749	1.68	5.07
			0.503 + 3.27 * CL		
		TR	1.15	2.91	10.3
a1->y	01->10	PD	0.502	1.25	3.40
			0.420 + 2.17 * CL		
		TR	0.920	1.87	5.61
a0->y	10->01	PD	0.636	1.50	4.85
			0.371 + 3.21 * CL		
		TR	1.10	2.84	10.3
a0->y	01->10	PD	0.609	1.31	3.43
			0.505 + 2.12 * CL		
		TR	1.05	1.97	5.68

NOR2

CL:

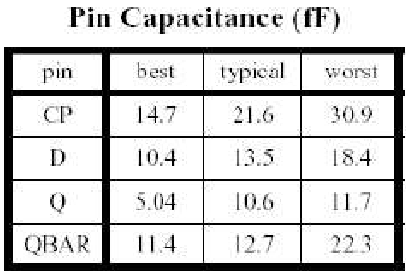


Pin Capacitance (fF)

pin	best	typical	worst
A0	16.5	24.2	37.1
A1	15.9	23.3	34.6
Y	5.54	10.1	13.6

NOR2

DFF



Pin Capacitance (fF)

pin	best	typical	worst
CP	14.7	21.6	30.9
D	10.4	13.5	18.4
Q	5.04	10.6	11.7
QBAR	11.4	12.7	22.3

CL= 0.0136 + 0.0184 = 0.032 pF

From A0 : Tlogic = max (0.503 + 3.27*0.032, 0.420+2.17*0.032)

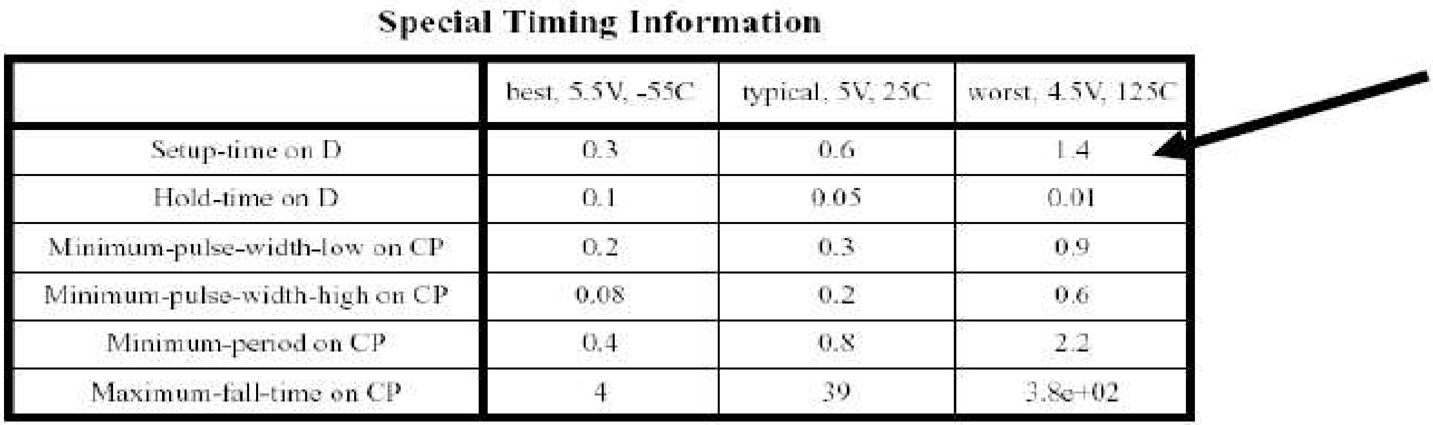
= 0.61 ns

From A1 : Tlogic = max (0.505+2.12*0.032, 0.371 +3.21*0.032)

= 0.57 ns

Data Sheet Example

Tsu



Special Timing Information

	best, 5.5V, -55C	typical, 5V, 25C	worst, 4.5V, 125C
Setup-time on D	0.3	0.6	1.4
Hold-time on D	0.1	0.05	0.01
Minimum-pulse-width-low on CP	0.2	0.3	0.9
Minimum-pulse-width-high on CP	0.08	0.2	0.6
Minimum-period on CP	0.4	0.8	2.2
Maximum-fall-time on CP	4	39	3.8e+02

Tclock > Tcp-Q_max + Tlogic_max + Tsu_max
= 1.2 + 0.61 + 1.4
= 3.21 ns
Fclock < 311 MHz

Delay Metric

- Usual Metric for delay:
  - Delay of an inverter with a fan-out of 4: FO4

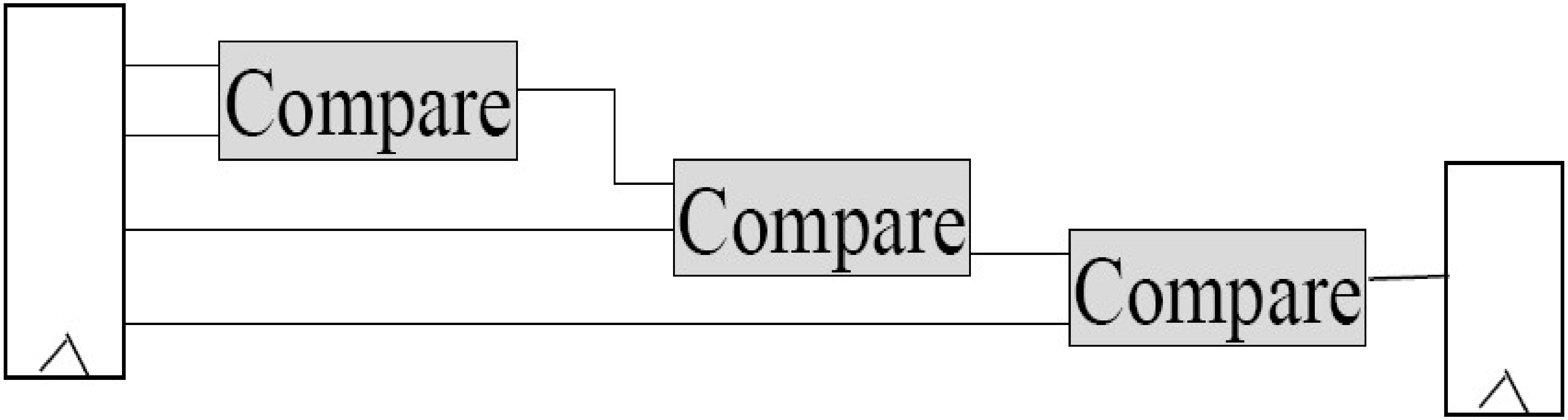
- Estimating FO4:
  - Typical ~ 360 x L_{eff} (ps)
  - Worst Case ~ 600 x L_{eff} (ps)
  - L_{eff} = Effective gate length in um ~ 0.7 x L_{drawn}
  - E.g., in a 0.18um process, L_{eff} = 0.126um and FO4 $\leq$ 75ps

- Examples:

Inverter	= FO4	2-input NAND gate	= 2·FO4
1-bit adder	= 10·FO4	2-input Multiplexer	= 4·FO4
Flip-flop t_{c-q}	= 4·FO4	Flip-flop t_{su} & t_h	= 2·FO4
Clock skew	= 4·FO4	Clock jitter	= 2·FO4

Examples of Improving Timing Performance

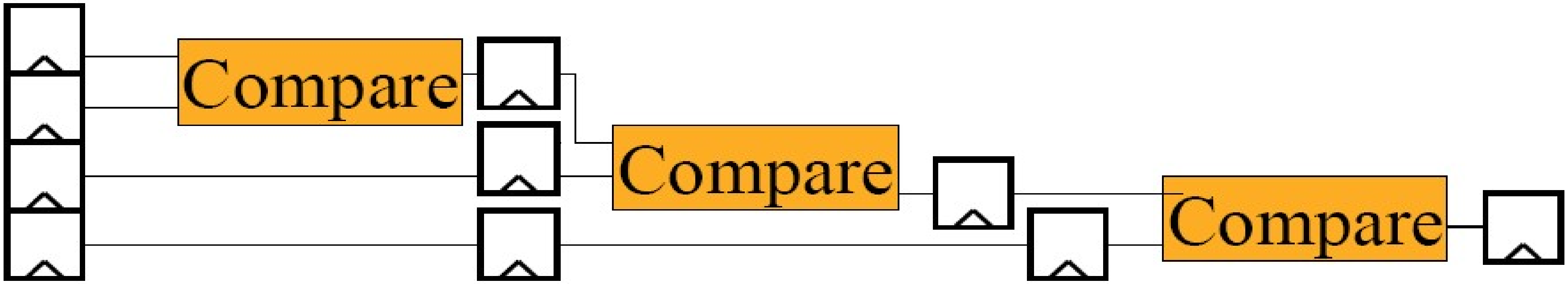
- **Example 1 : Benefits of Pipelining and Parallelism**
- **Example:**



If t_comparator = 20 · FO4, what is the clock period? (Use values on previous page)

Pipelining

- **Replace with:**

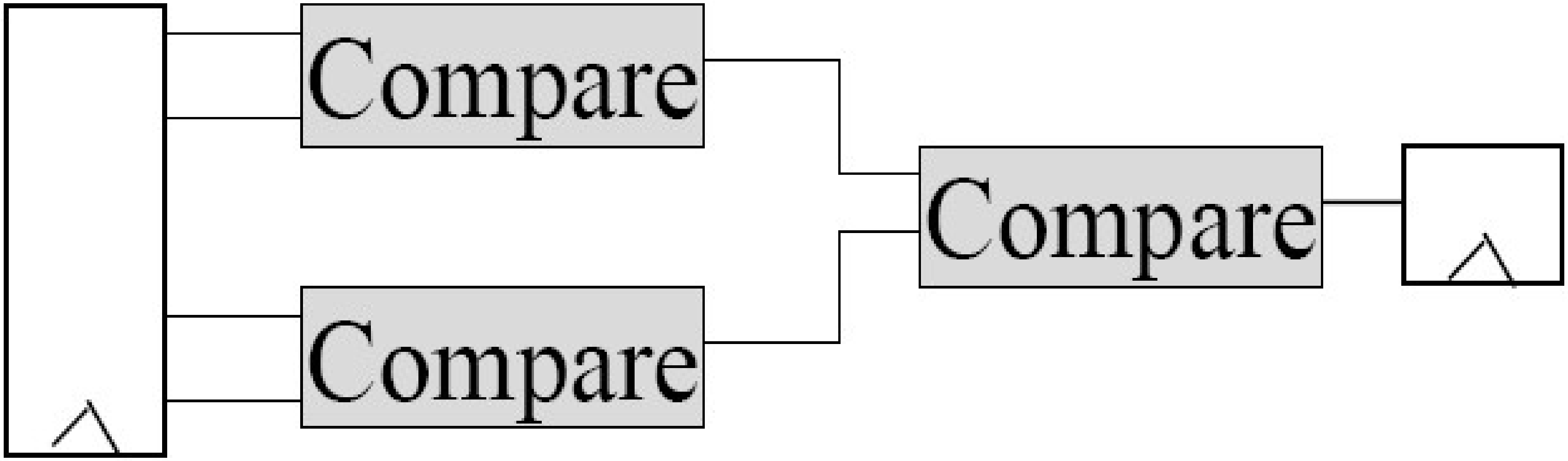


$$T\_cp = t\_ck\text{-}Q + t\_logic + t\_su + t\_skew + t\_jitter$$
$$= 4 + 20 + 2 + 4 + 2 = 32\ FO4$$

- **What is the delay improvement?**
- **What is the drawback?**

Logic Level Parallelism

- **Replace with:**



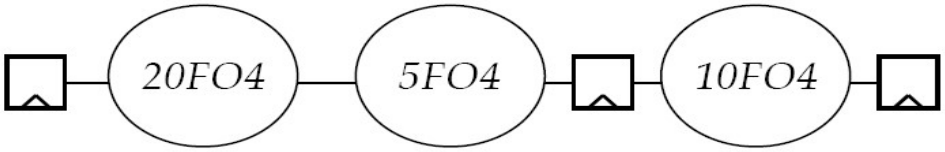
- **Clock Period = 52 · FO-4**
- **No increase in area**

Retiming

- Impact of critical paths can often be reduced by retiming or rebalancing a design.

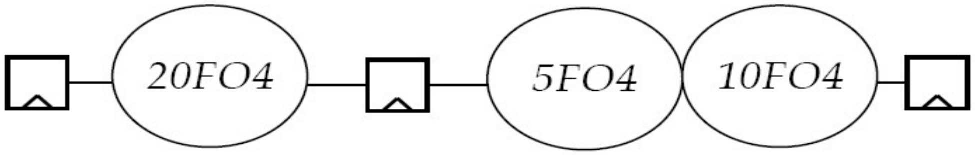
- **Example:**

  - **Before:**



$$T\_cp = 4 + 20 + 5 + 2 + 4 + 2 = 37\ FO4$$

  - **After:**



$$T\_cp = 4 + 20 + 2 + 4 + 2 = 32\ FO4$$

Note: Clock level logic sequence has been changed

Timing Mantra

- One clock, one edge; Flip-flops only
  - For your design (at least for each module), use one clock source and only one edge of that clock.
- Only use edge-triggered flip-flops; why?
- Moving data between different clock domains requires careful timing design and synthesis “scripting”.
- If you need multiple clocks in your design:
  - Make them related by a powers of 2, e.g., 50, 100 and 200 MHz
  - Consider one clock per module
  - Consider resynchronizing using flip-flops between clock domains

Caveat

- Tools and designers are getting better at using latches and multi-phase clocks. However, this requires experience to get correct.