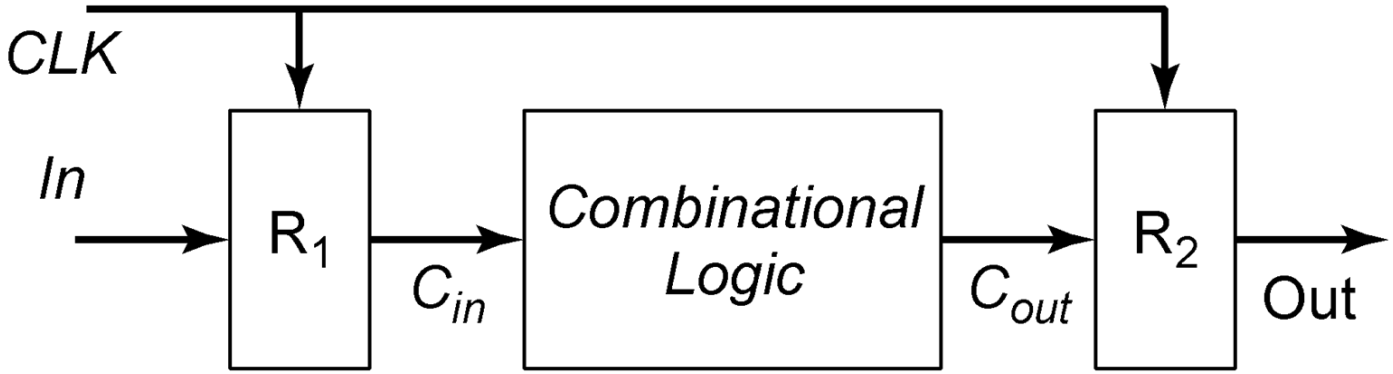
Digital Integrated Circuits
A Design Perspective

Jan M. Rabaey
Anantha Chandrakasan
Borivoje Nikolić

Timing Issues

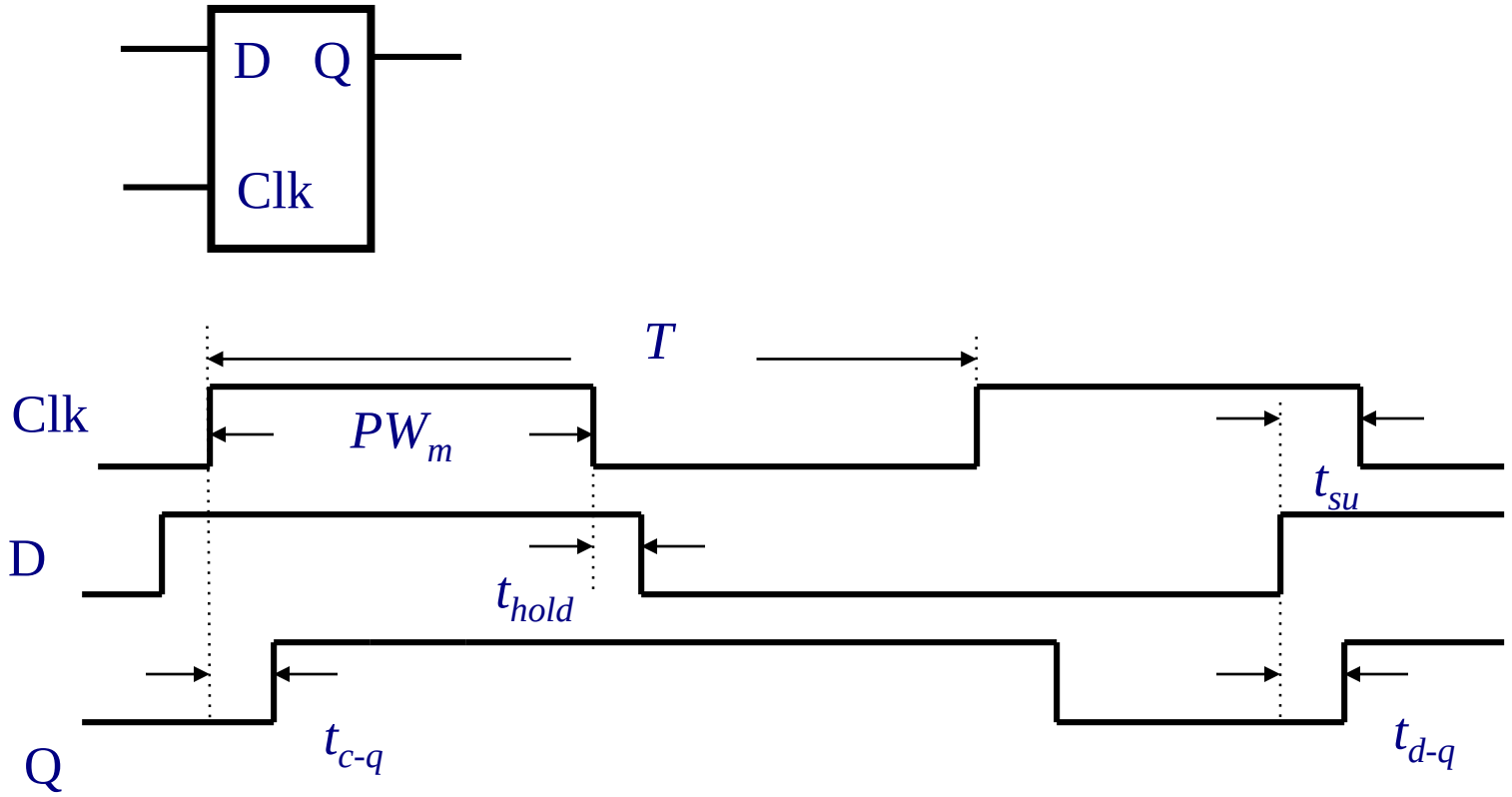
January 2003

Synchronous Timing



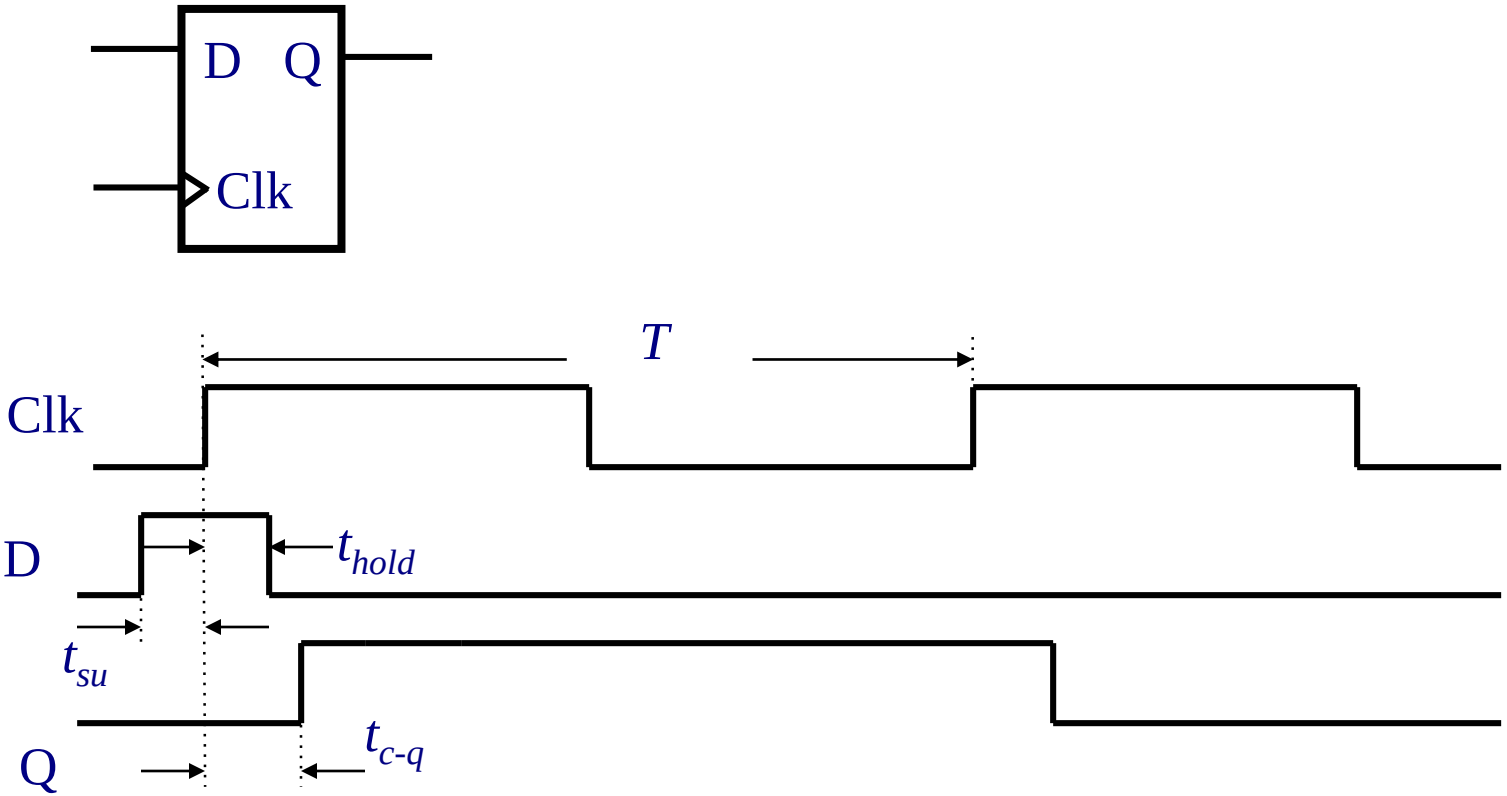
Timing Definitions

Latch Parameters



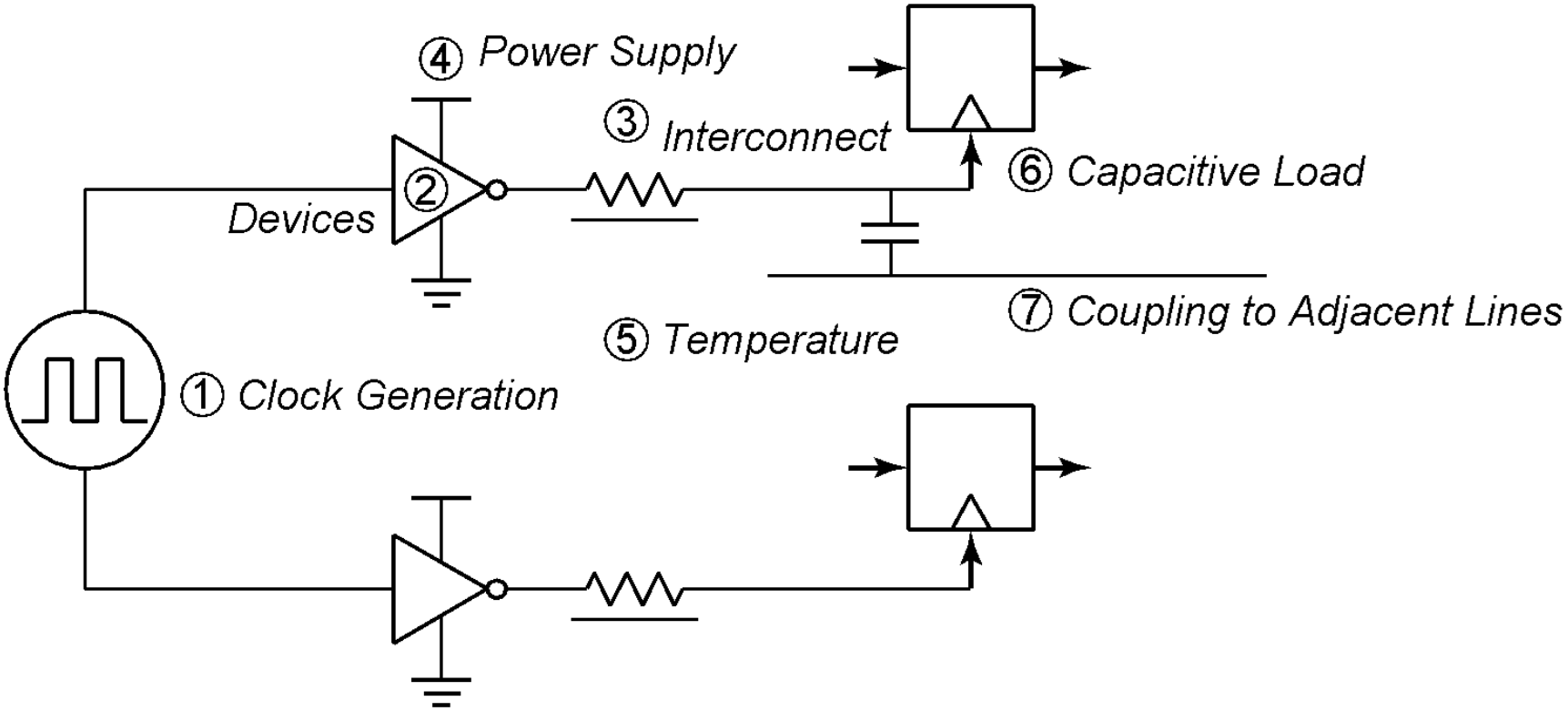
Delays can be different for rising and falling data transitions

Register Parameters



Delays can be different for rising and falling data transitions

Clock Uncertainties

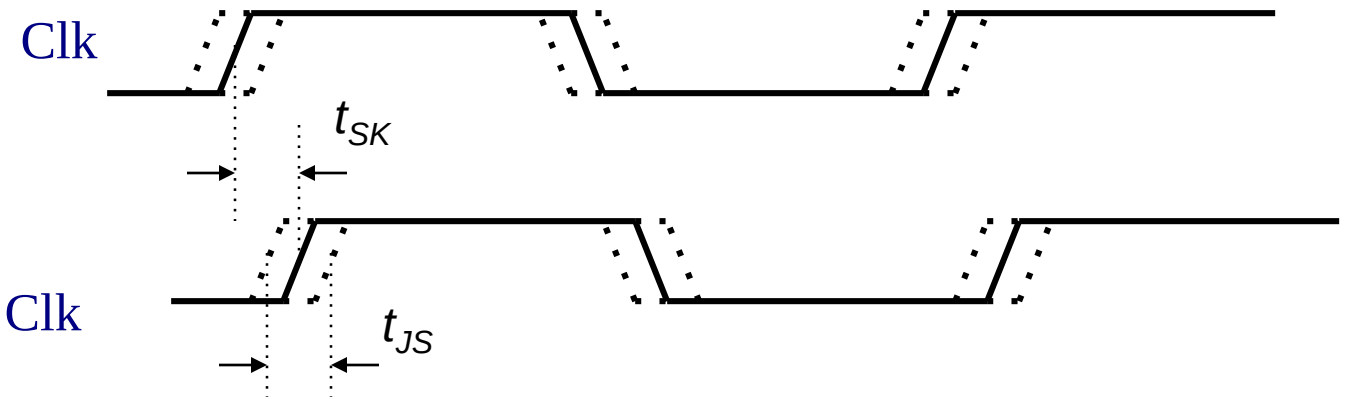


Sources of clock uncertainty

Clock Nonidealities

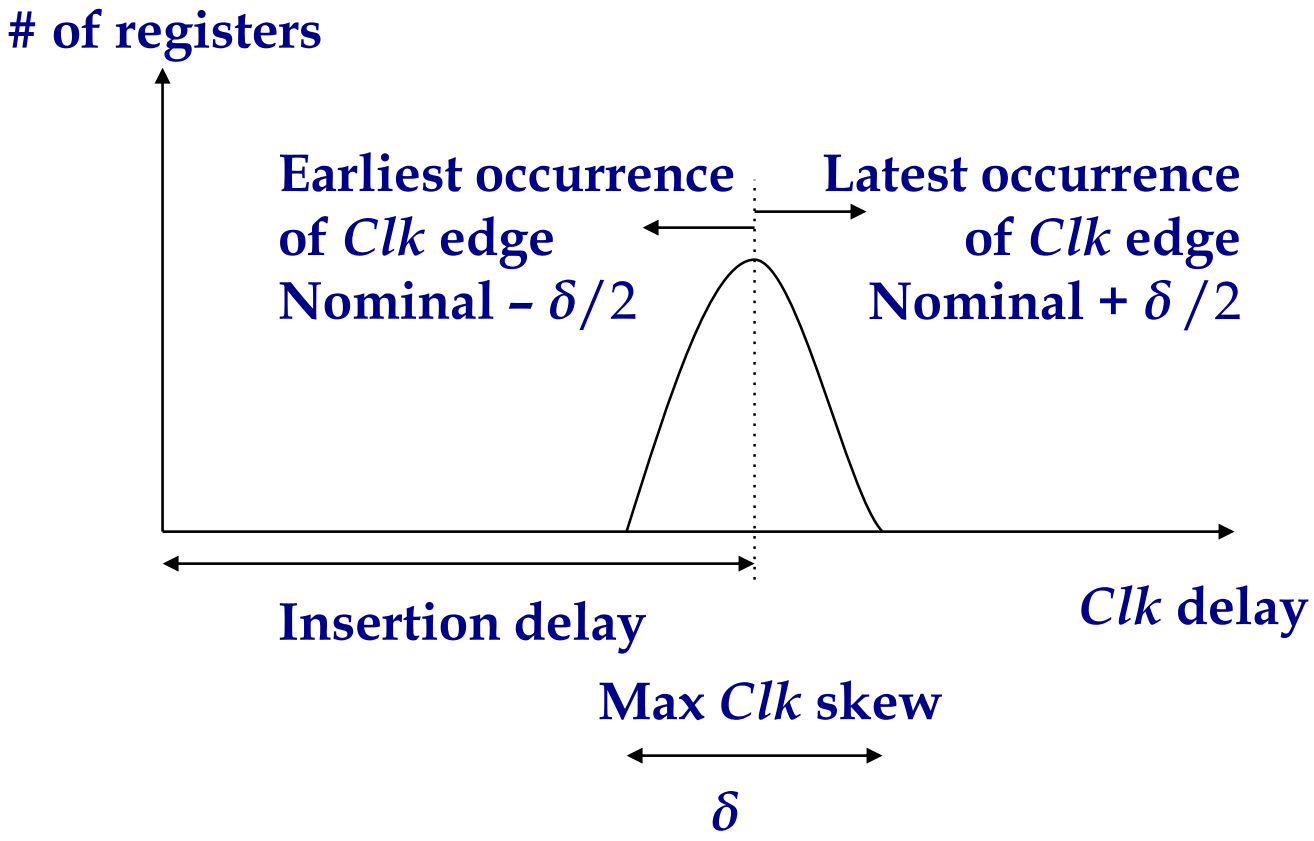
- **Clock skew**
  - Spatial variation in temporally equivalent clock edges; deterministic + random, t_{SK}
- **Clock jitter**
  - Temporal variations in consecutive edges of the clock signal; modulation + random noise
  - Cycle-to-cycle (short-term) t_{JS}
  - Long term t_{JL}
- **Variation of the pulse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

Clock Skew and Jitter

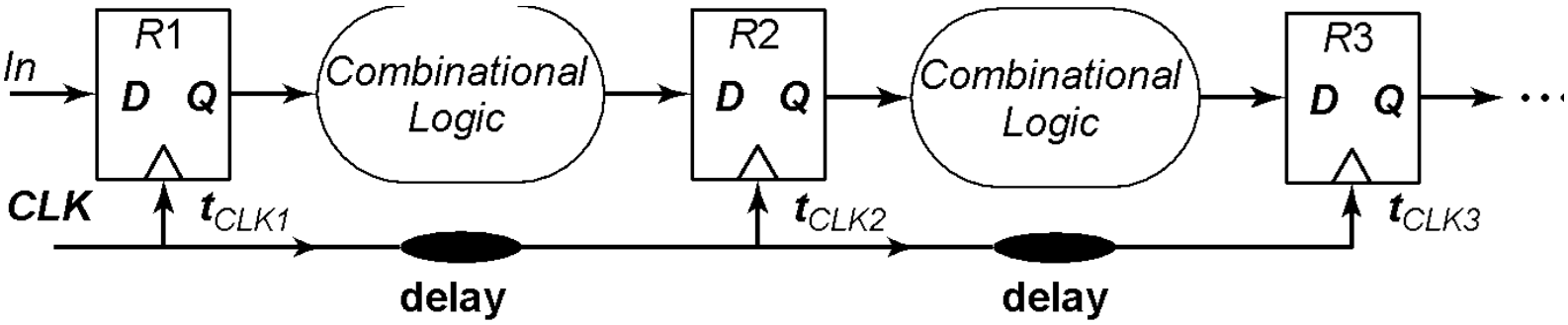


- Both skew and jitter affect the effective cycle time
- Only skew affects the race margin

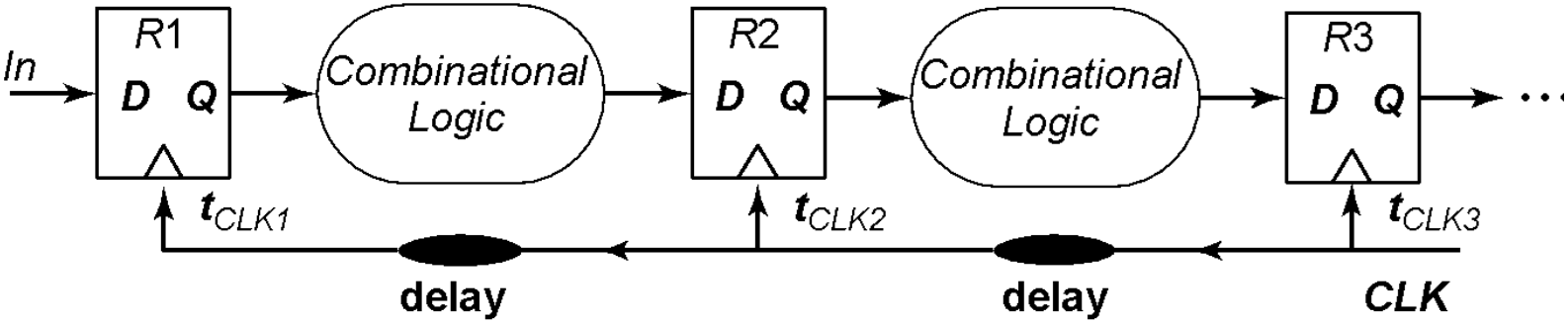
Clock Skew


# 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

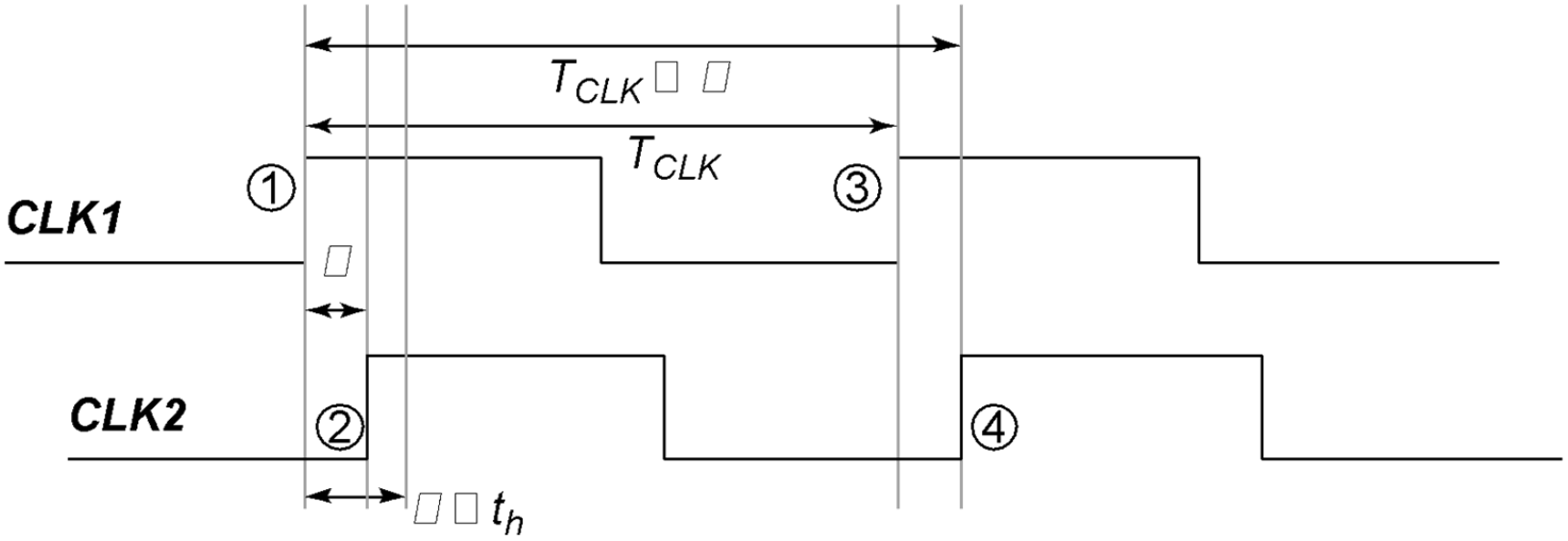


(a) Positive skew



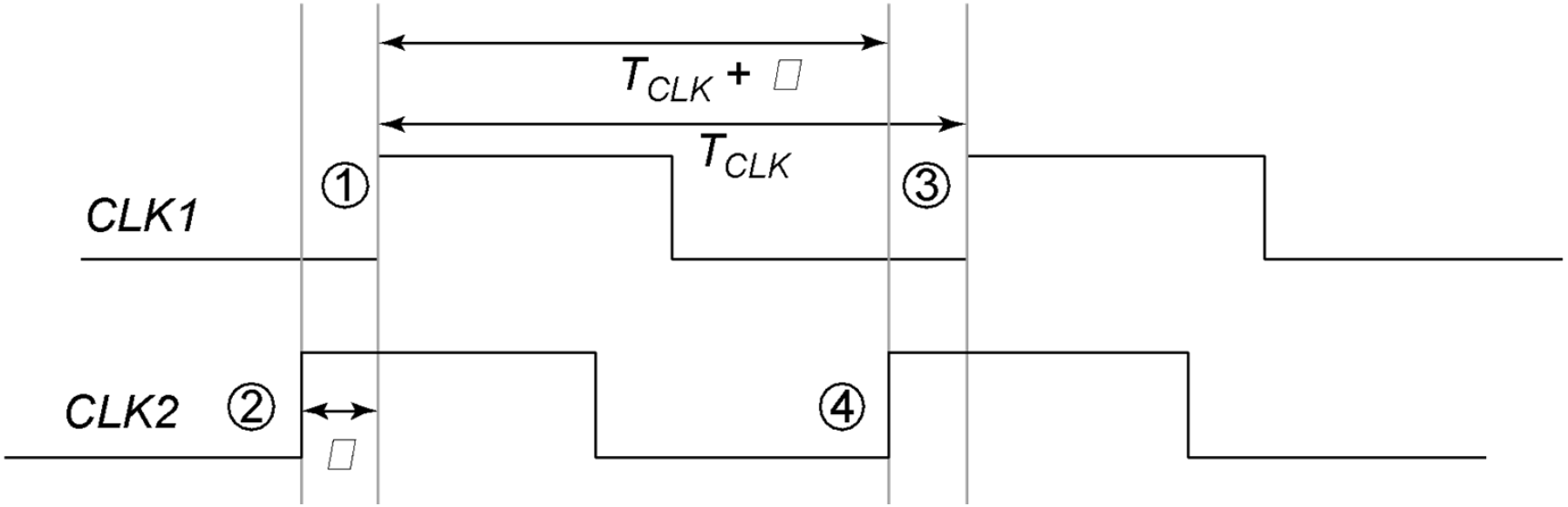
(b) Negative skew

Positive Skew



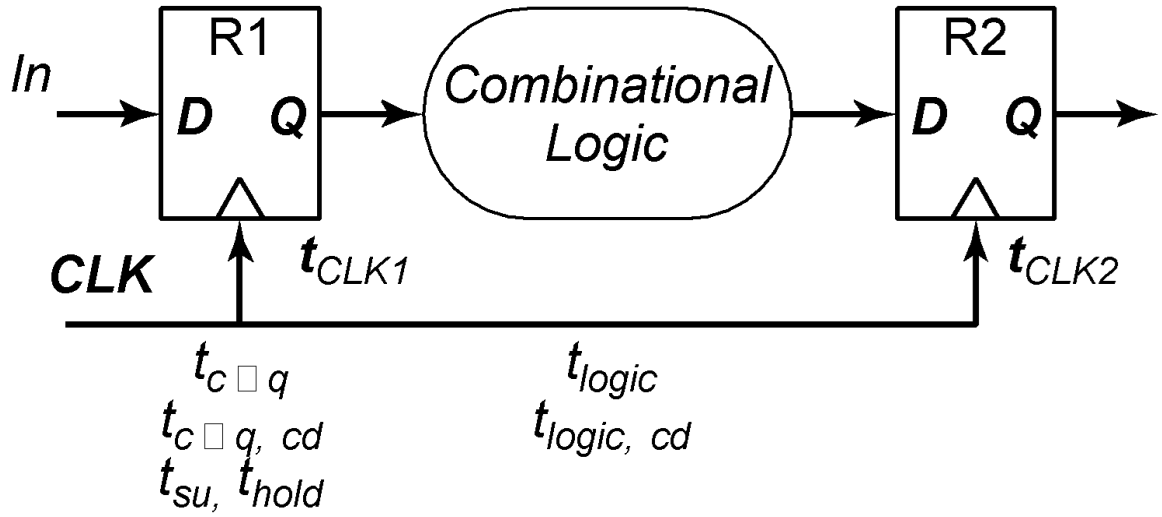
Launching edge arrives before the receiving edge

Negative Skew



Receiving edge arrives before the launching edge

Timing Constraints

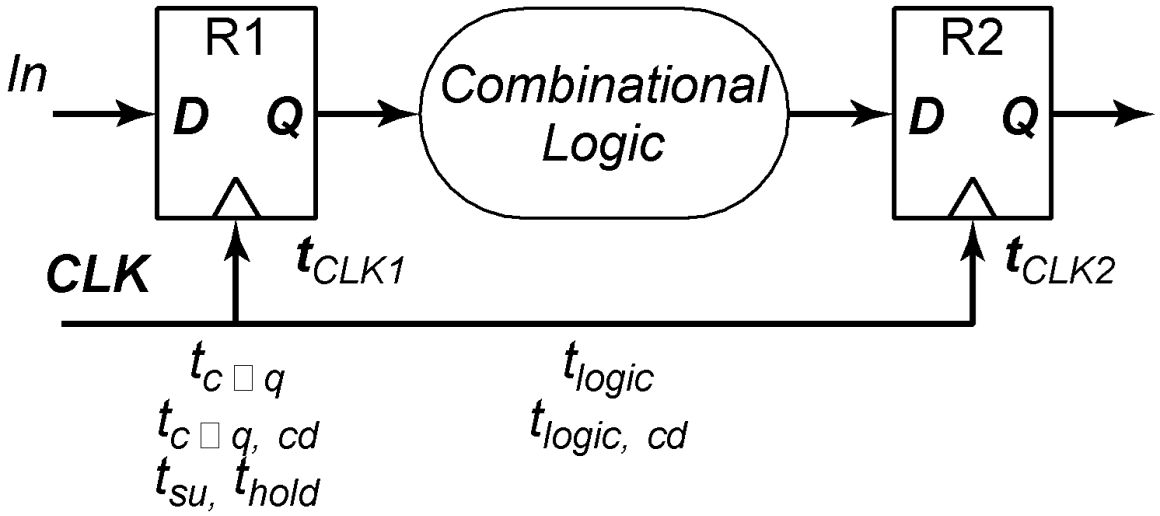


Minimum cycle time:

$T - \delta = t_{c\text{-}q} + t_{su} + t_{logic}$

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

Timing Constraints

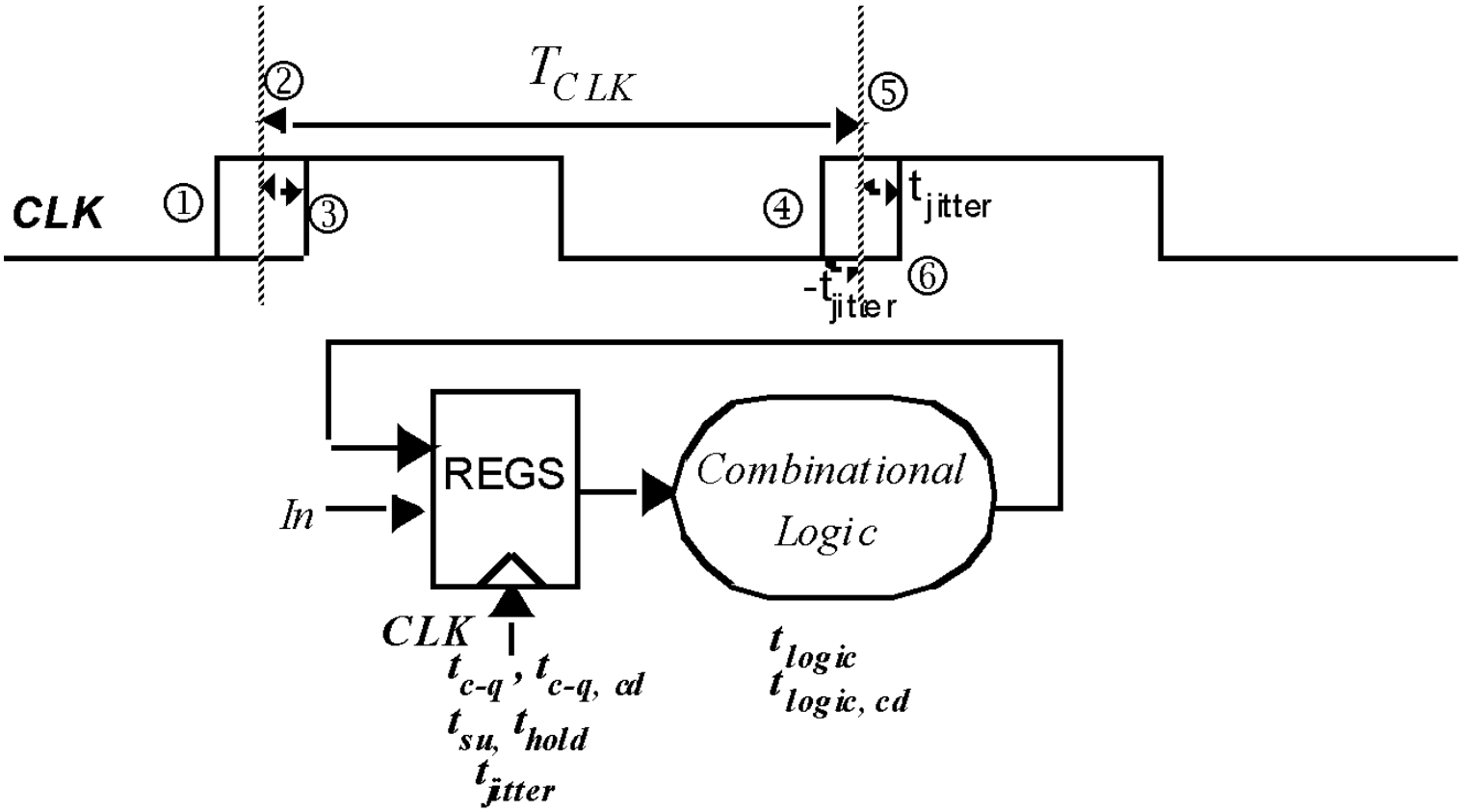


Hold time constraint:

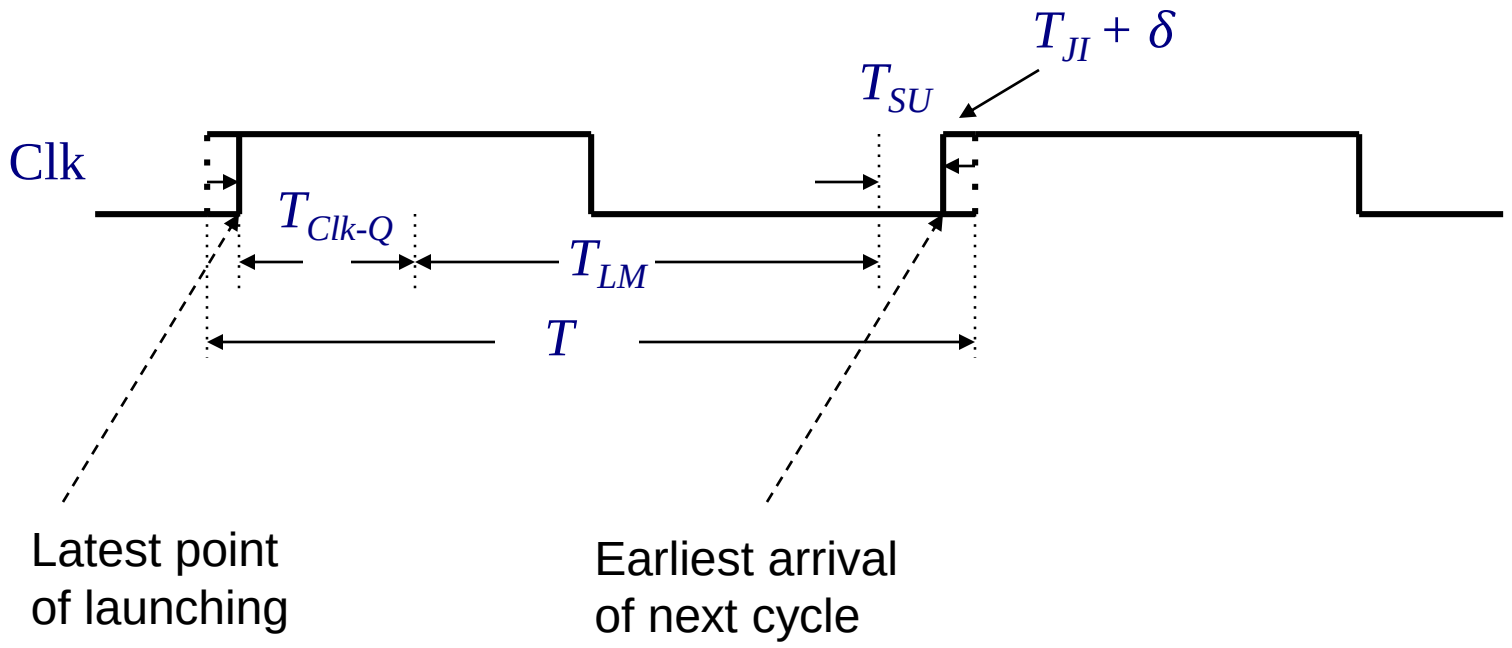
$$t_{(c\text{-}q,\ cd)} + t_{(logic,\ cd)} > t_{hold} + \delta$$

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

Impact of Jitter



Longest Logic Path in Edge-Triggered Systems



Clock Constraints in Edge-Triggered Systems

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

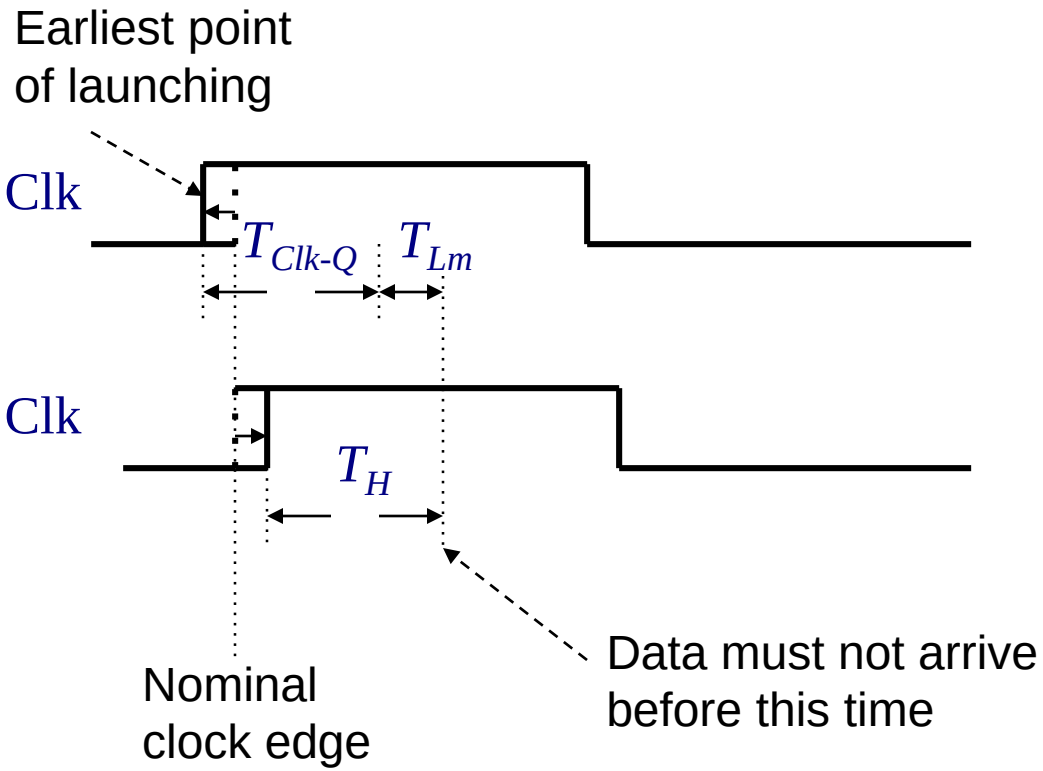
$$T_{c\text{-}q} + T_{LM} + T_{SU} < T - T_{JI,1} - T_{JI,2} - \delta$$

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

$$T_{c\text{-}q} + T_{LM} + T_{SU} + \delta + 2\,T_{JI} < T$$

Skew can be either positive or negative

Shortest Path



Clock Constraints in Edge-Triggered Systems

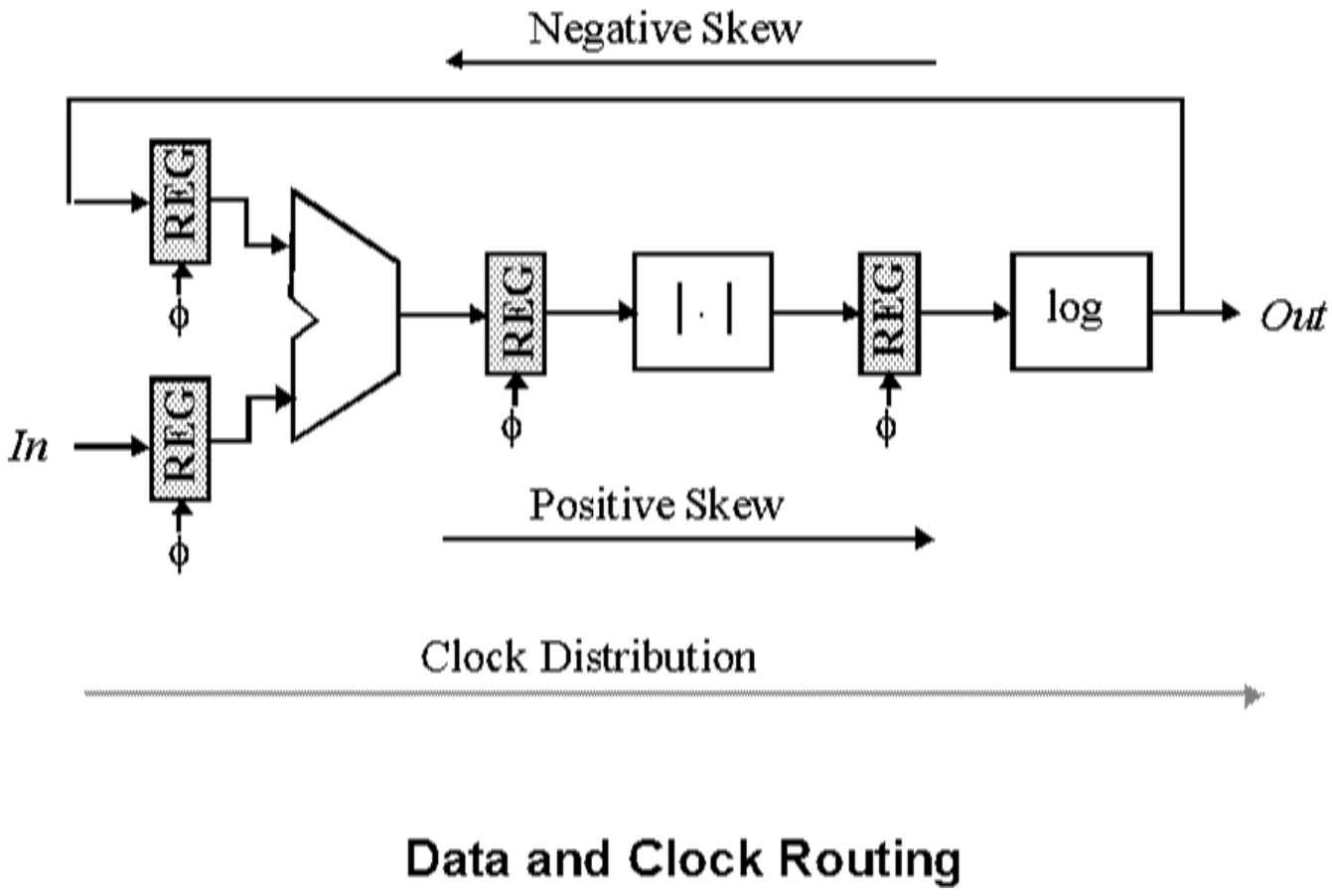
If launching edge is early and receiving edge is late:

$$T_{c\text{-}q} + T_{LM} - T_{JI,1} < T_H + T_{JI,2} + \delta$$

Minimum logic delay

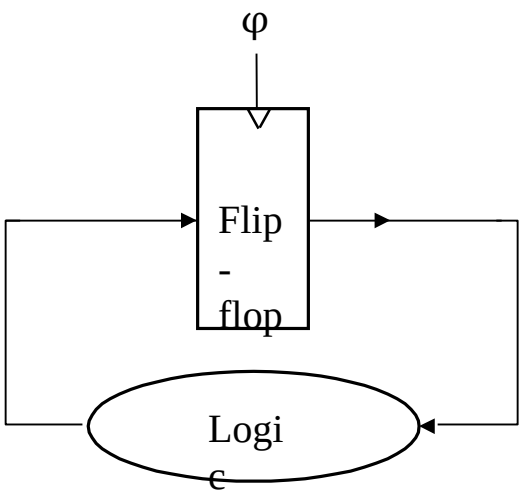
$$T_{c\text{-}q} + T_{LM} < T_H + 2T_{JI} + \delta$$

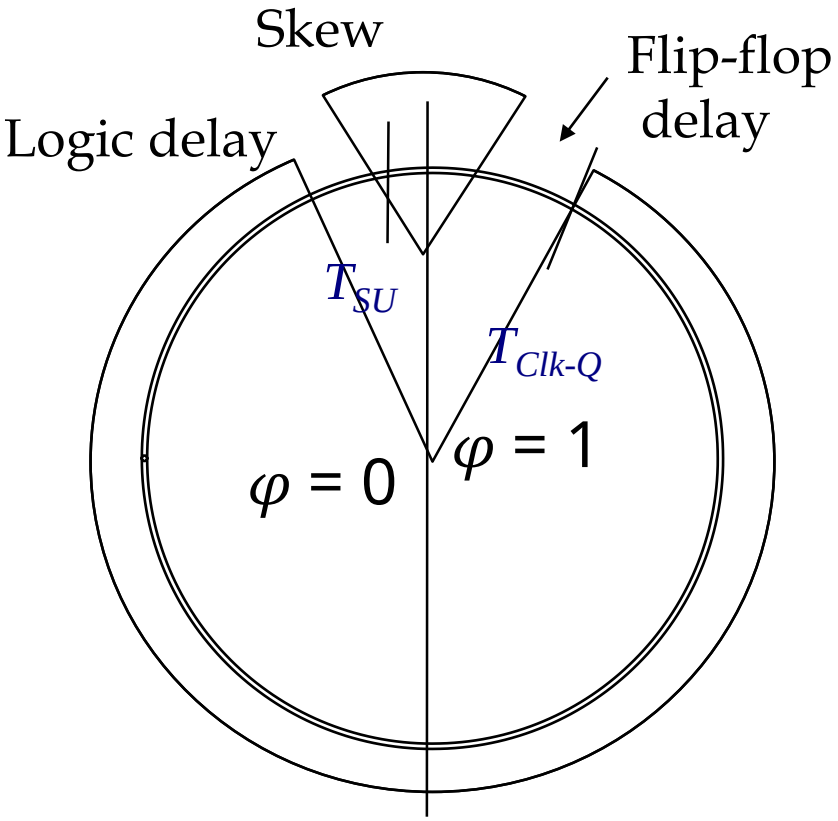
How to counter Clock Skew?



Data and Clock Routing

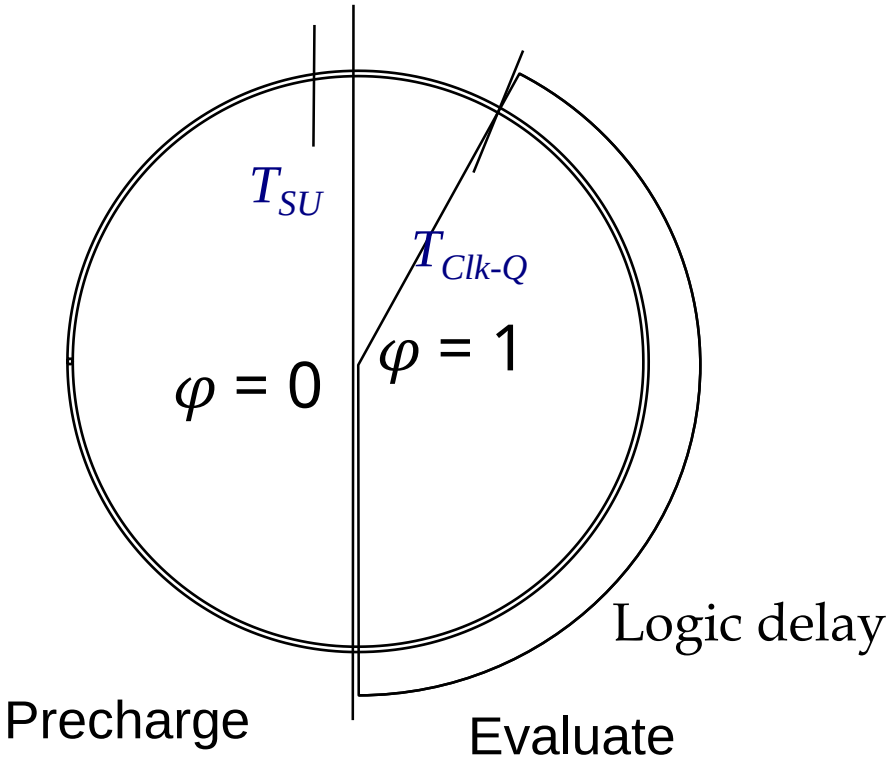
Flip-Flop – Based Timing

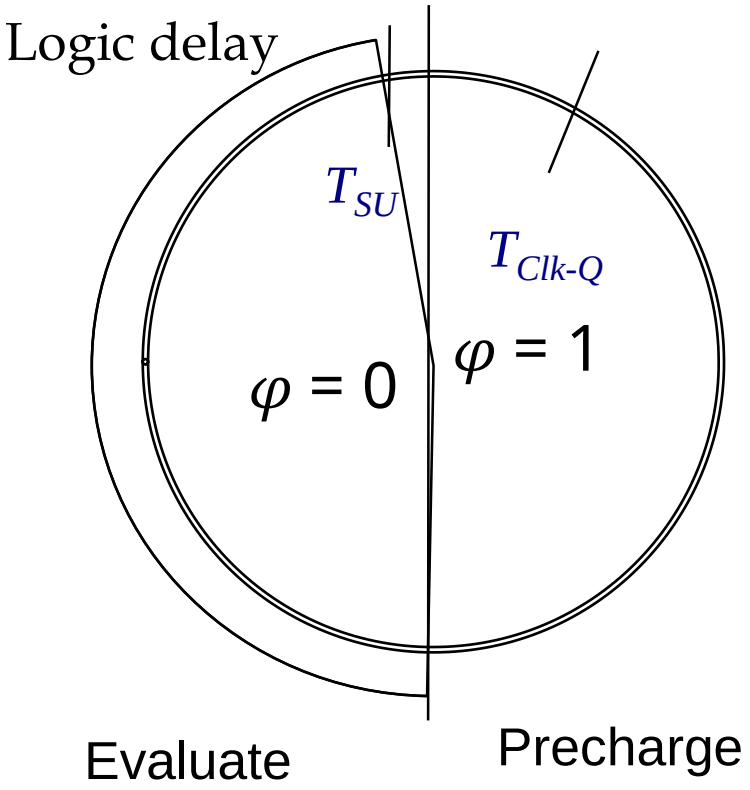




Representation after
M. Horowitz, VLSI Circuits 1996.

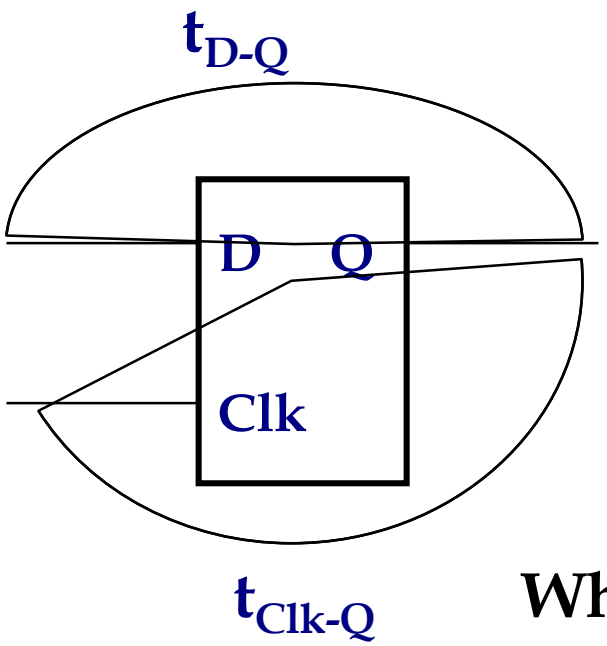
Flip-Flops and Dynamic Logic





Flip-flops are used only with static logic

Latch timing



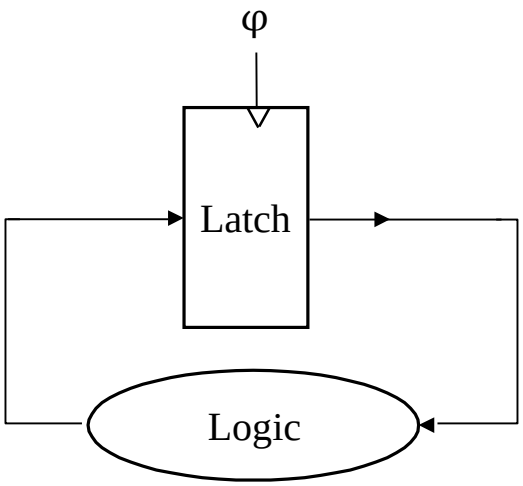
When data arrives to transparent latch

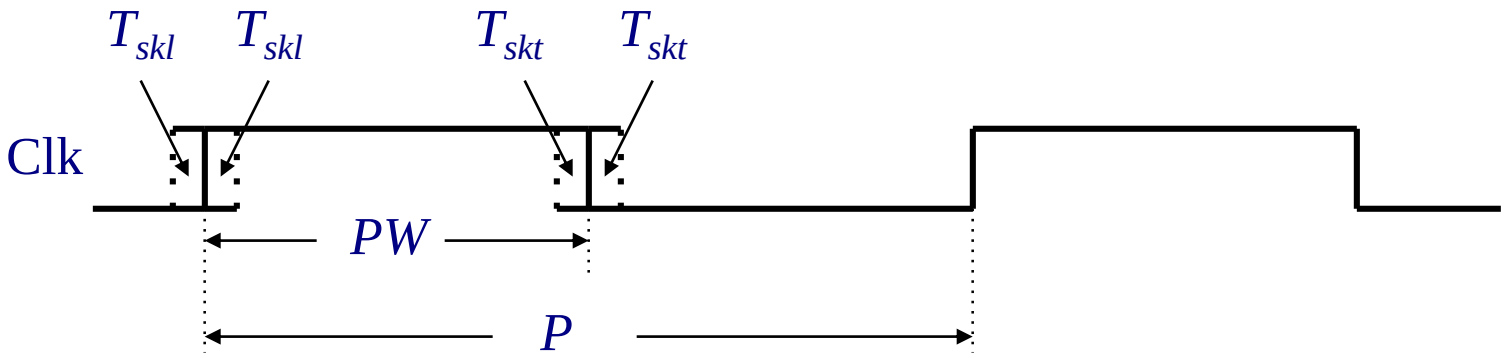
Latch is a 'soft' barrier

When data arrives to closed latch

Data has to be 're-launched'

Single-Phase Clock with Latches

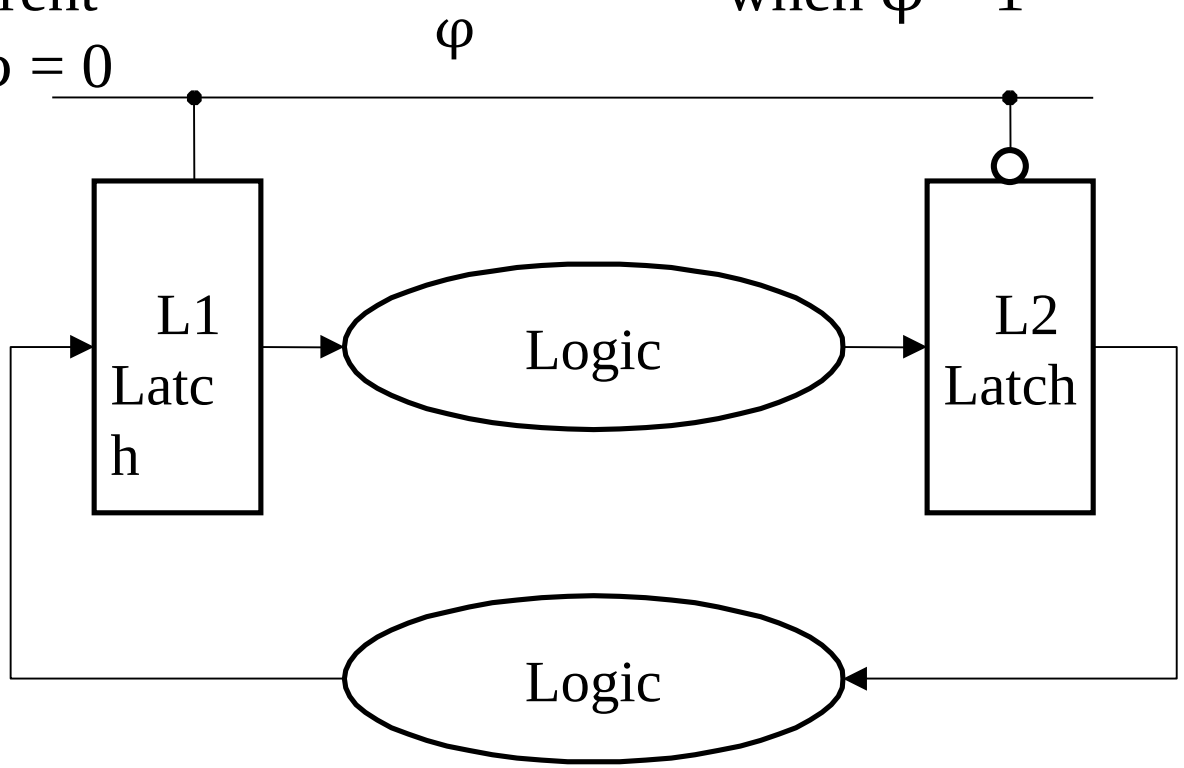




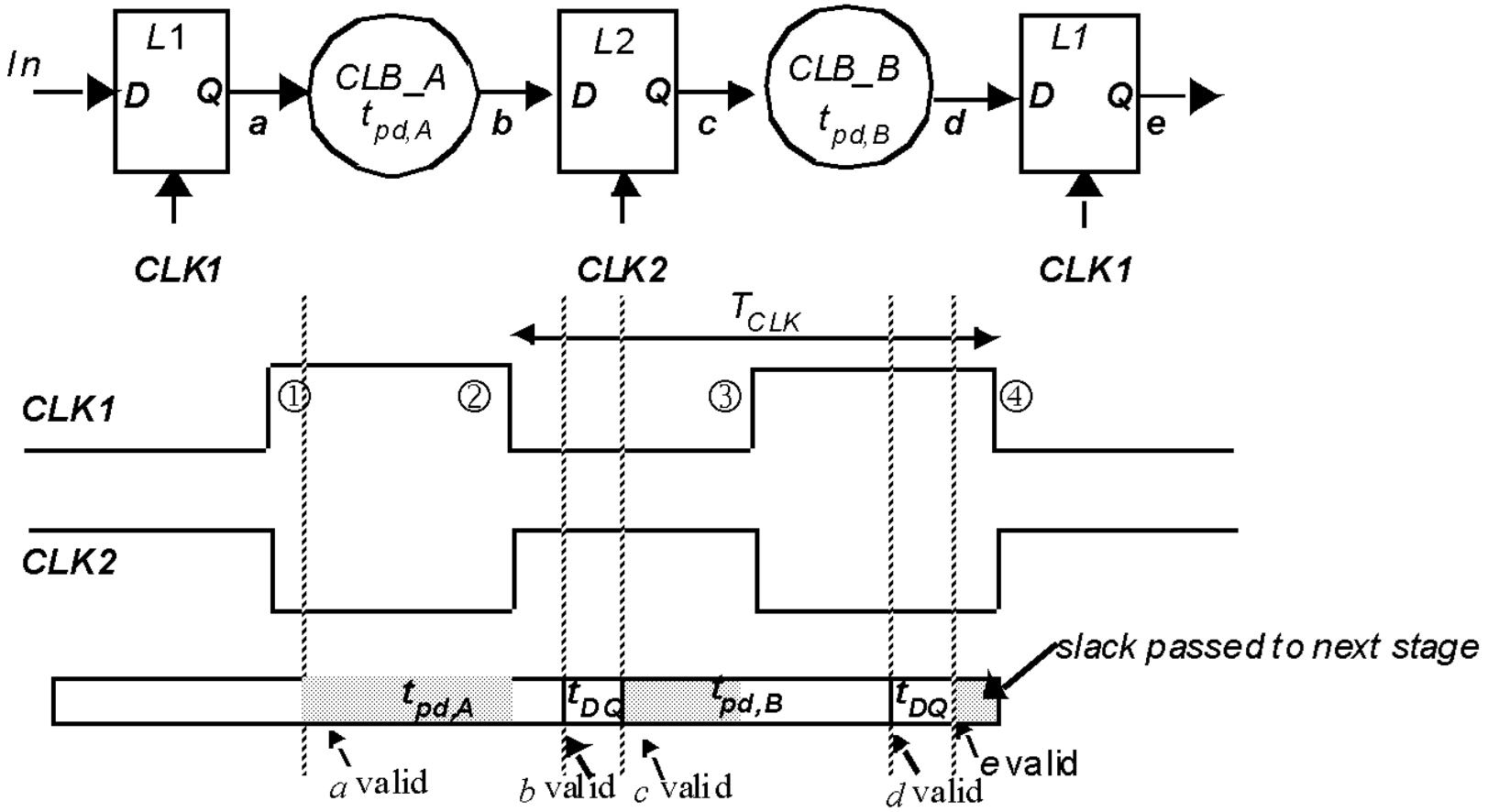
Latch-Based Design

L1 latch is transparent when φ = 0

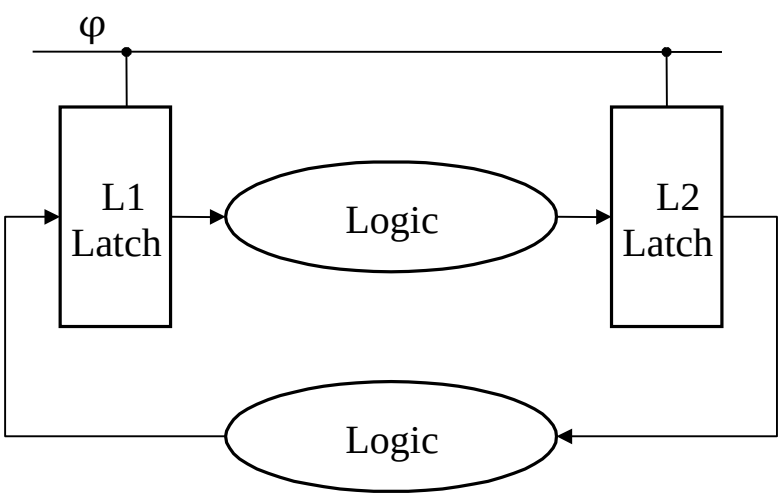
L2 latch is transparent when φ = 1

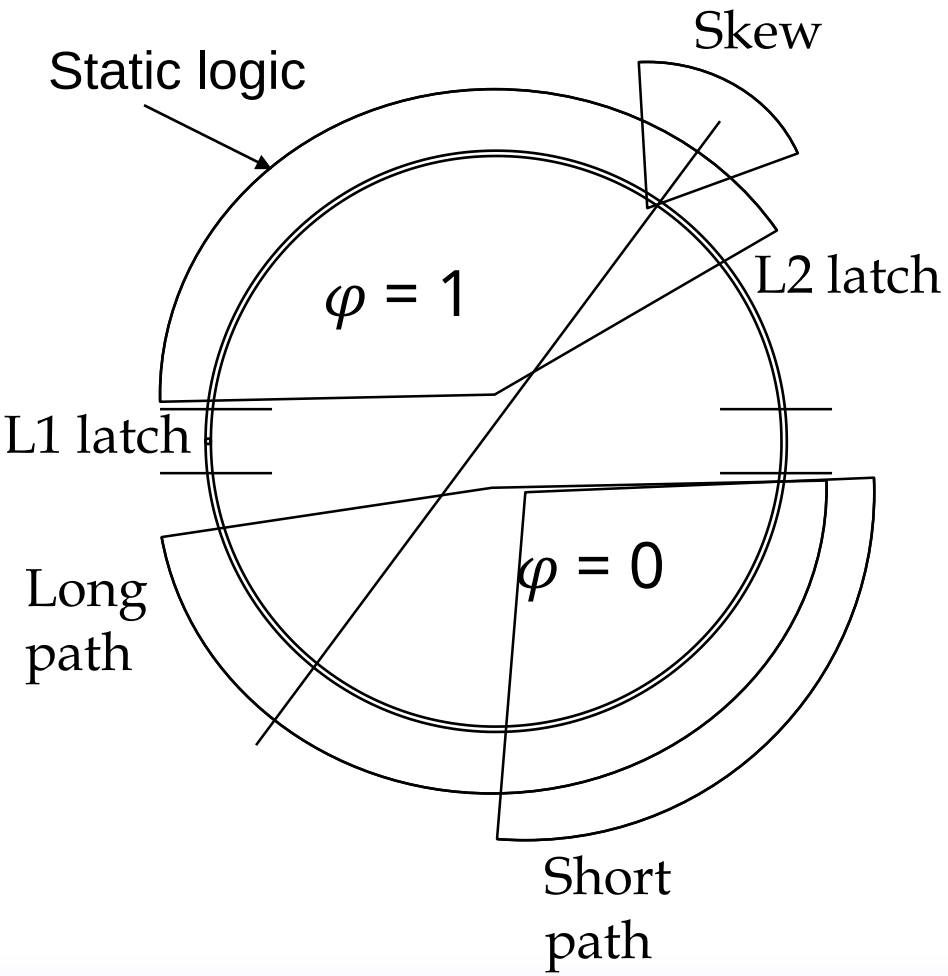


Slack-borrowing



Latch-Based Timing

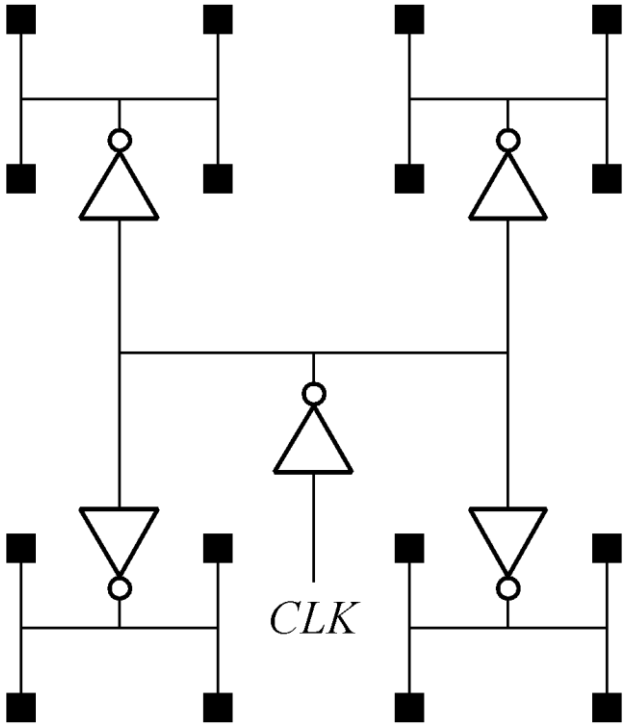




Can tolerate skew!

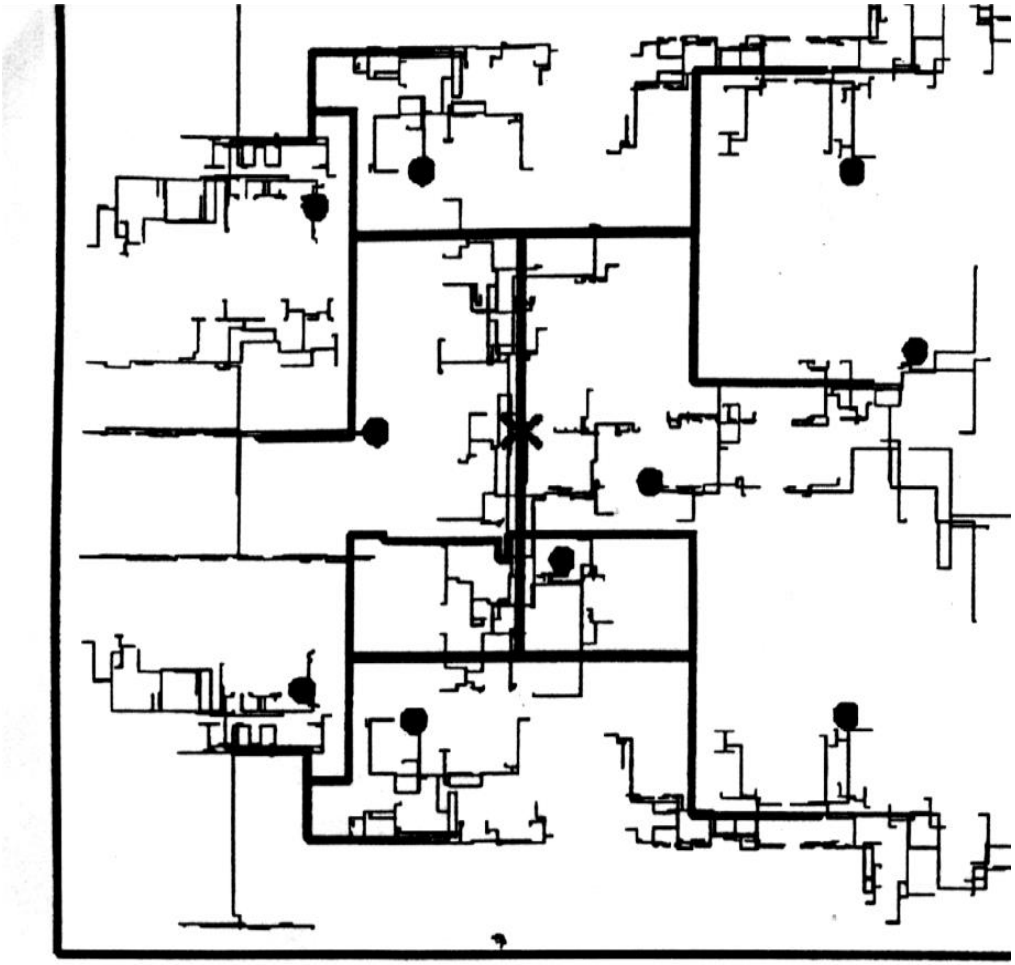
Clock Distribution

H-tree



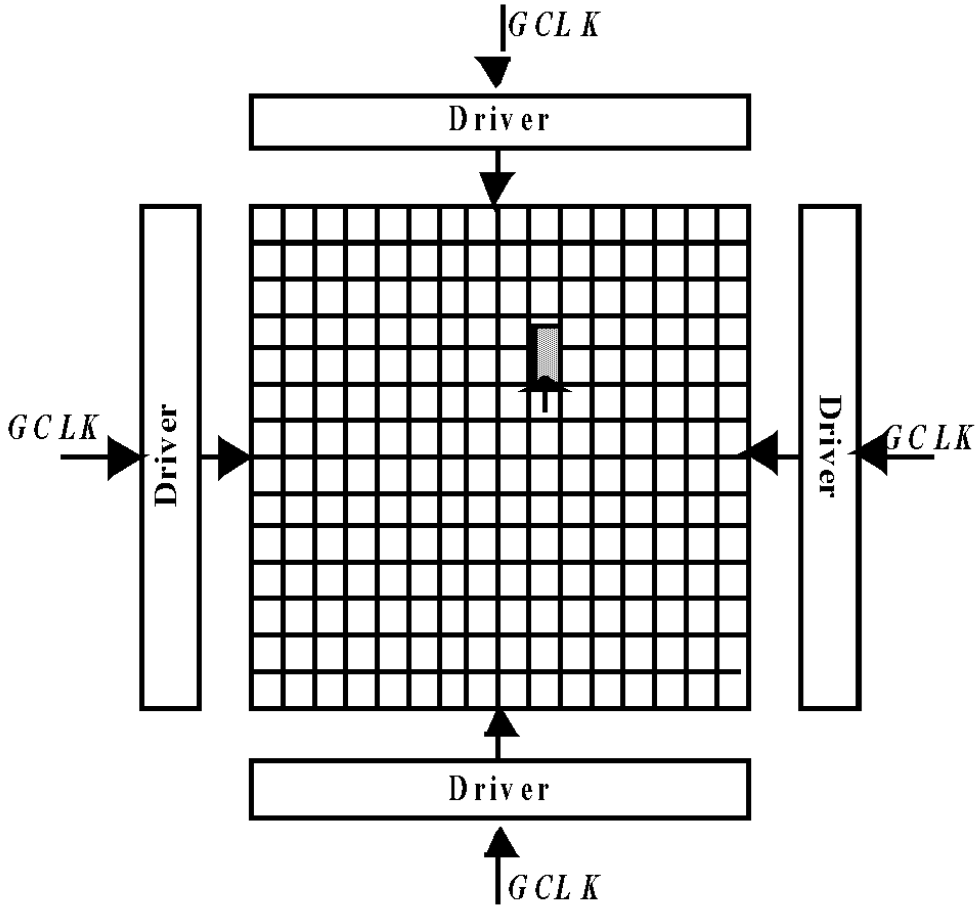
Clock is distributed in a tree-like fashion

More realistic H-tree



[Restle98]

The Grid System



- *No rc-matching*
- *Large power*

Example: DEC Alpha 21164

Clock Frequency: 300 MHz - 9.3 Million Transistors

Total Clock Load: 3.75 nF

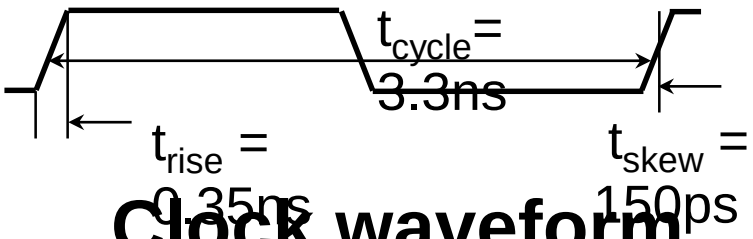
Power in Clock Distribution network : 20 W (out of 50)

Uses Two Level Clock Distribution:

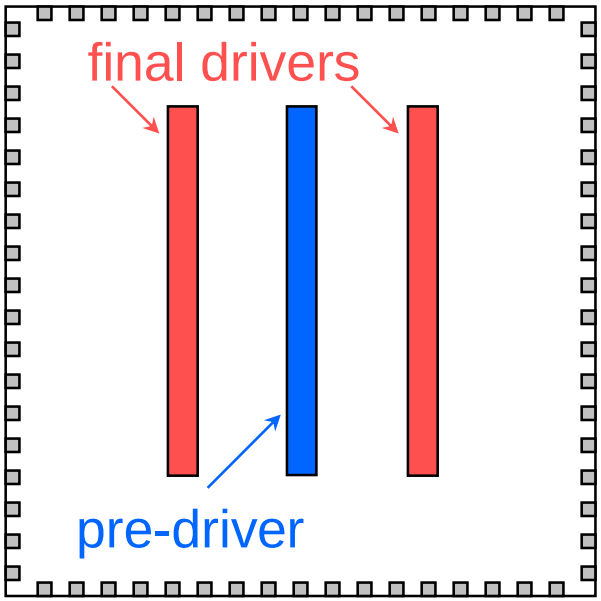
- Single 6-stage driver at center of chip
- Secondary buffers drive left and right side clock grid in Metal3 and Metal4

Total driver size: 58 cm!

21164 Clocking

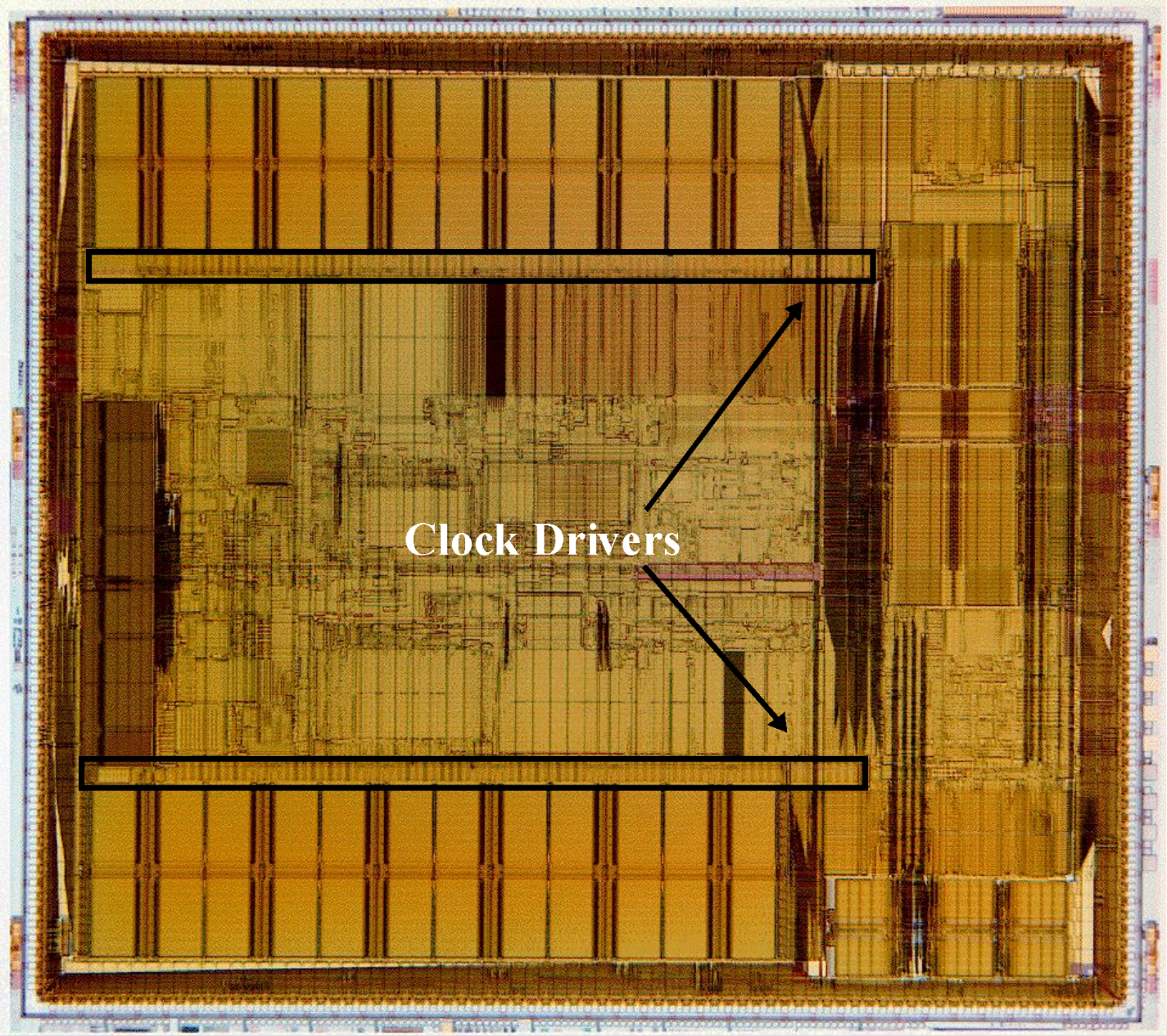


Clock waveform

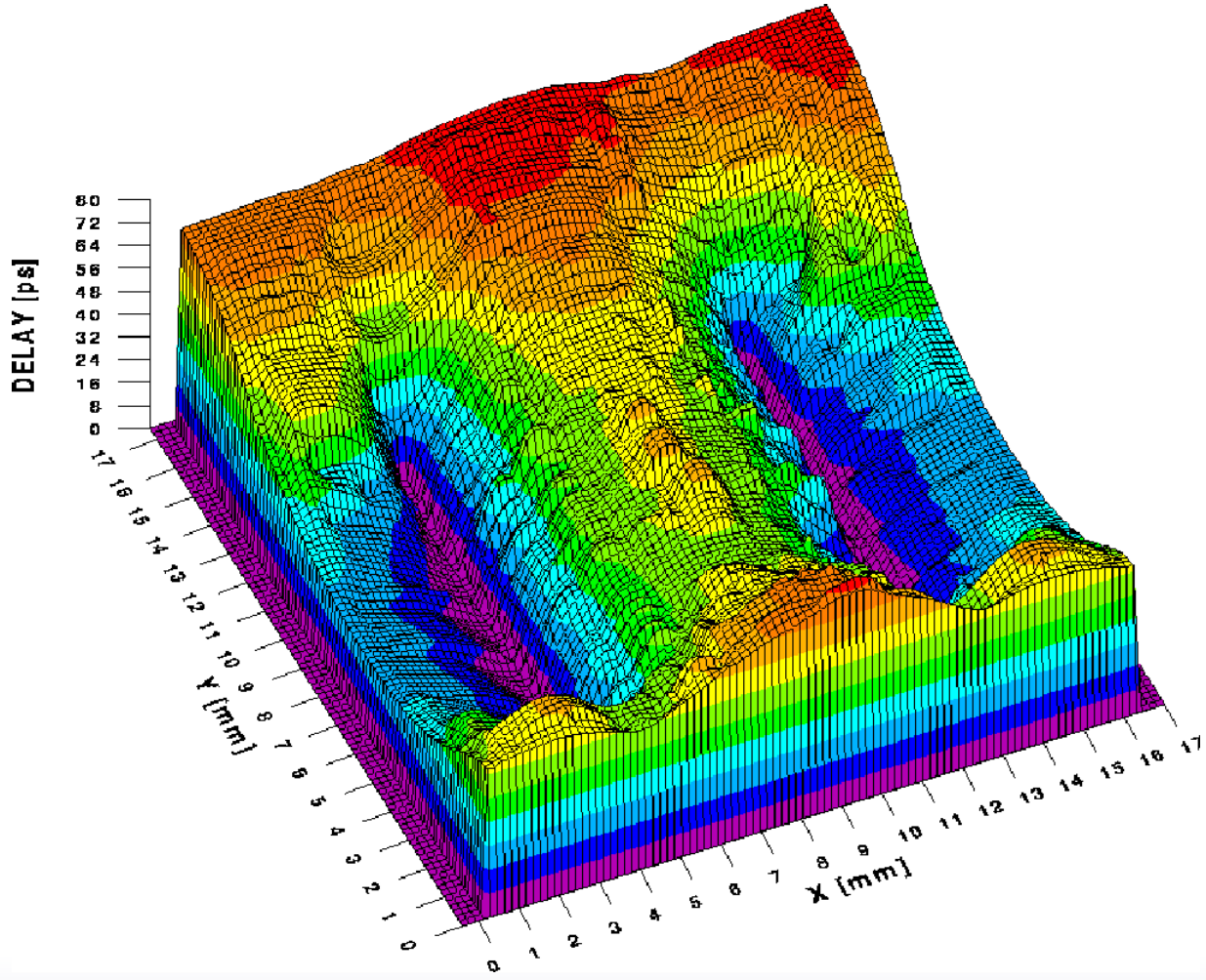


Location of clock driver on die

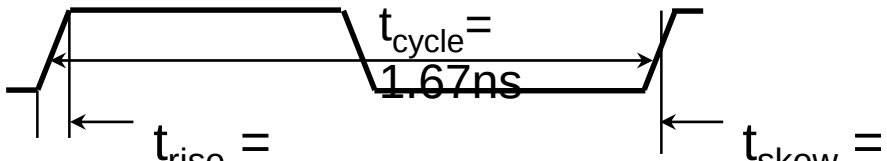
- 2 phase single wire clock, distributed globally
- 2 distributed driver channels
  - Reduced RC delay/skew
  - Improved thermal distribution
  - 3.75nF clock load
  - 58 cm final driver width
- Local inverters for latching
- Conditional clocks in caches to reduce power
- More complex race checking
-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


Clock Drivers

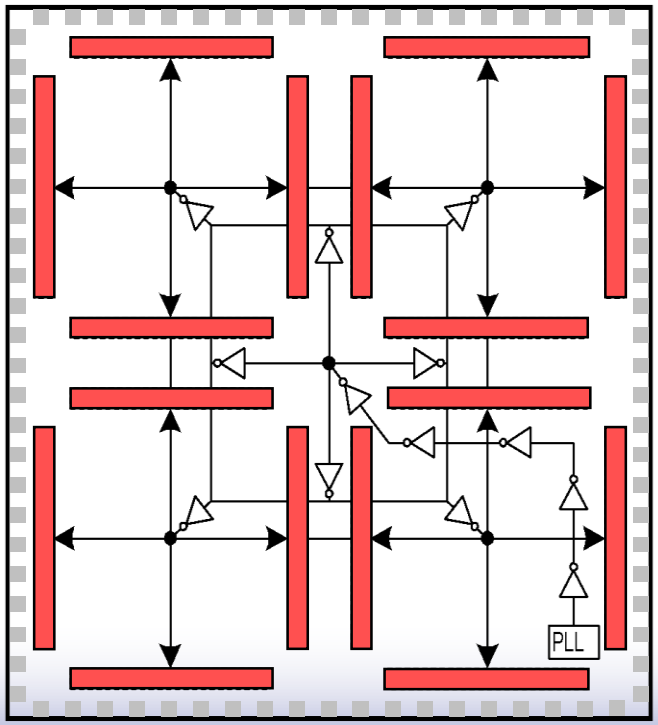
Clock Skew in Alpha Processor



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

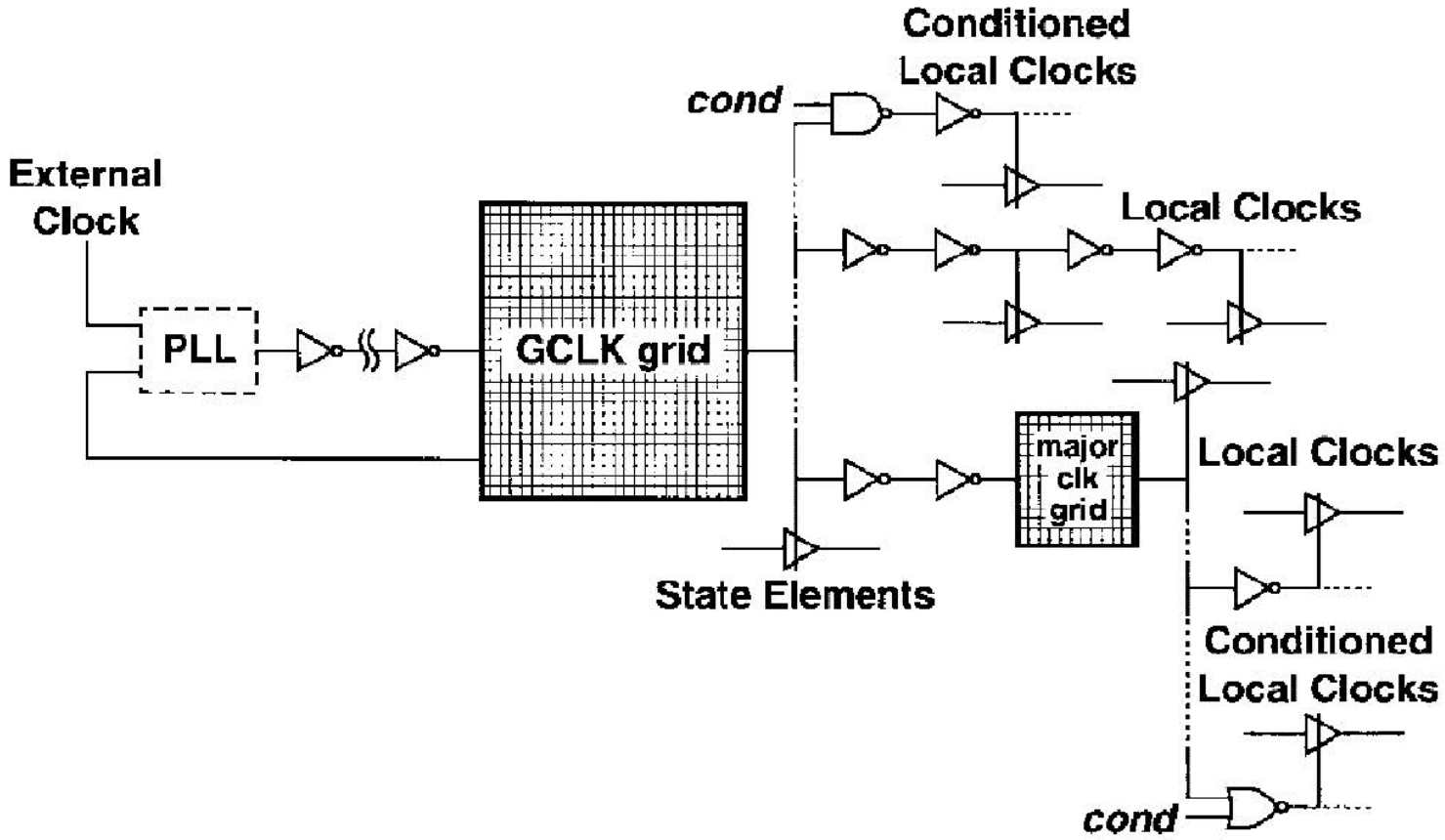


Global clock waveform

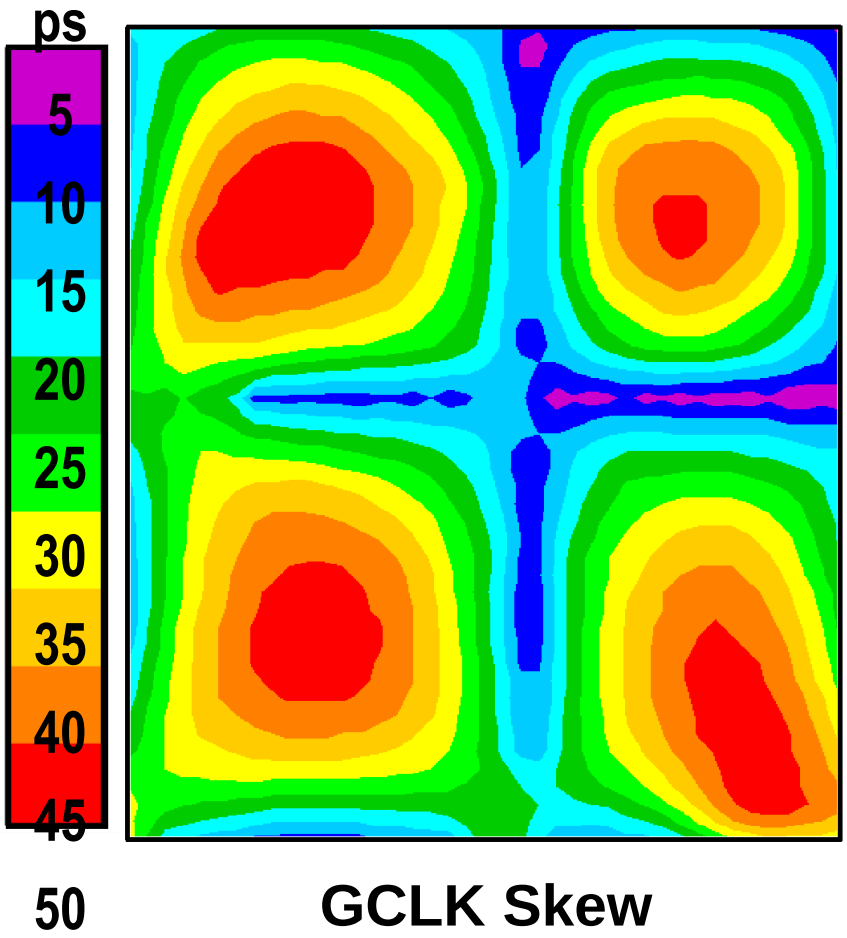


- 2 Phase, with multiple conditional buffered clocks
  - 2.8 nF clock load
  - 40 cm final driver width
- Local clocks can be gated “off” to save power
- Reduced load/skew
- Reduced thermal issues
- Multiple clocks complicate race checking

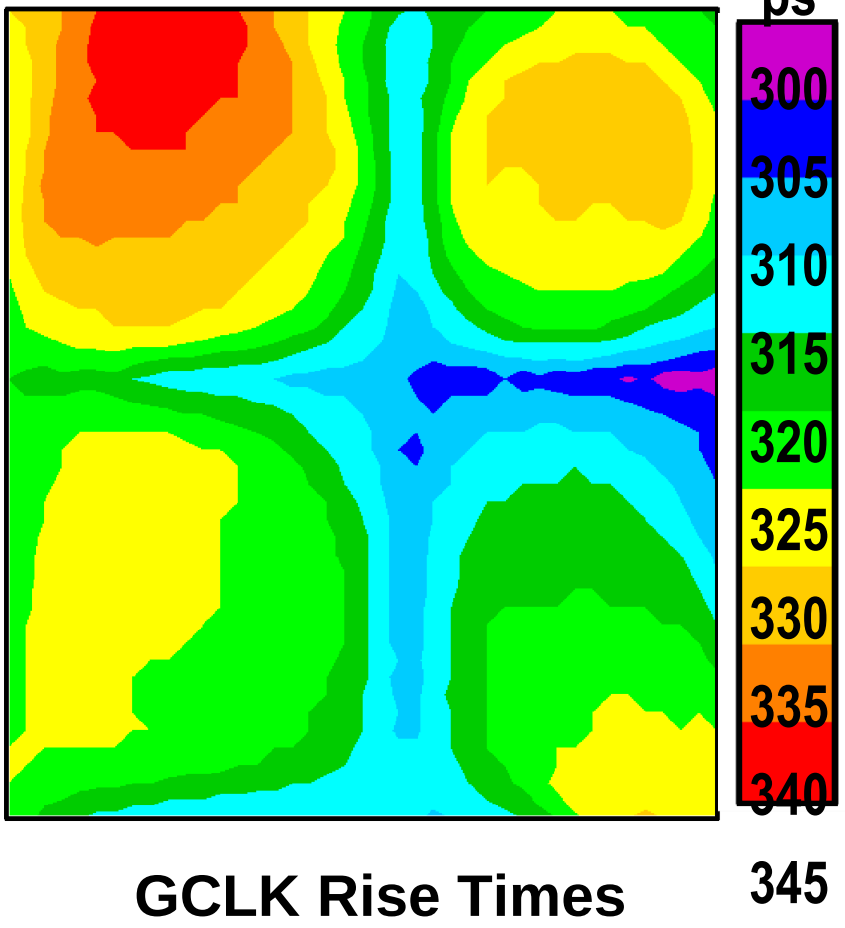
21264 Clocking



EV6 Clock Results



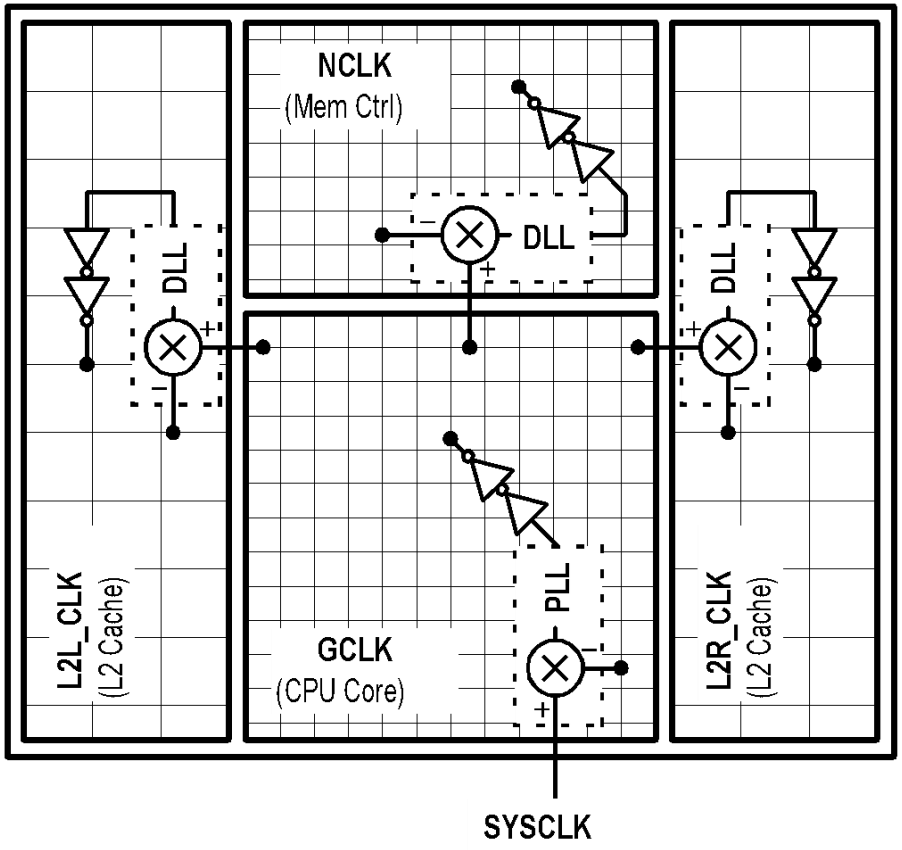
GCLK Skew
(at Vdd/2 Crossings)



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

Self-timed and Asynchronous Design

Functions of clock in synchronous design

1) Acts as completion signal
2) Ensures the correct ordering of events

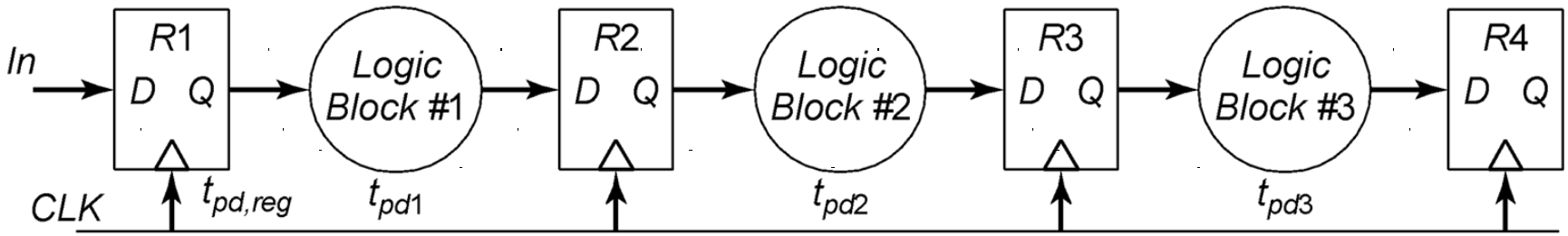
Truly asynchronous design

1) Completion is ensured by careful timing analysis
2) Ordering of events is implicit in logic

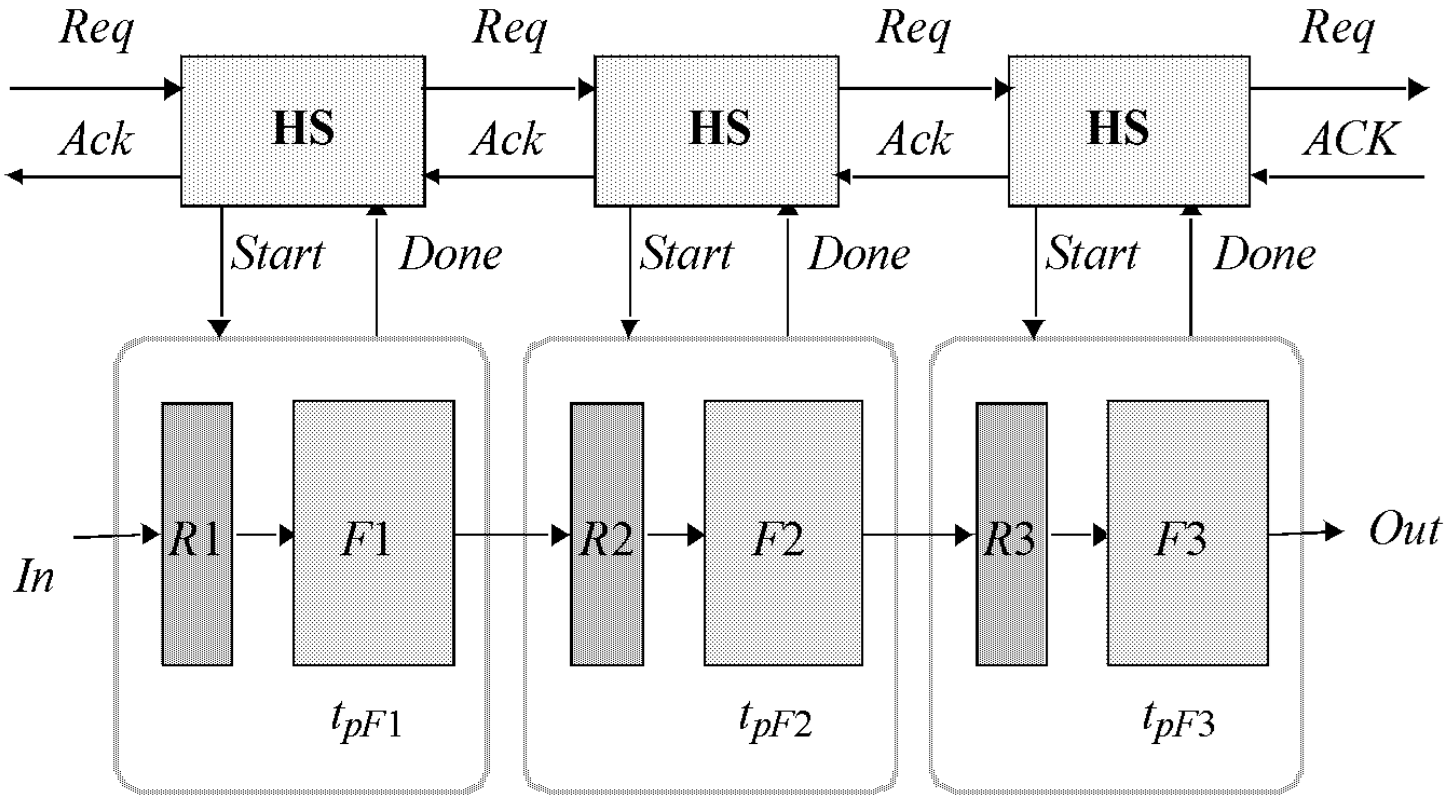
Self-timed design

1) Completion ensured by completion signal
2) Ordering imposed by handshaking protocol

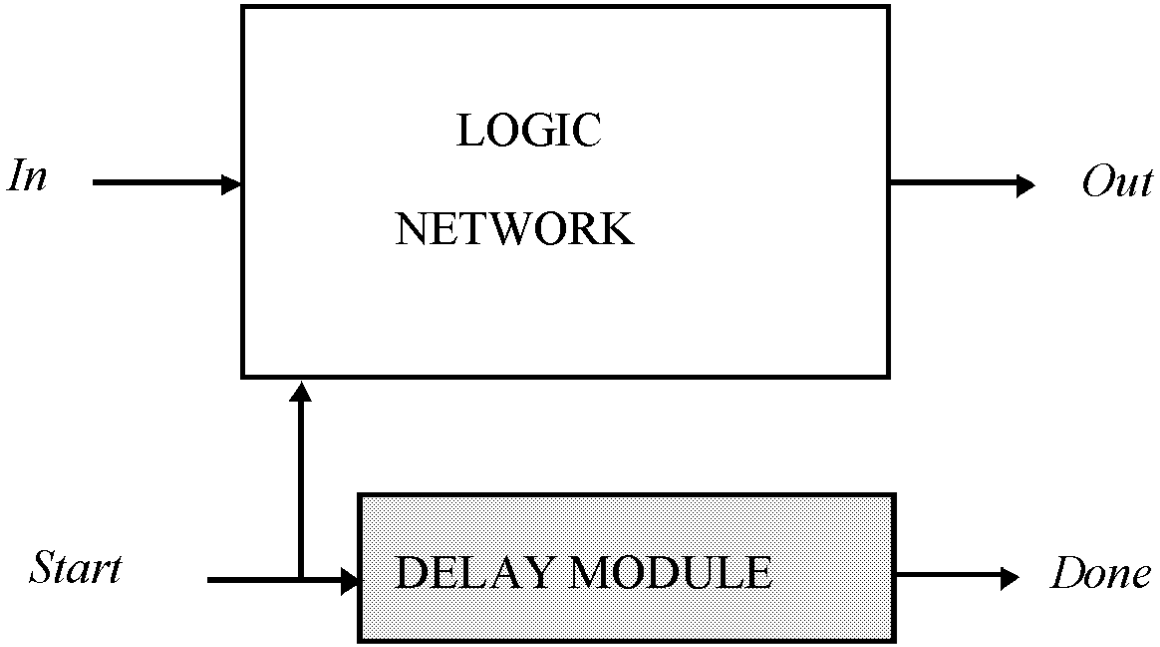
Synchronous Pipelined Datapath



Self-Timed Pipelined Datapath



Completion Signal Generation



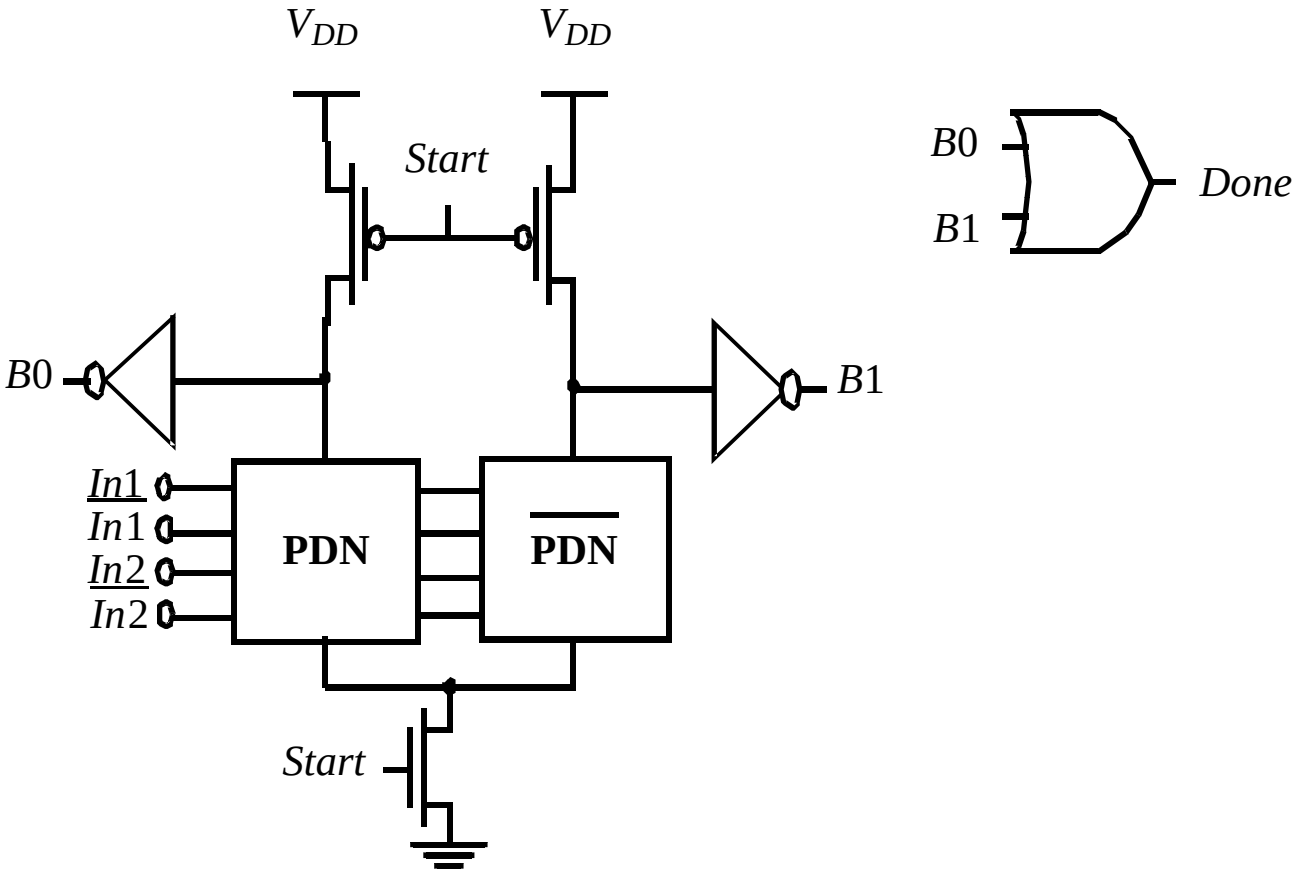
Using Delay Element (e.g. in memories)

Completion Signal Generation

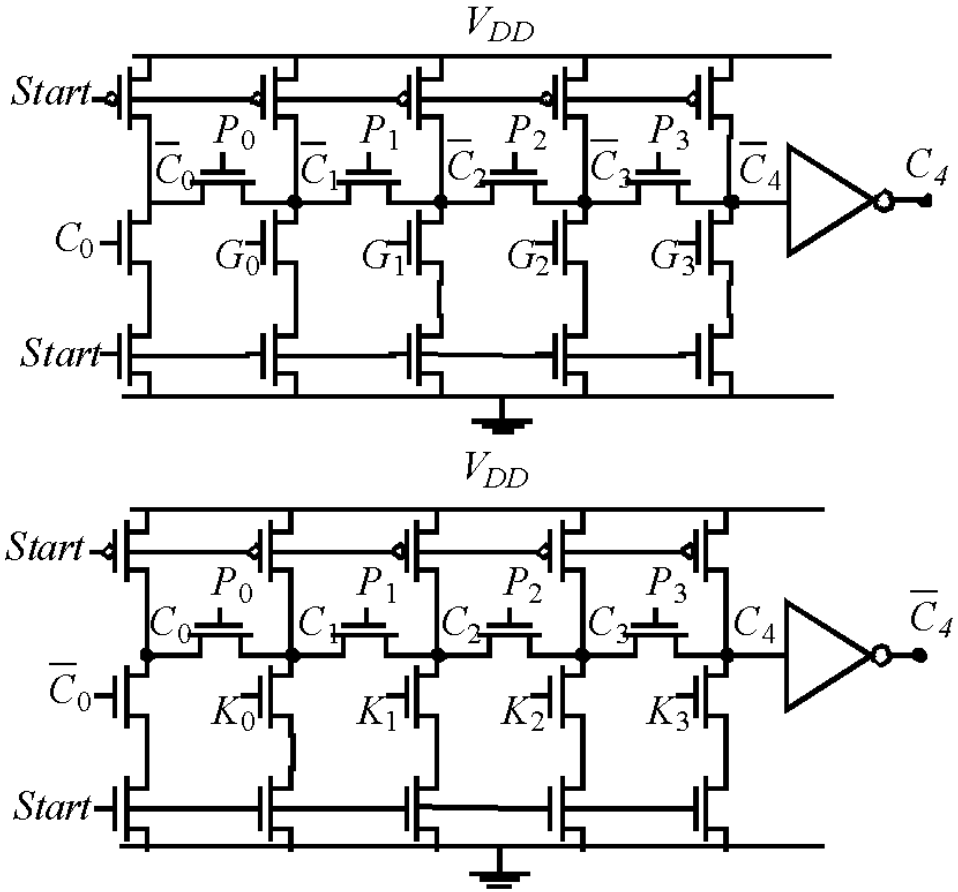
B	B0	B1
in transition (or reset)	0	0
0	0	1
1	1	0
illegal	1	1

Using Redundant Signal Encoding

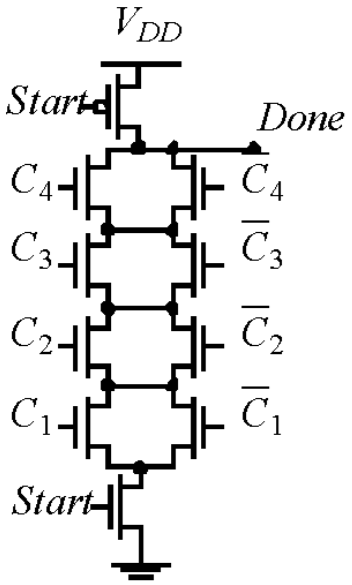
Completion Signal in DCVSL



Self-Timed Adder

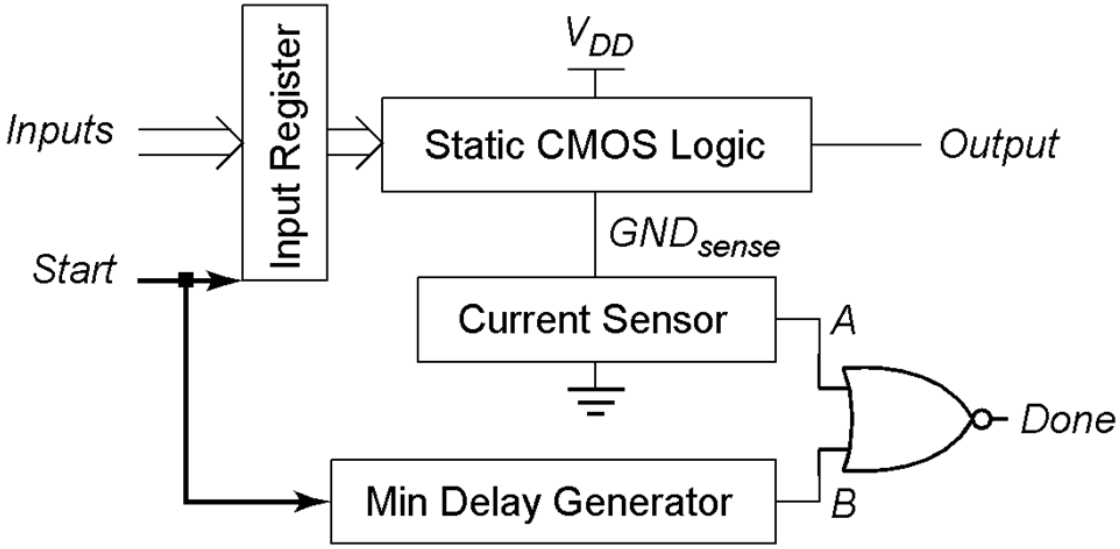


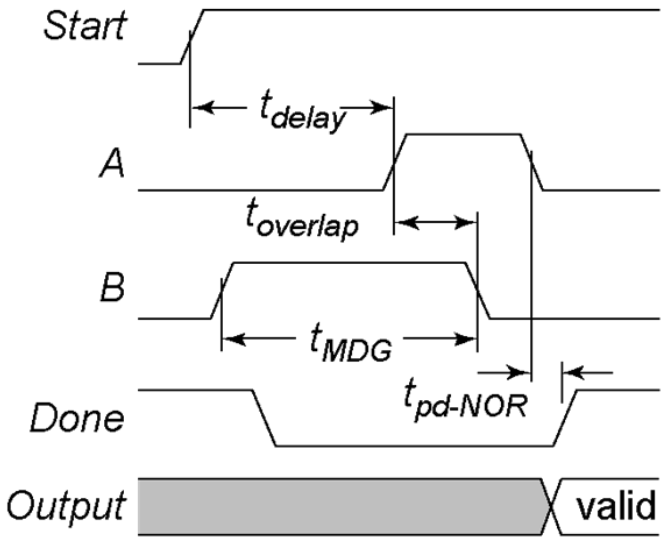
(a) Differential carry generation



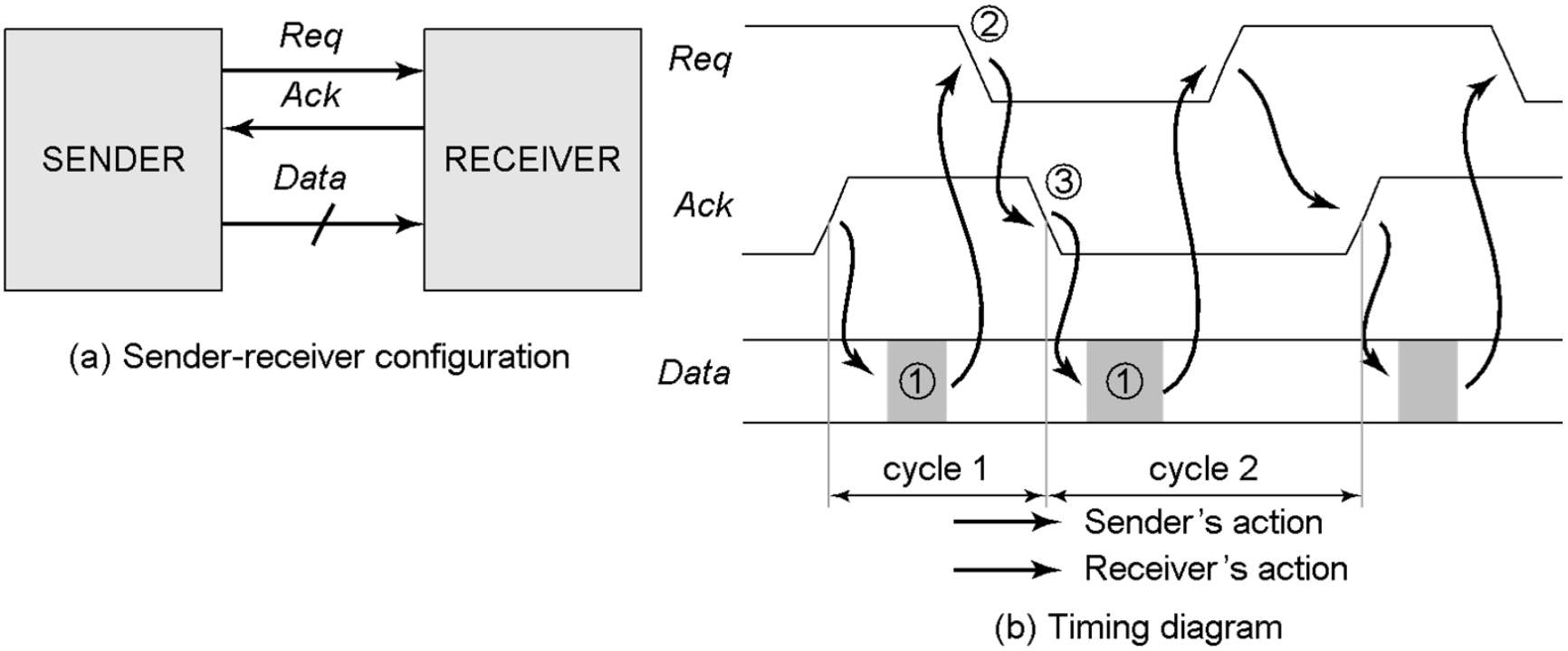
(b) Completion signal

Completion Signal Using Current Sensing



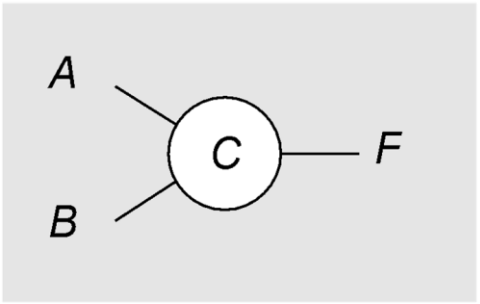


Hand-Shaking Protocol



Two Phase Handshake

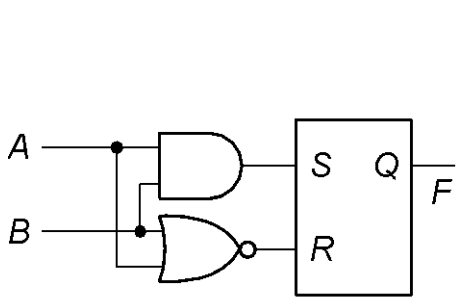
Event Logic – The Muller-C Element



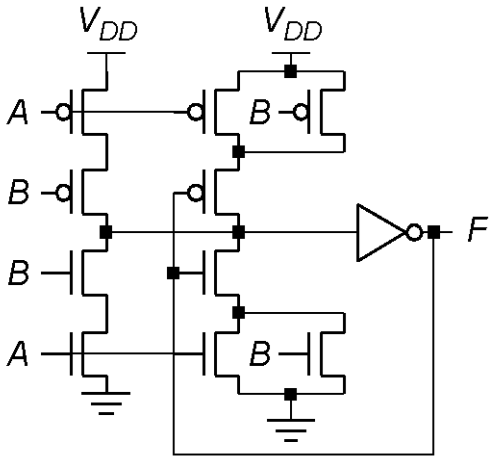
(a) Schematic

A	B	$F_{n\square 1}$
0	0	0
0	1	F_n
1	0	F_n
1	1	1

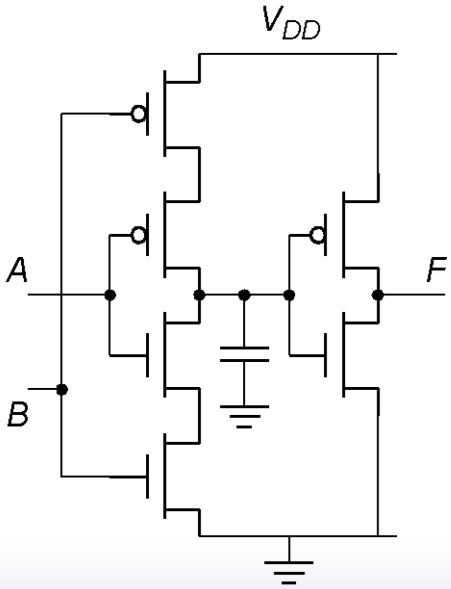
(b) Truth table



(a) Logic

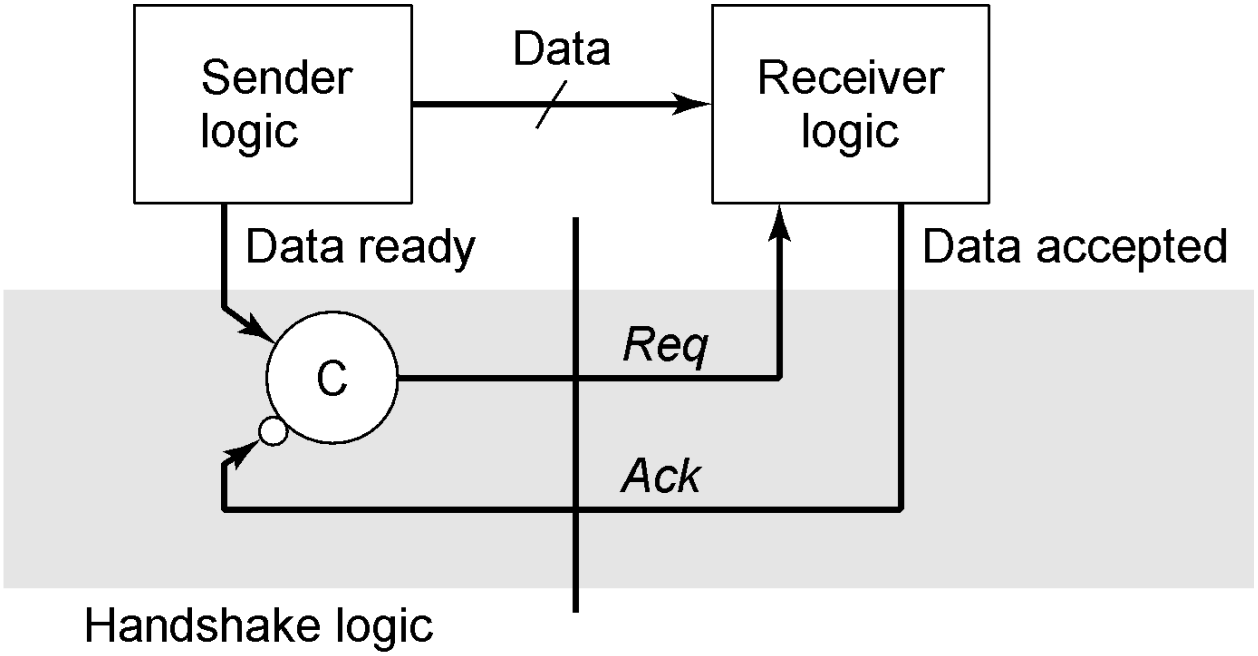


(b) Majority Function



(c) Dynamic

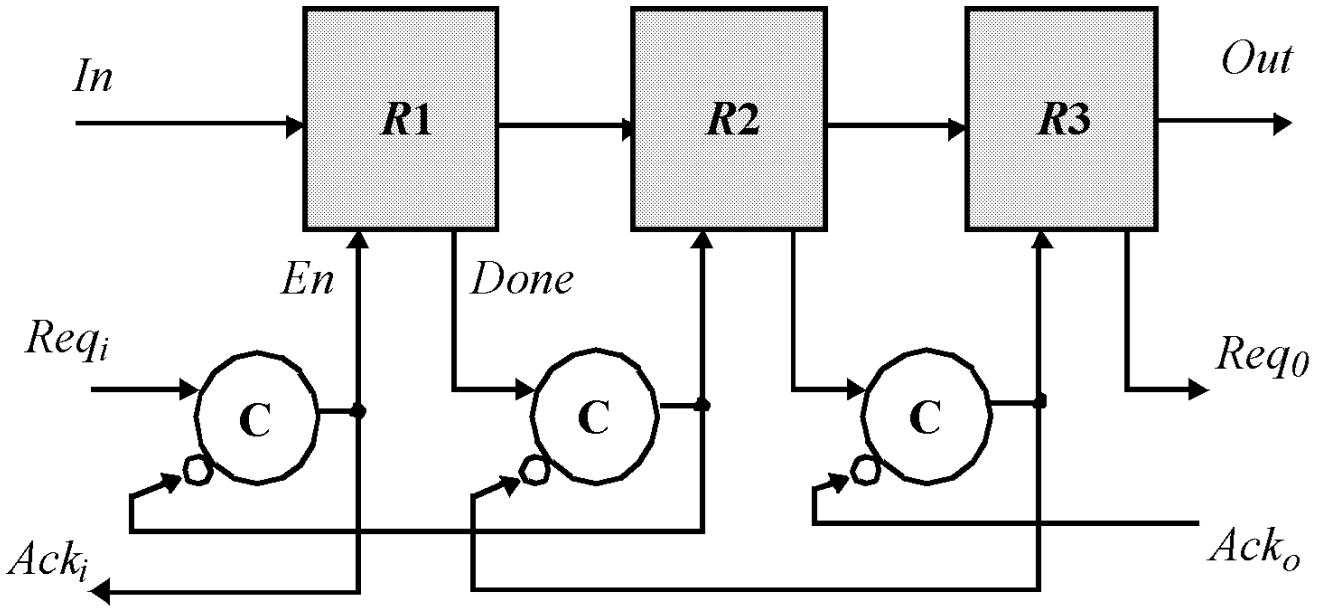
2-Phase Handshake Protocol



Advantage : FAST - minimal # of signaling events (important for global interconnect)

Disadvantage : edge - sensitive, has state

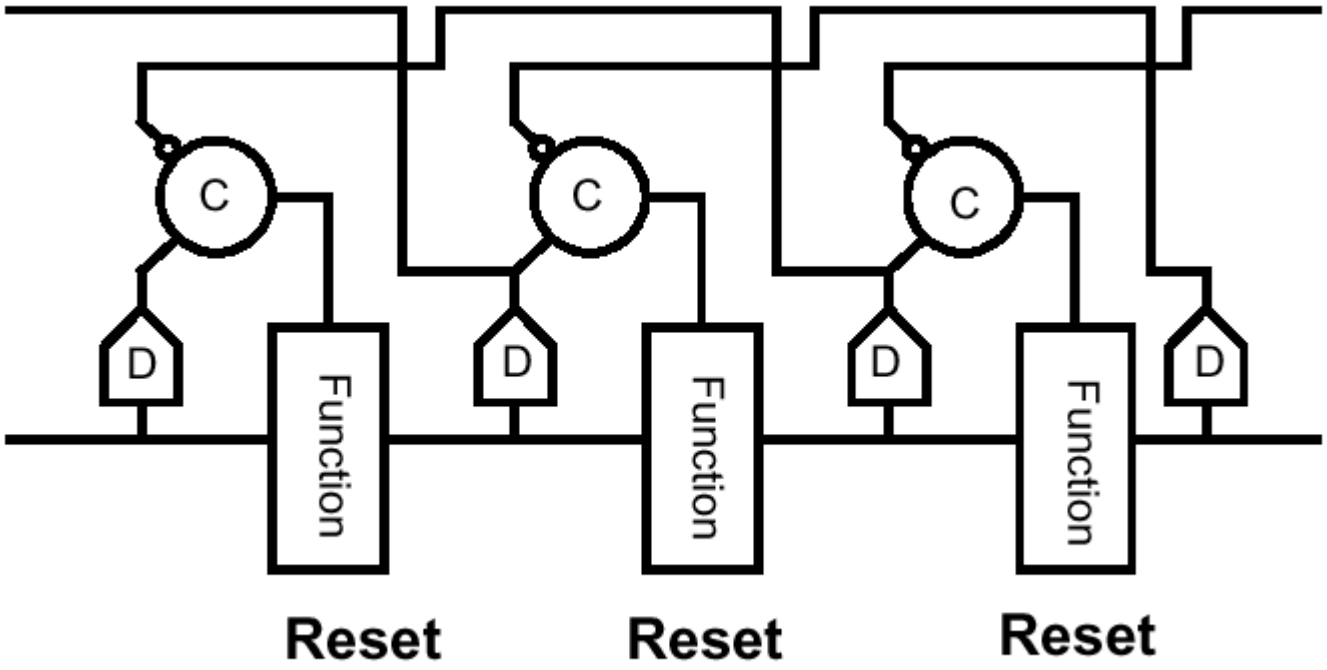
Example: Self-timed FIFO



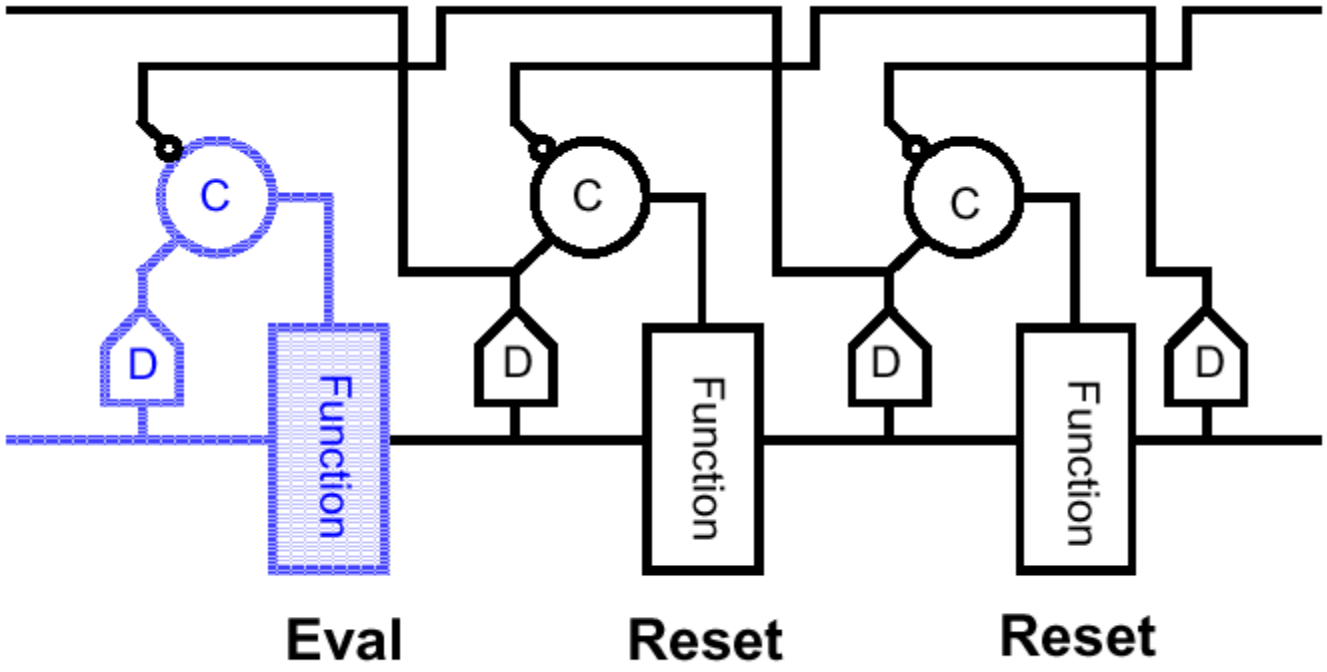
All 1s or 0s -> pipeline empty

Alternating 1s and 0s -> pipeline full

2-Phase Protocol

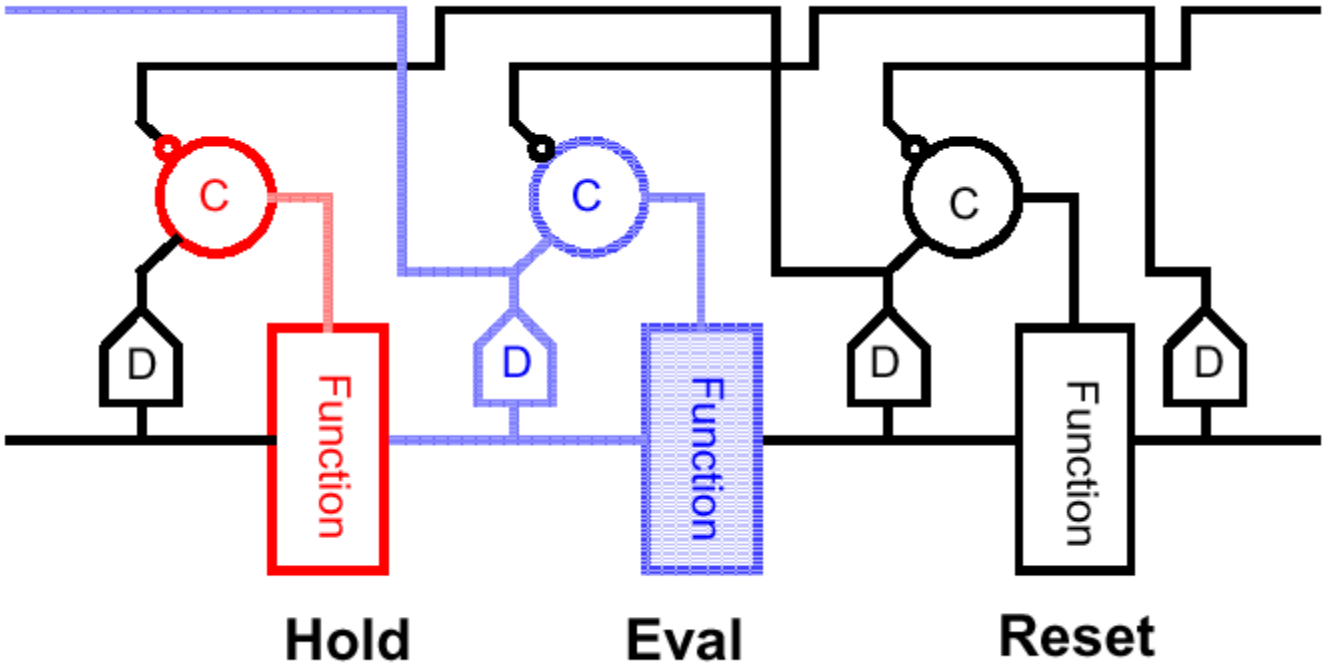


Example

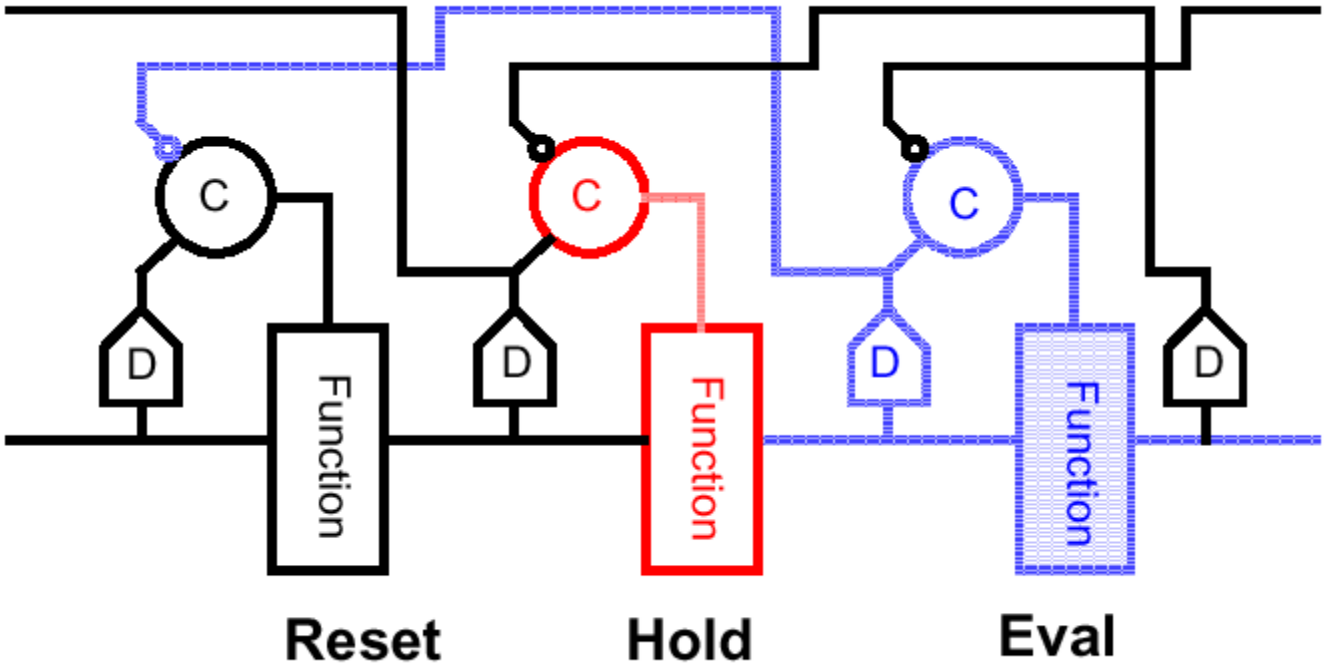


From [Horowitz]

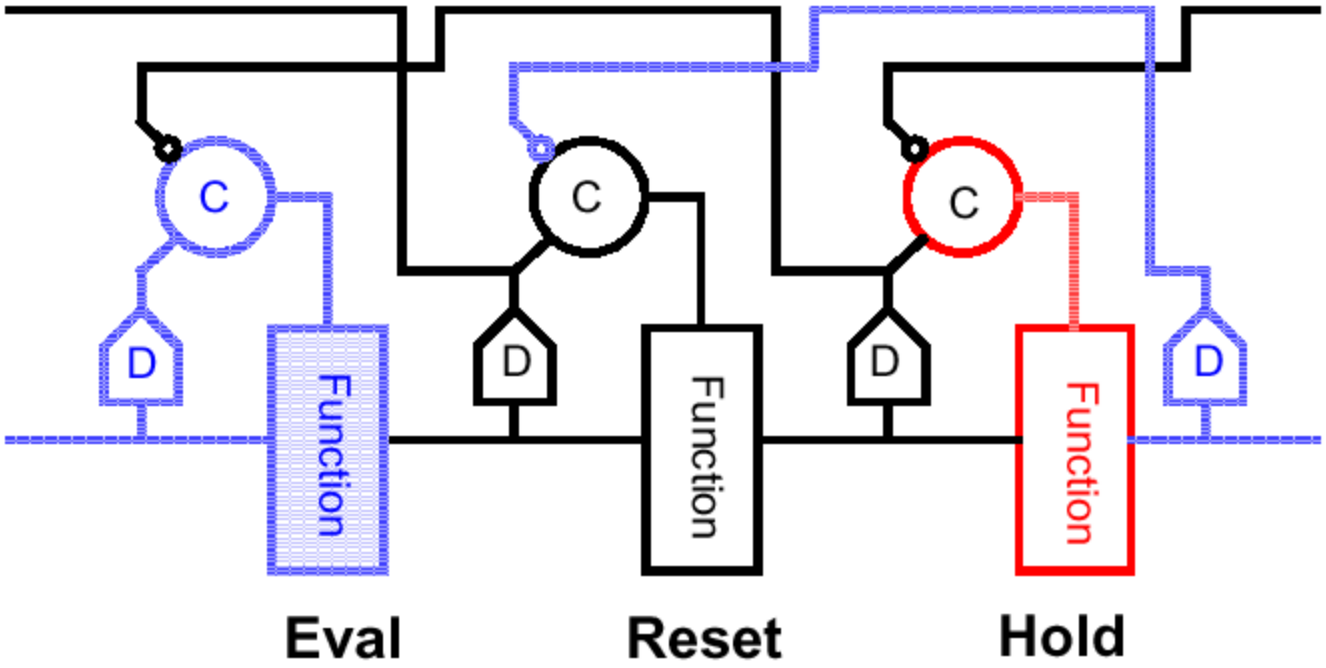
Example



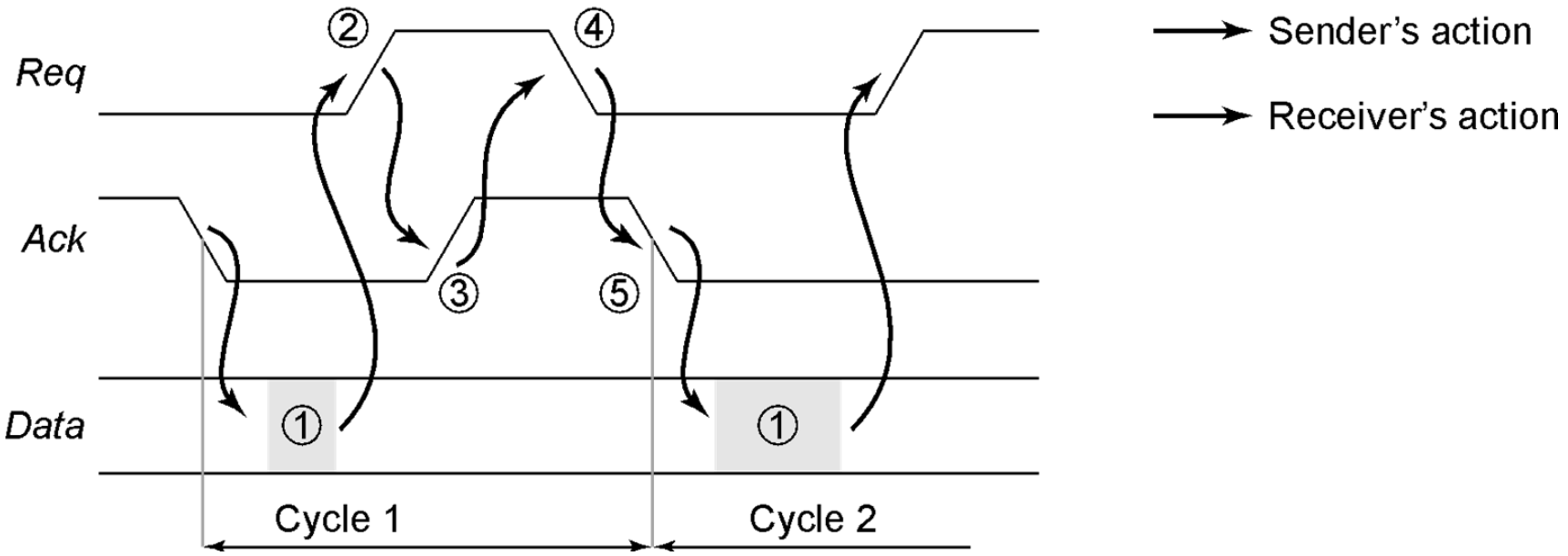
Example



Example



4-Phase Handshake Protocol

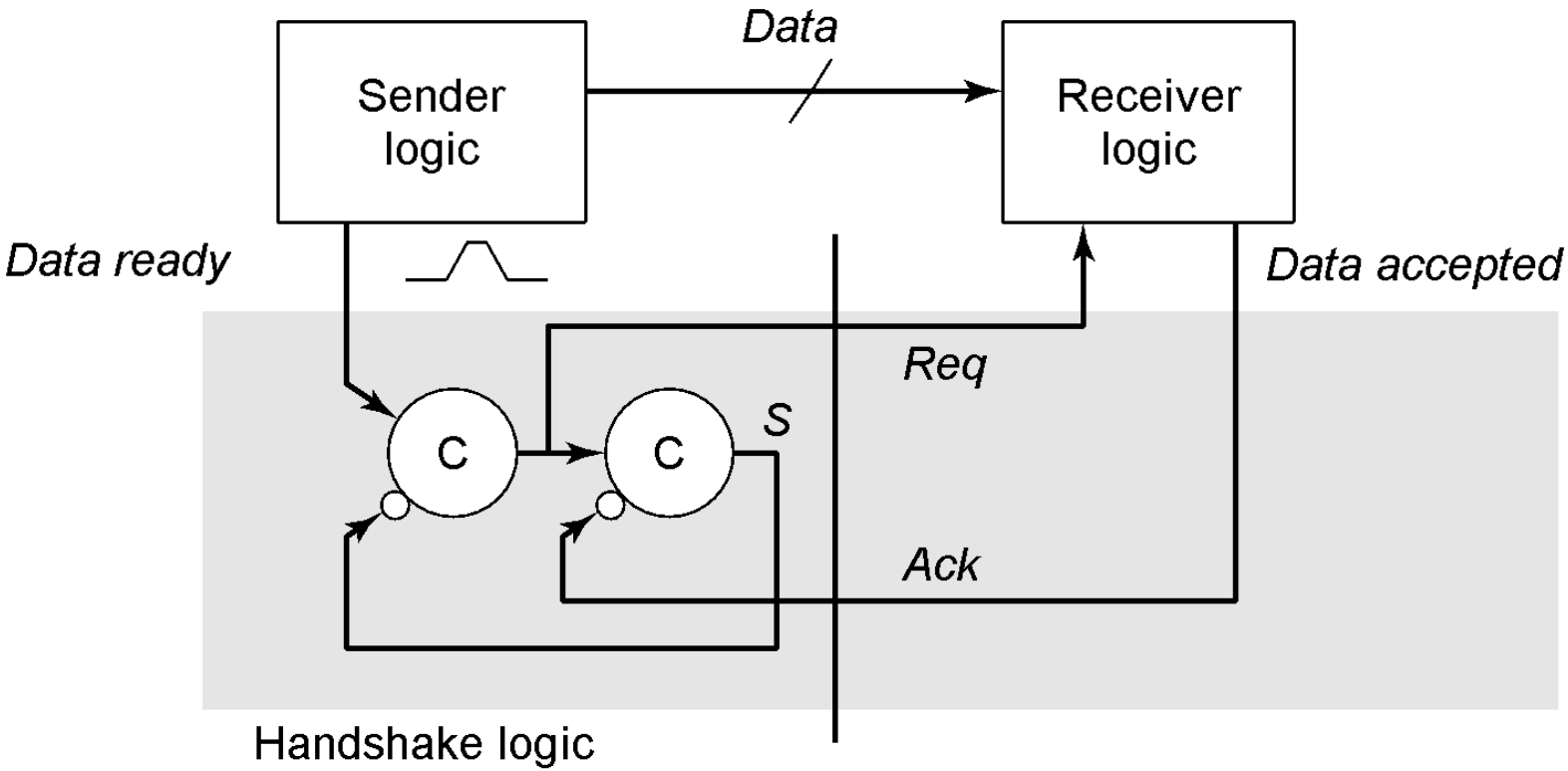


Also known as RTZ

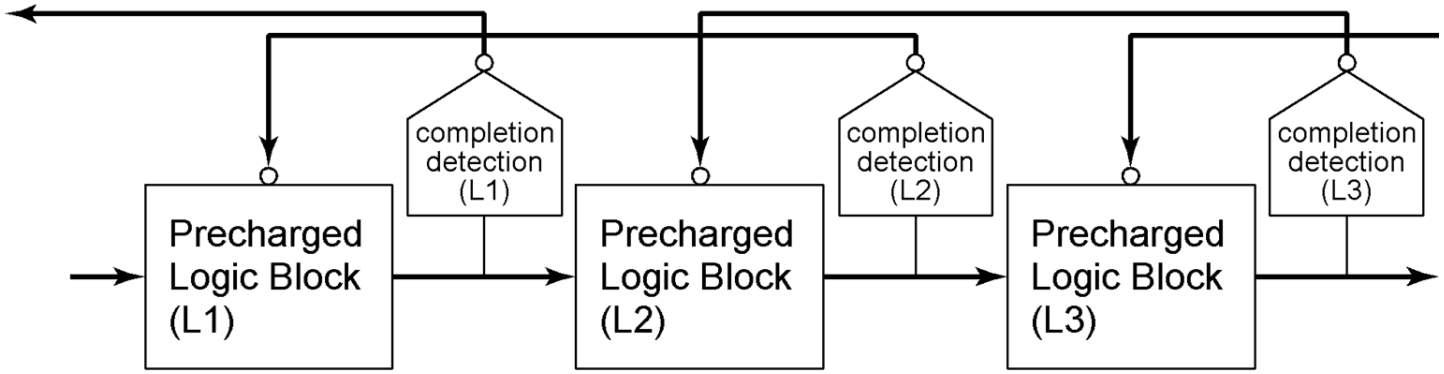
Slower, but unambiguous

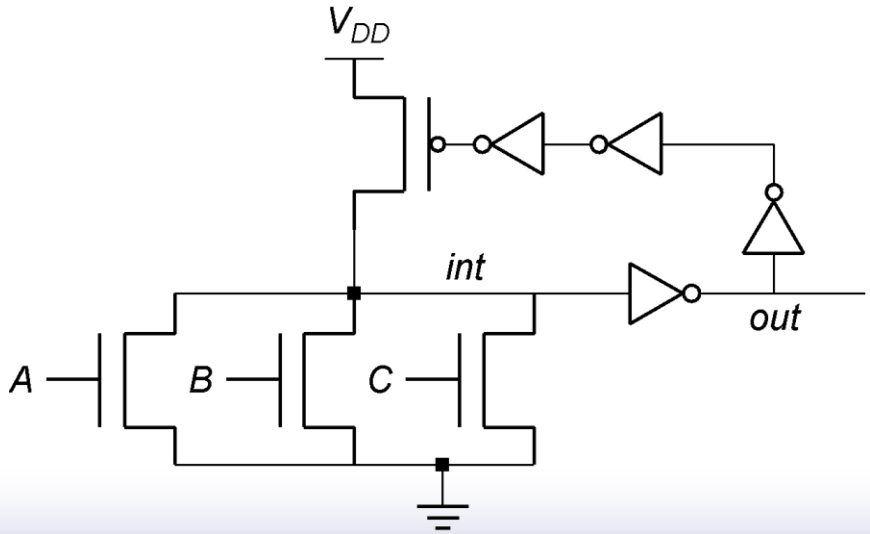
4-Phase Handshake Protocol

Implementation using Muller-C elements



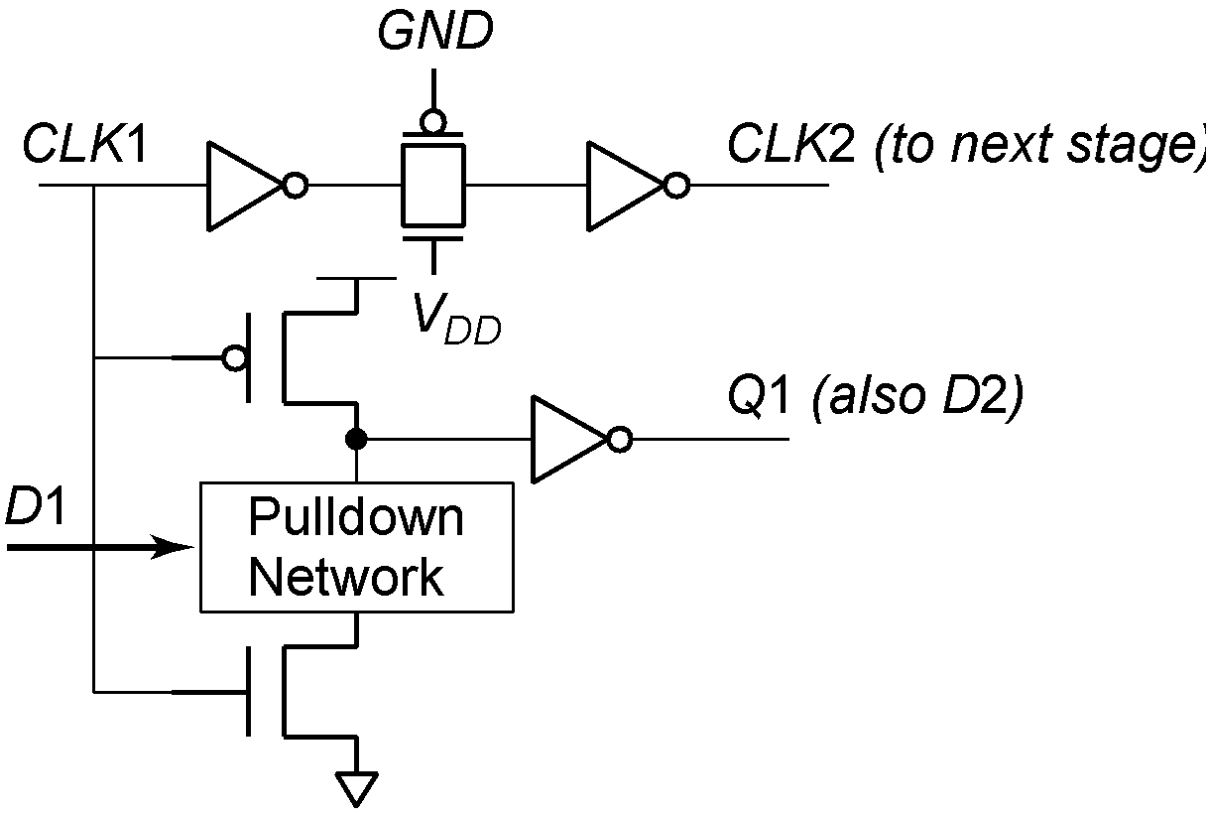
Self-Resetting Logic



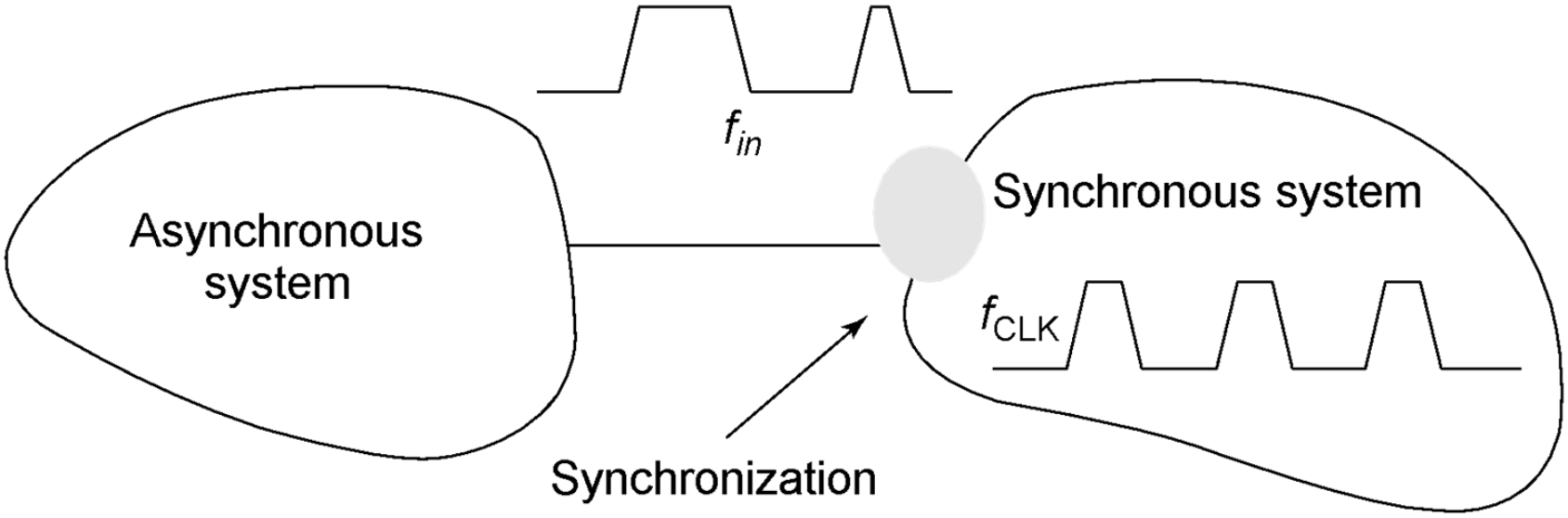


Post-charge logic

Clock-Delayed Domino



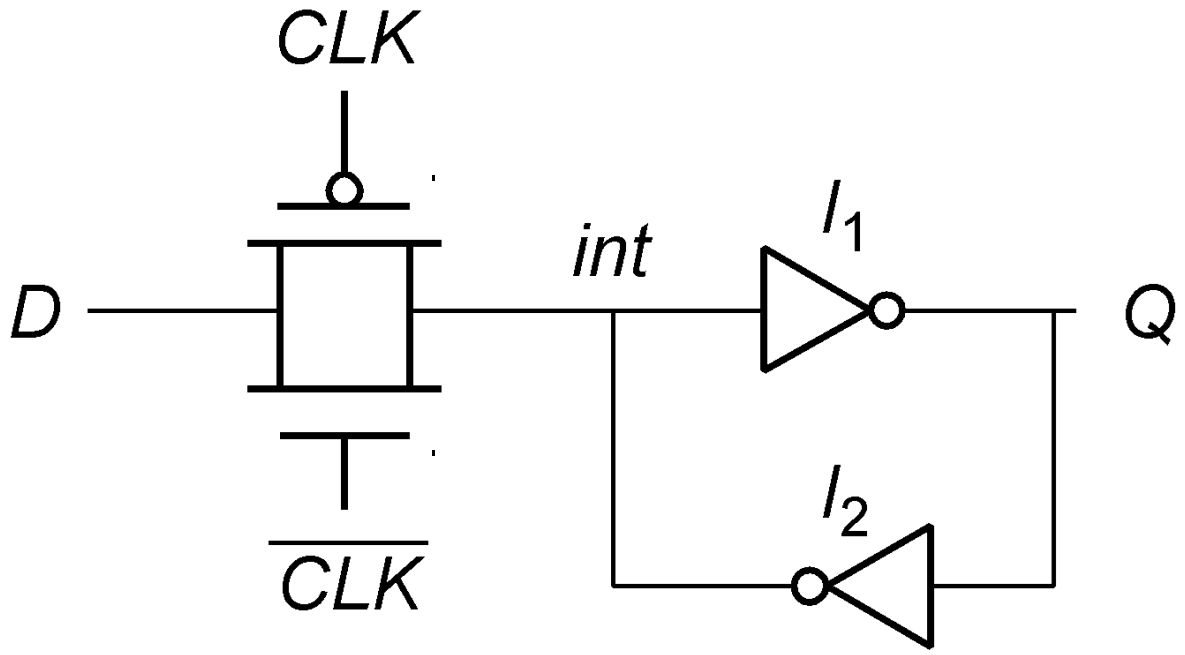
Asynchronous-Synchronous Interface



Synchronizers and Arbiters

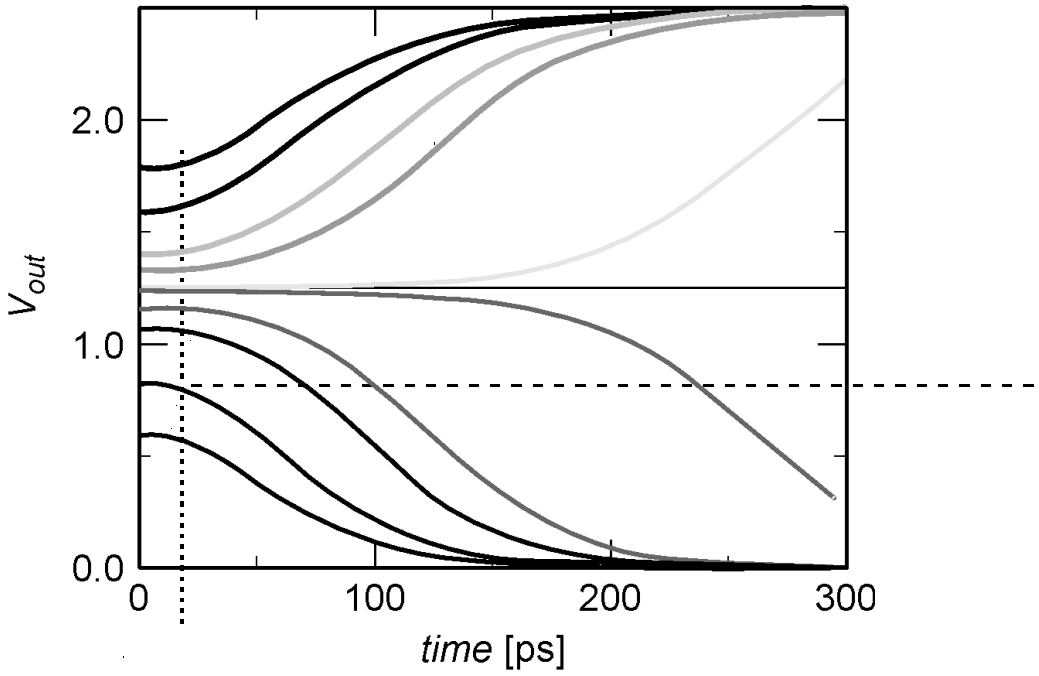
- Arbiter: Circuit to decide which of 2 events occurred first
- Synchronizer: Arbiter with clock φ as one of the inputs
- Problem: Circuit HAS to make a decision in limited time - which decision is not important
- Caveat: It is impossible to ensure correct operation
- But, we can decrease the error probability at the expense of delay

A Simple Synchronizer



- **Data sampled on rising edge of the clock**
- **Latch will eventually resolve the signal value, but ... this might take infinite time!**

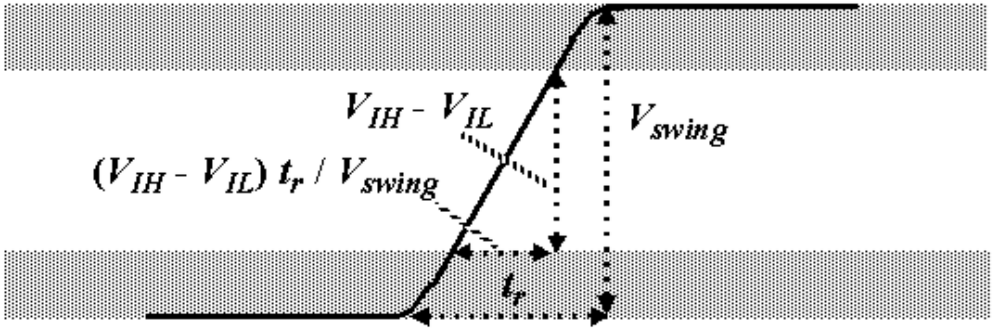
Synchronizer: Output Trajectories



Single-pole model for a flip-

$$v(t) = V_{MS} + (v(0) - V_{MS})e^{t/\tau}$$

Mean Time to Failure



$$N_{sync}(0) = \frac{P_{init}}{T_\phi} = \frac{\left(\frac{V_{IH} - V_{IL}}{V_{swing}}\right) t_r}{T_{signal}} \frac{1}{T_\phi}$$

$$N_{sync}(T) = \frac{P_{init} e^{-T/\tau}}{T_\phi} = \frac{(V_{IH} - V_{IL}) e^{-T/\tau}}{V_{swing}} \frac{t_r}{T_{signal} T_\phi}$$

Example

$T_f = 10$ nsec = T

$T_{signal} = 50$ nsec

$t_r = 1$ nsec

t = 310 psec

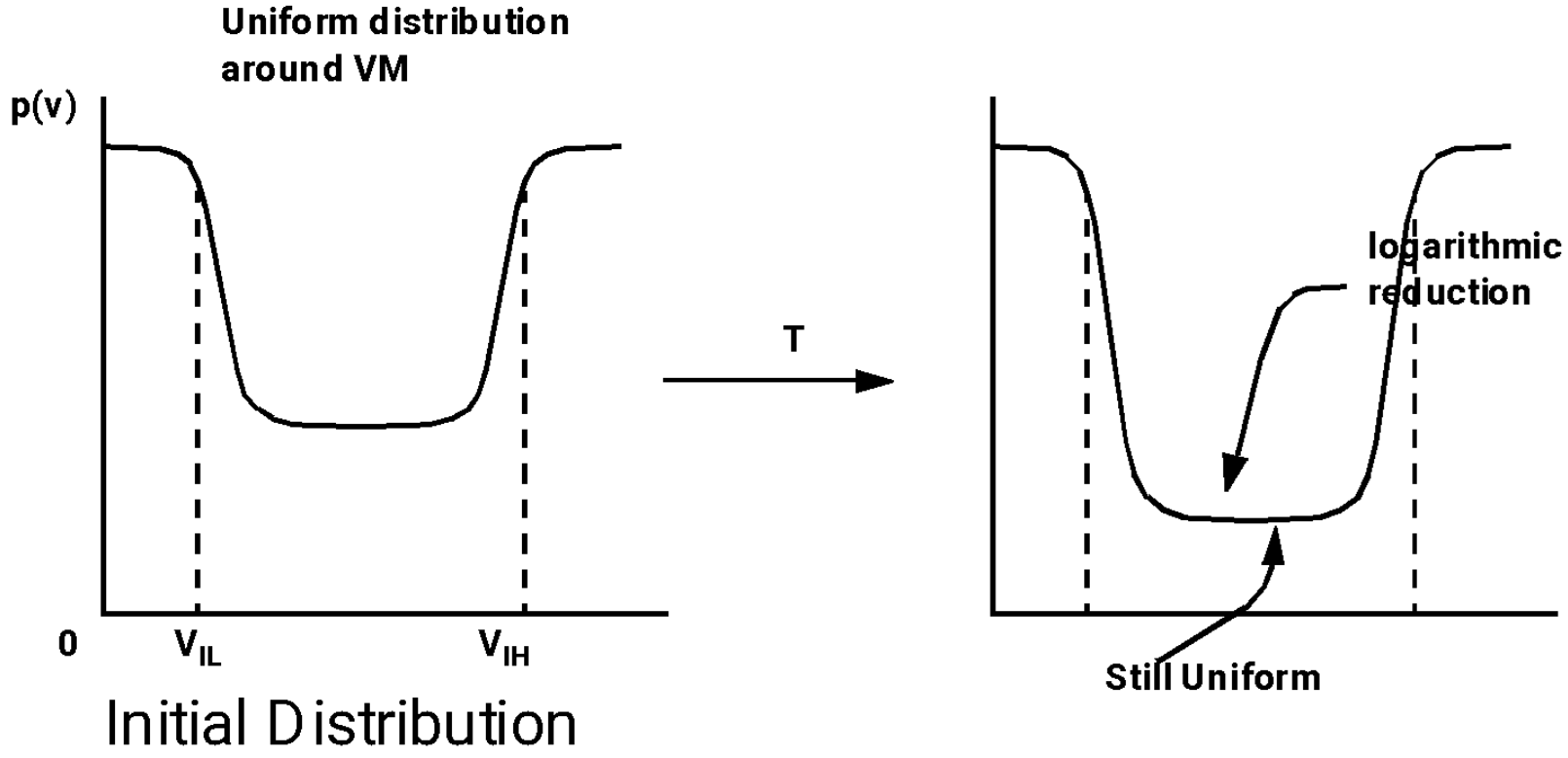
$V_{IH} - V_{IL} = 1$ V ($V_{DD} = 5$ V)

$N(T) = 3.9\ 10^{-9}$ errors/sec

MTF (T) = $2.6\ 10^{8}$ sec = 8.3 years

MTF (0) = 2.5 μsec

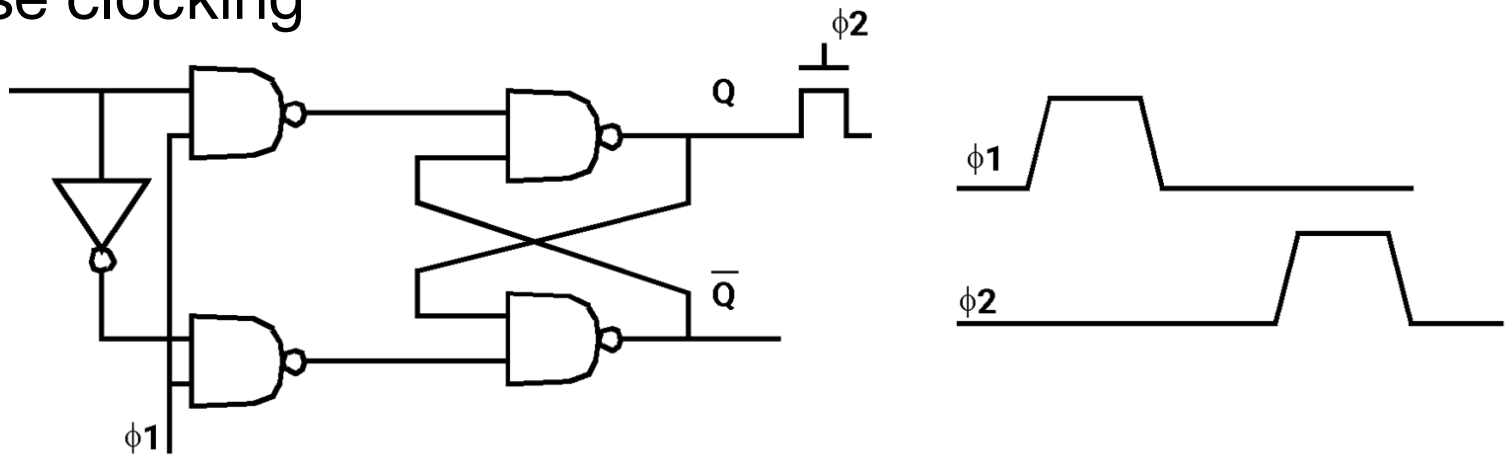
Influence of Noise



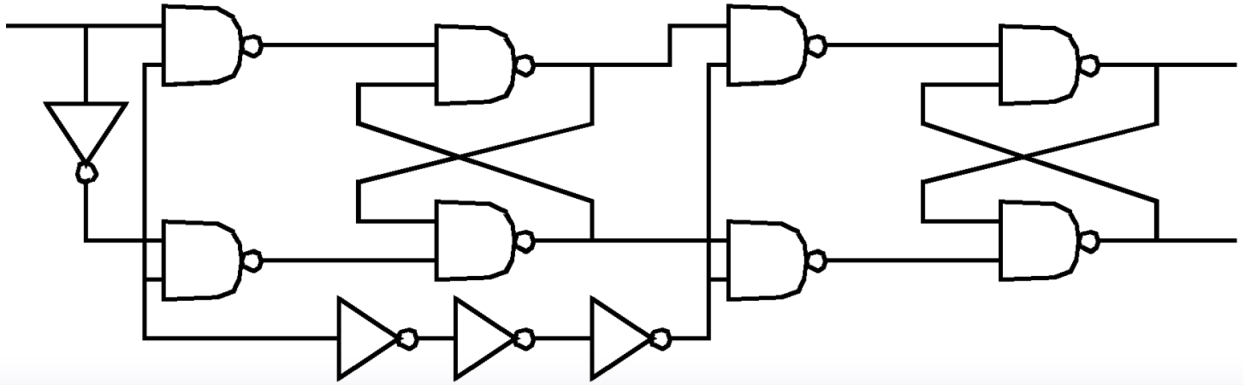
Low amplitude noise does not influence synchronization behavior

Typical Synchronizers

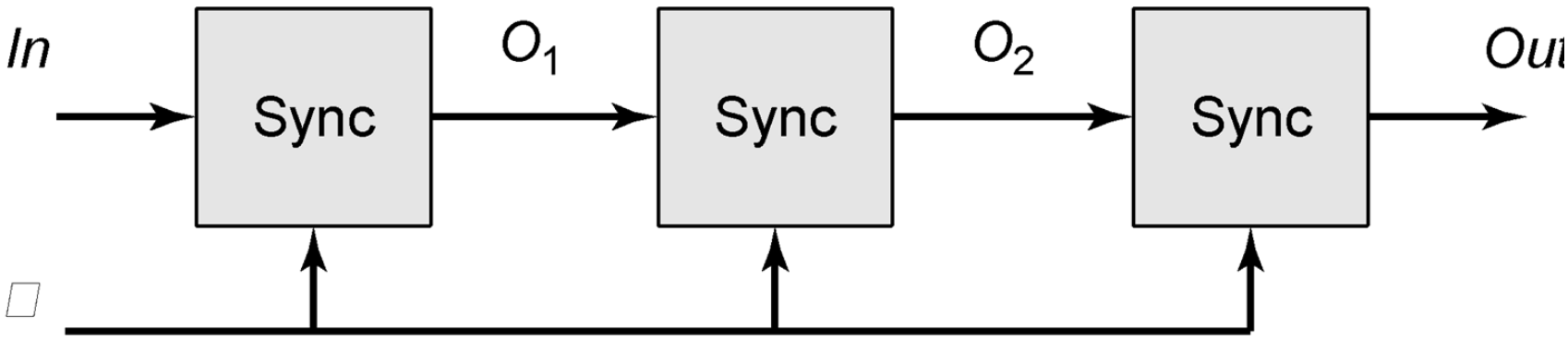
2 phase clocking circuit



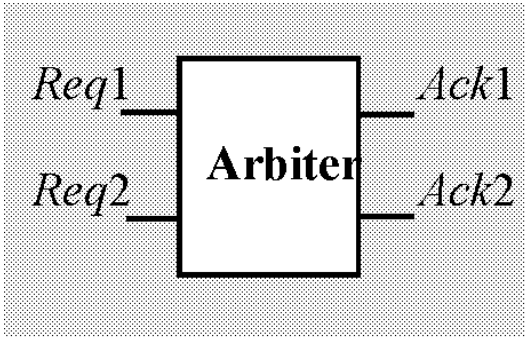
Using delay line



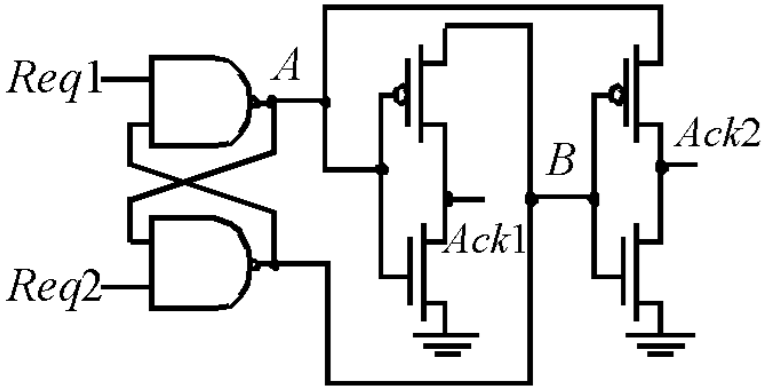
Cascaded Synchronizers Reduce MTF



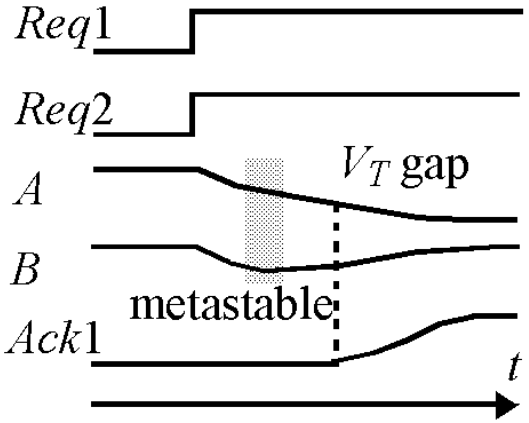
Arbiters



(a) Schematic symbol

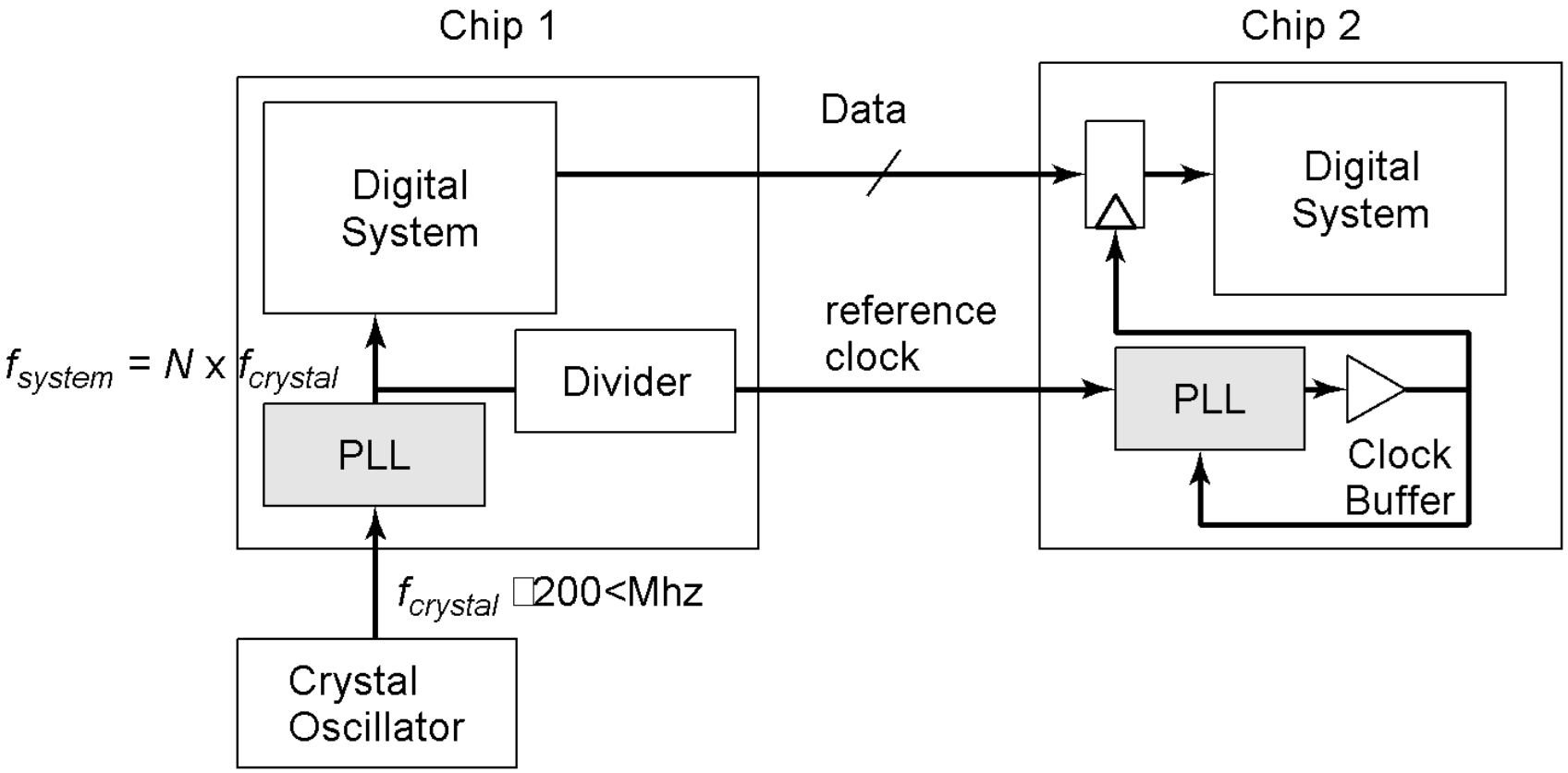


(b) Implementation

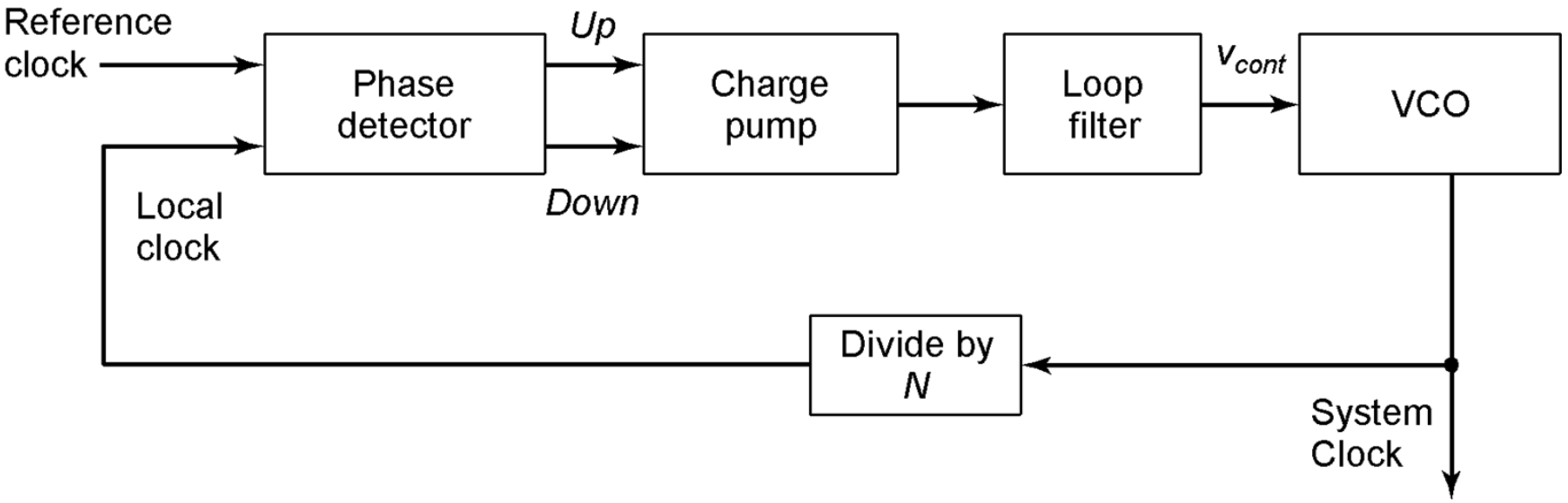


(c) Timing diagram

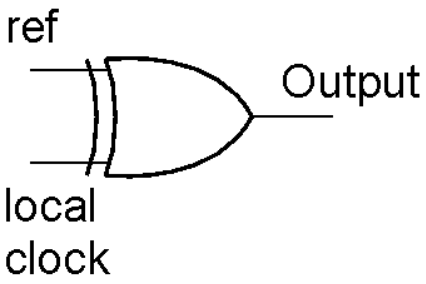
PLL-Based Synchronization



PLL Block Diagram

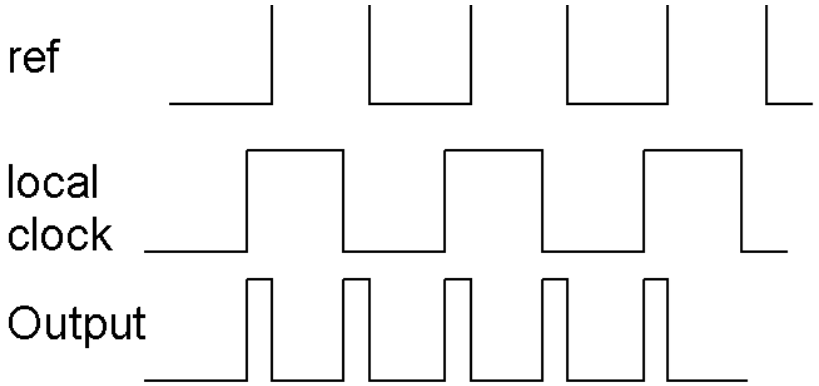


Phase Detector

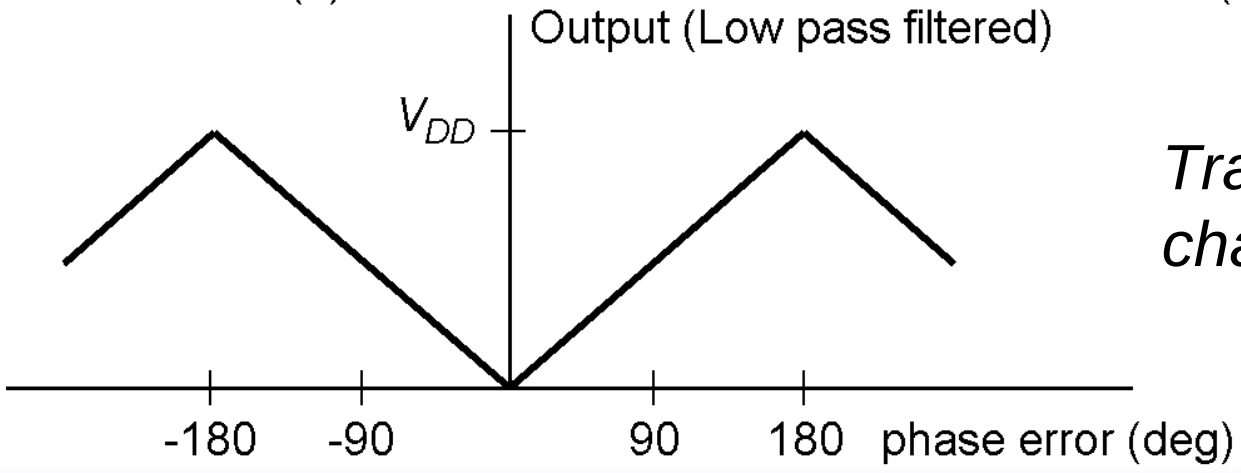


(a)

Output before filtering



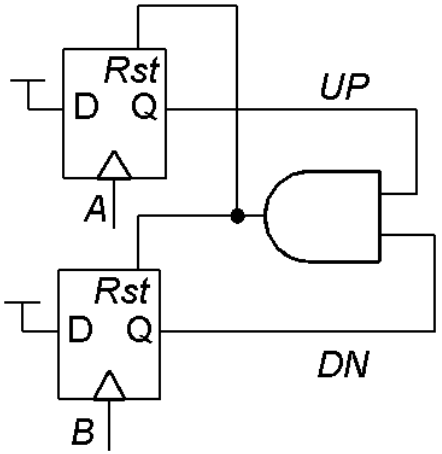
(b)



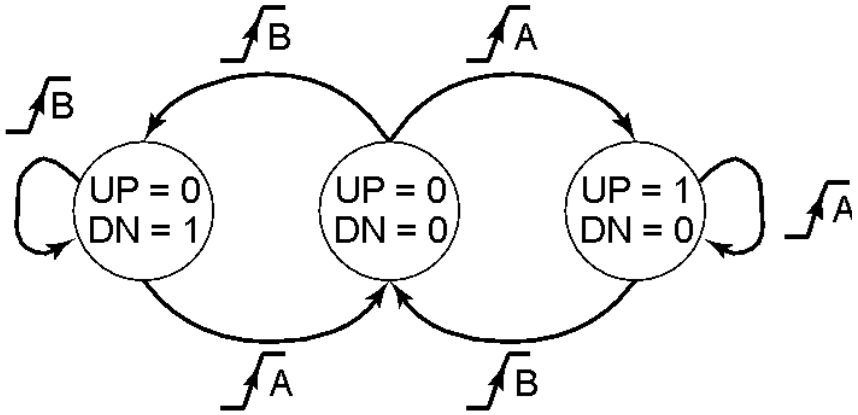
(c)

Transfer characteristic

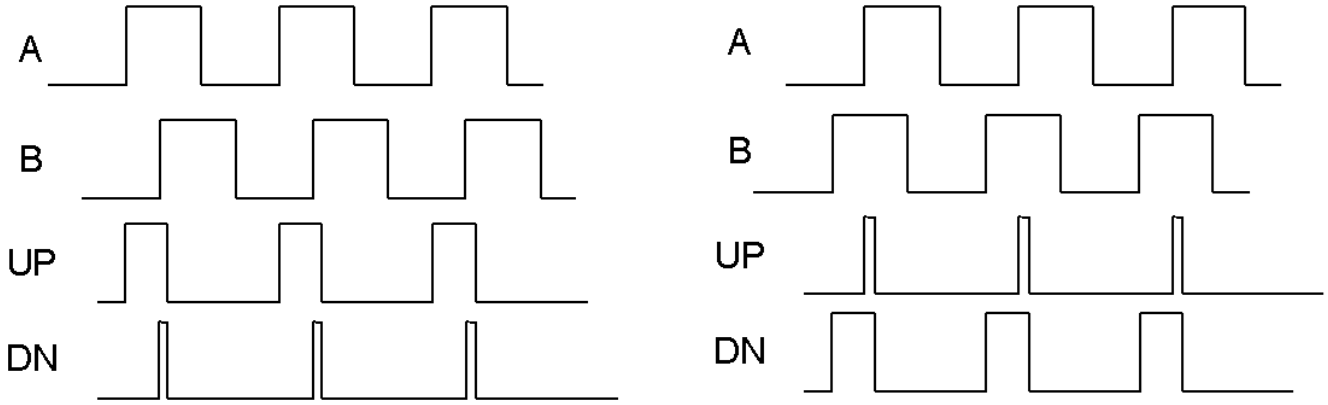
Phase-Frequency Detector



(a) schematic

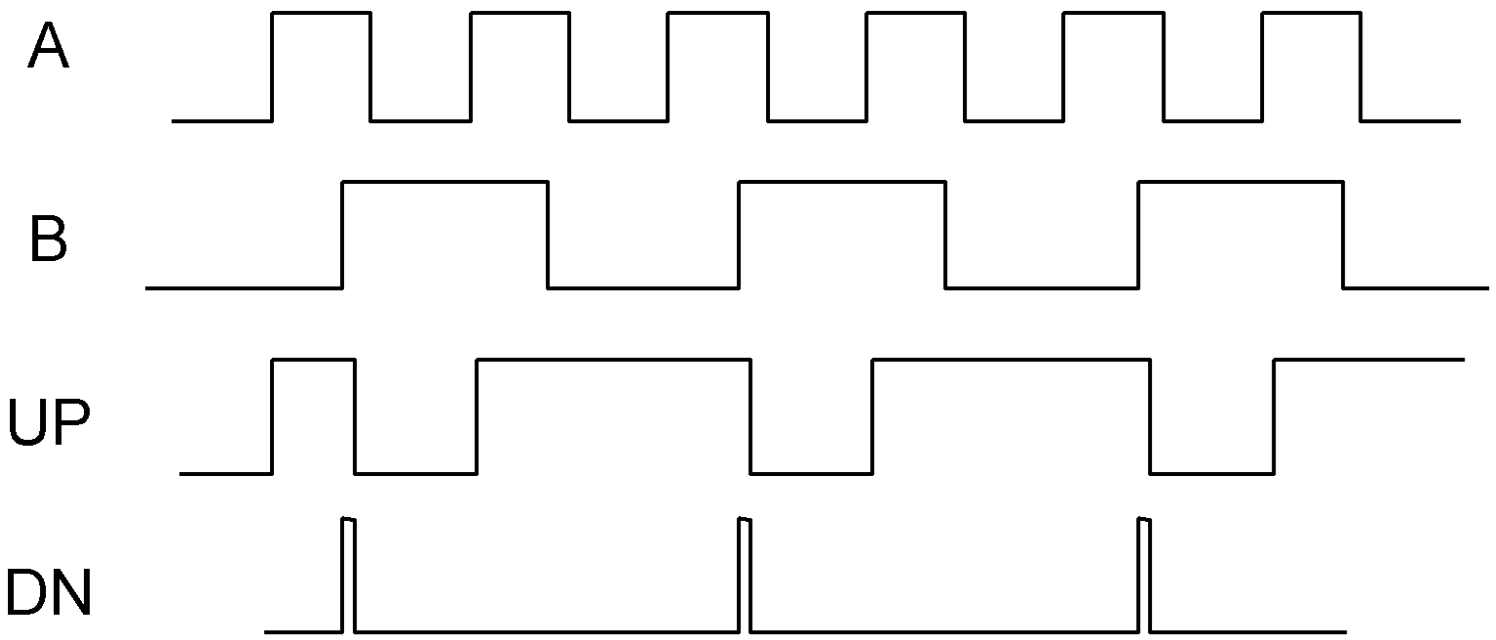


(b) state transition diagram

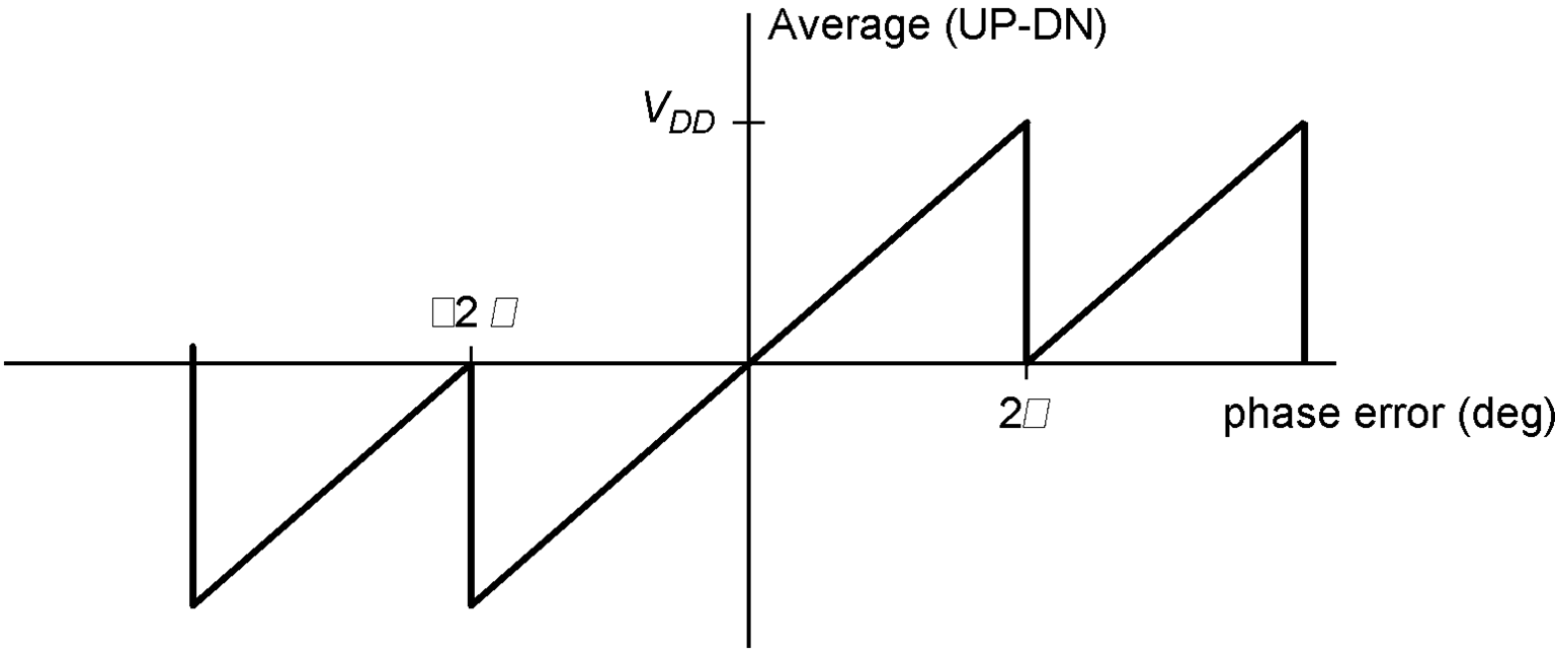


(c) Timing waveforms

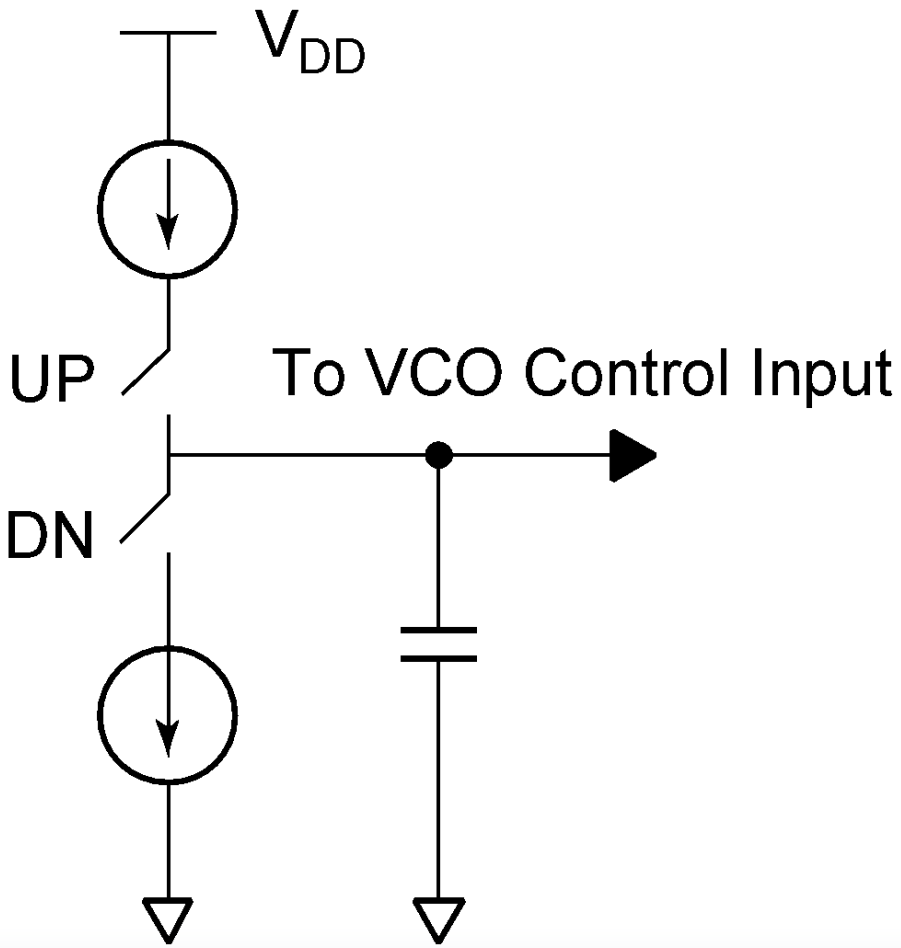
PFD Response to Frequency



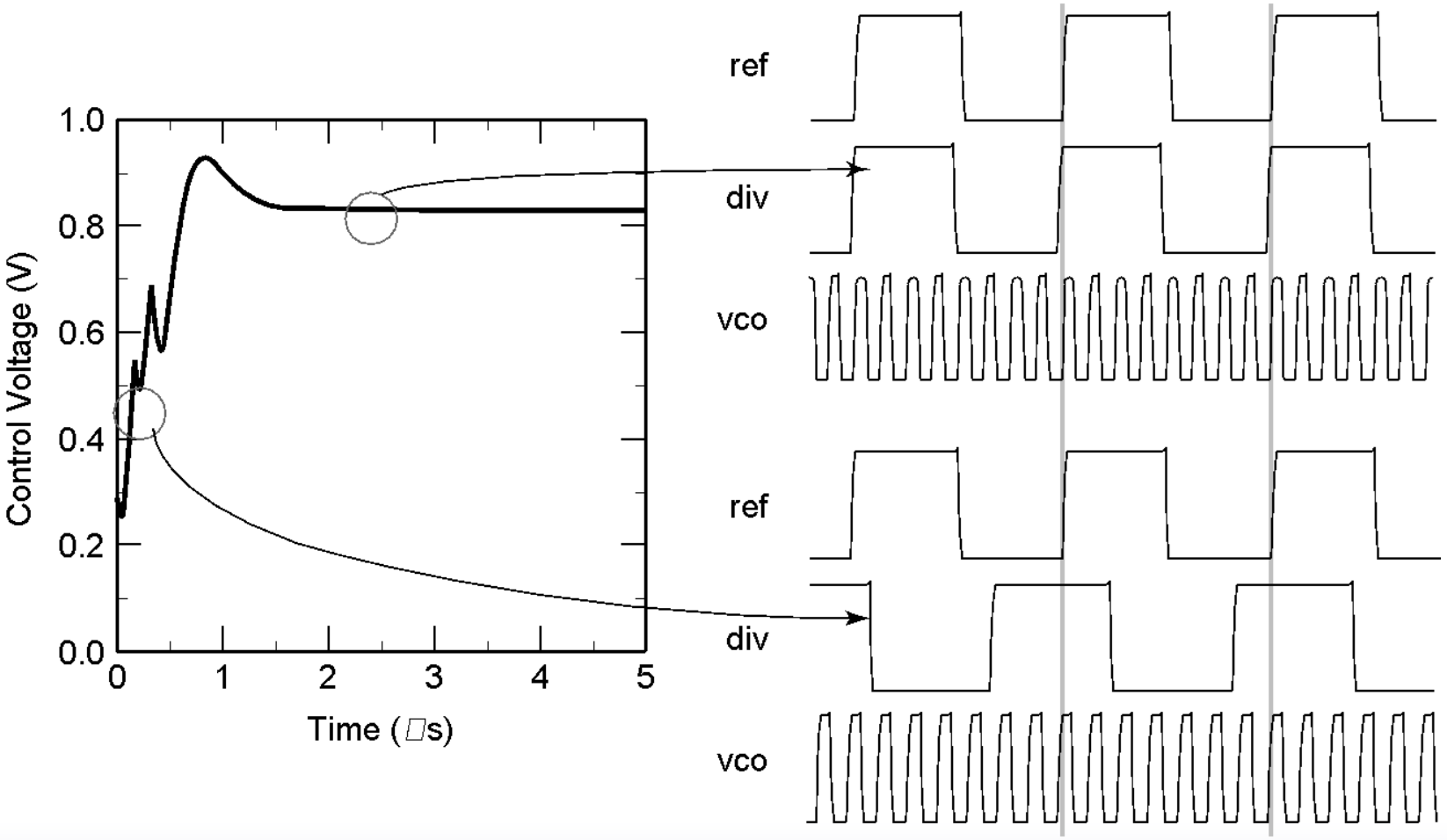
PFD Phase Transfer Characteristic



Charge Pump

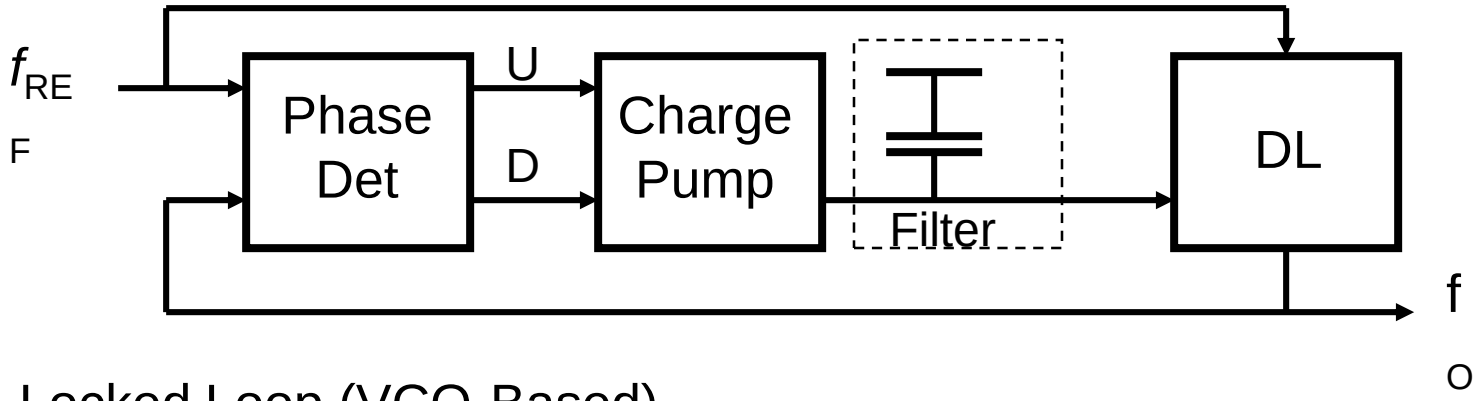


PLL Simulation

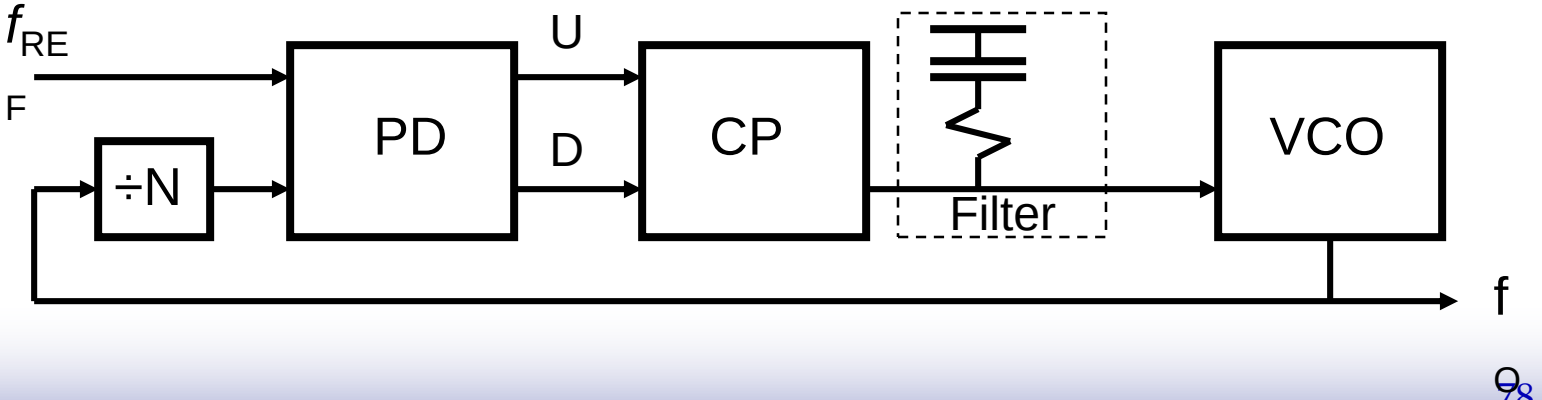


Clock Generation using DLLs

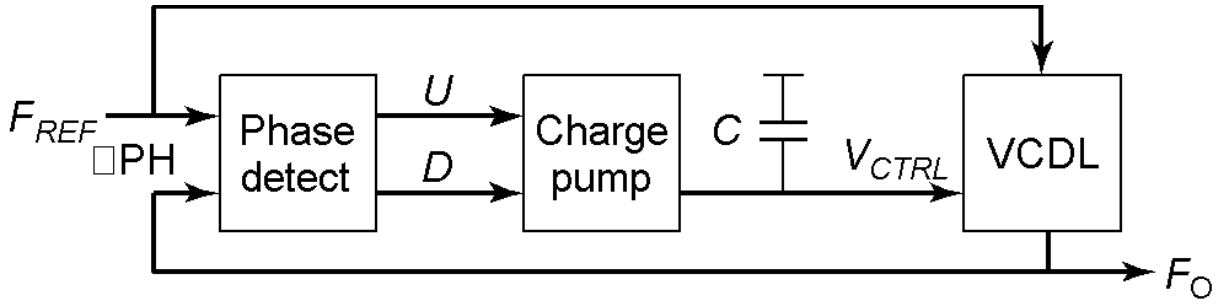
Delay-Locked Loop (Delay Line Based)



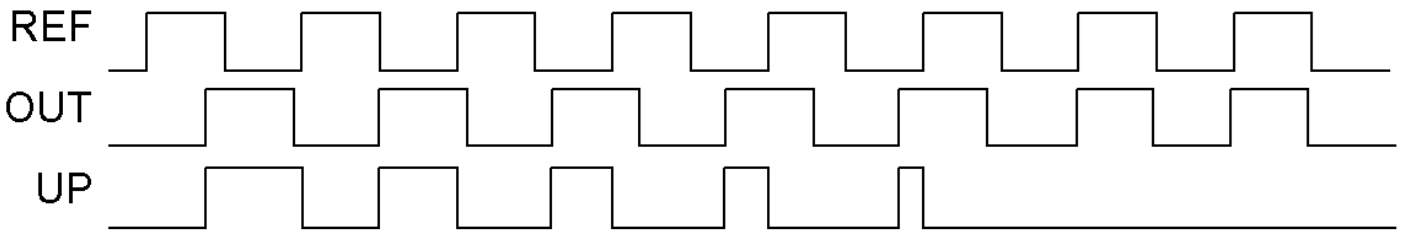
Phase-Locked Loop (VCO-Based)



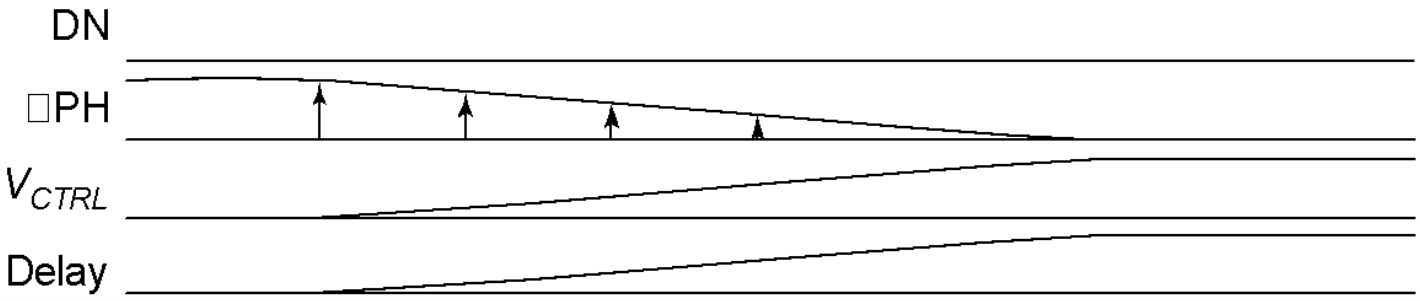
Delay Locked Loop



(a)



(b)



(c)

DLL-Based Clock Distribution