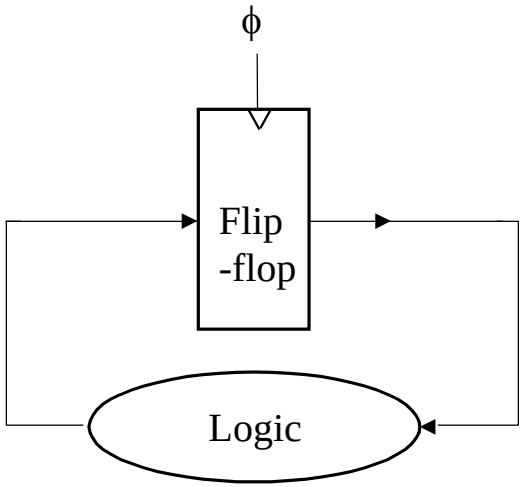
ECE 225
High Speed Integrated Circuit Design

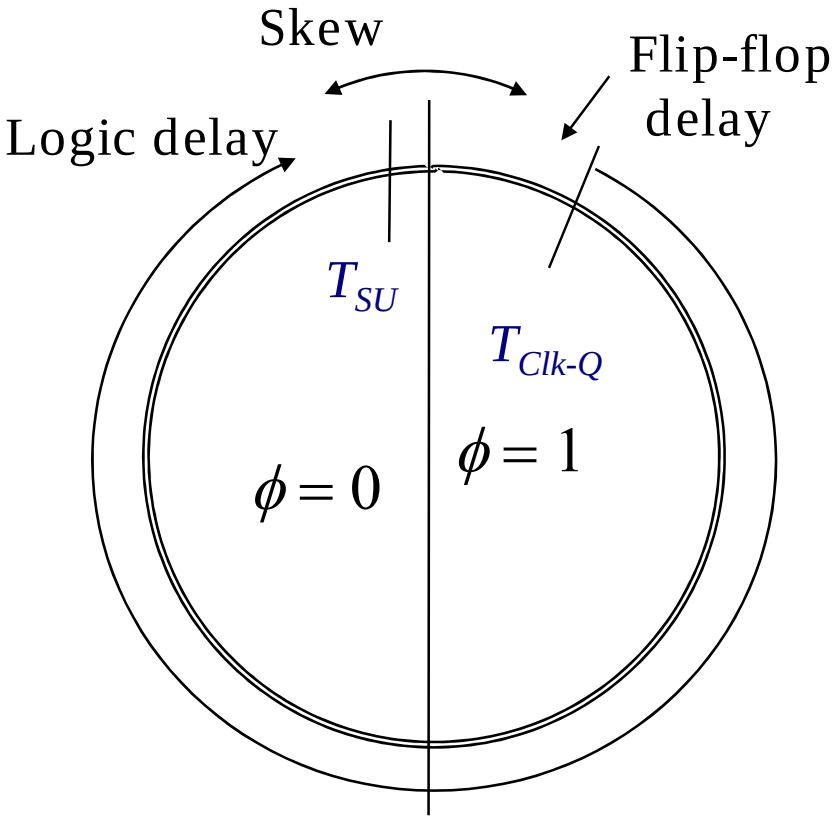
Lecture 14

Prof. Kaustav Banerjee
Electrical and Computer Engineering
E-mail: kaustav@ece.ucsb.edu

Timing Issues

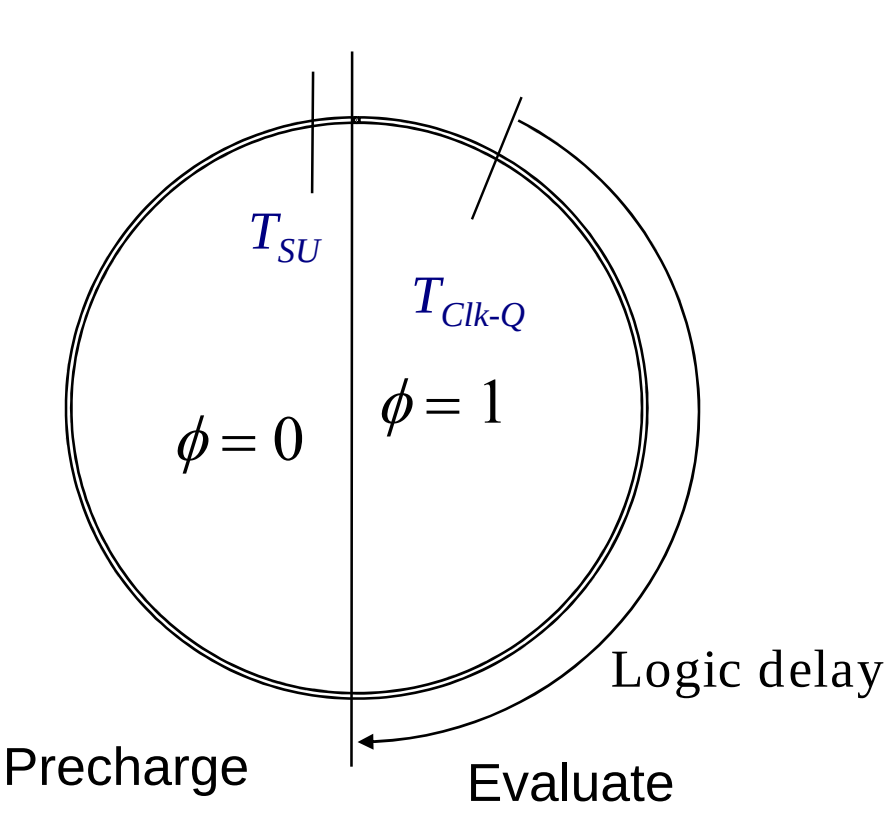
Flip-Flop – Based Timing

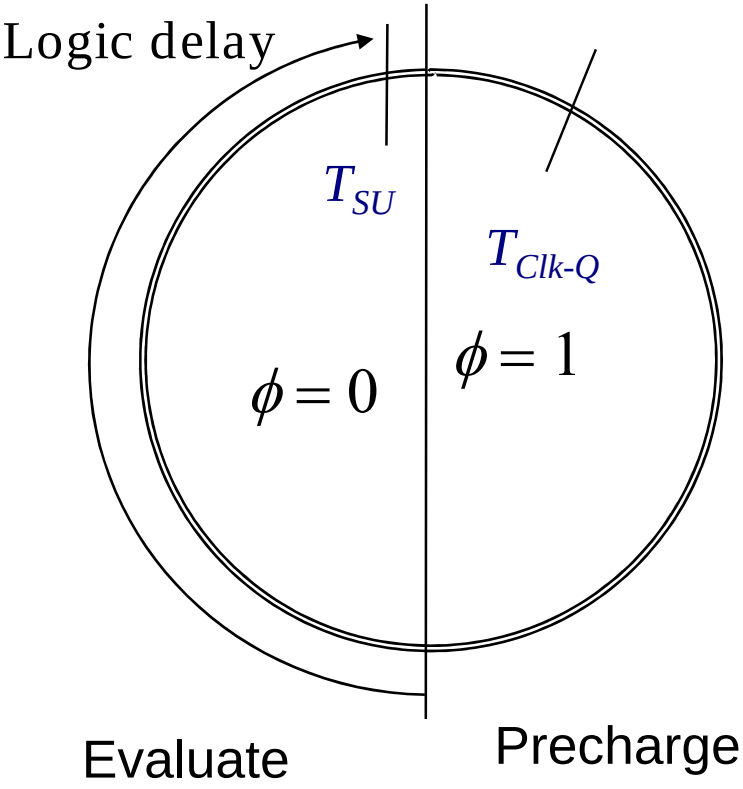




Representation after
M. Horowitz, VLSI Circuits 1996.

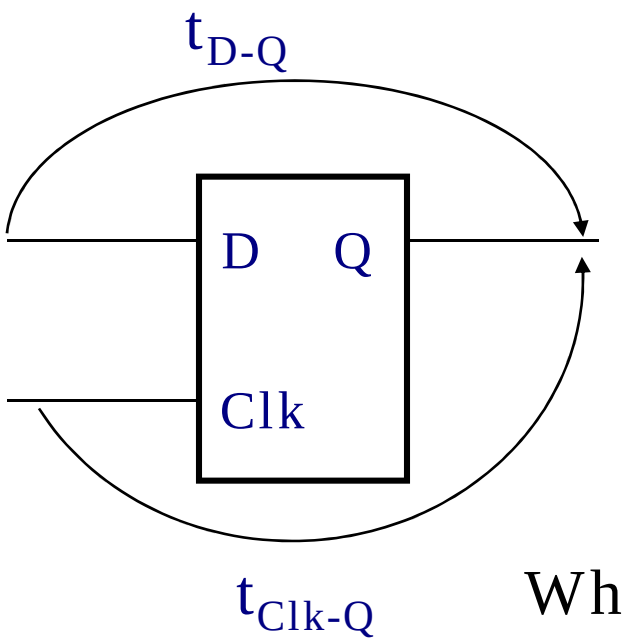
Flip-Flops and Dynamic Logic





Flip-flops are used only with static logic

Latch timing



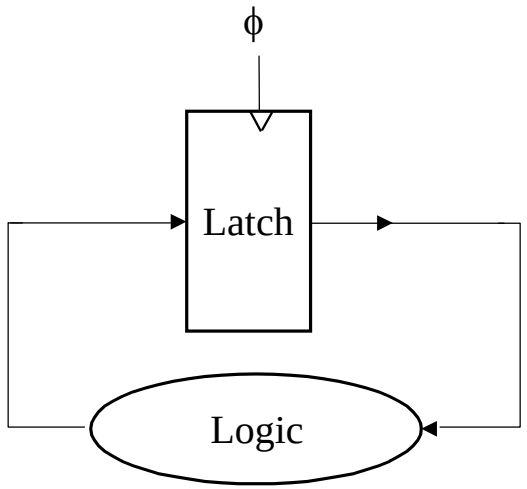
When data arrives to transparent latch

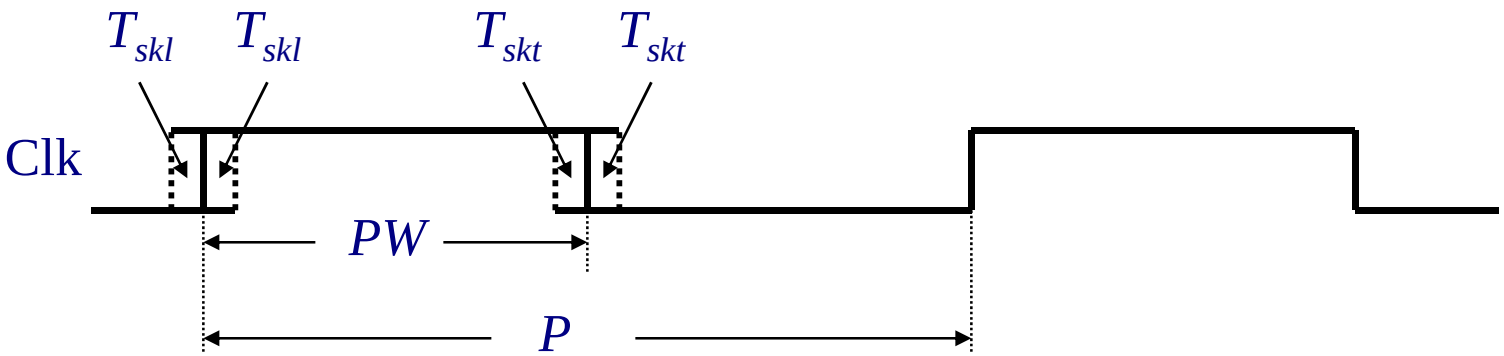
Latch is a ‘soft’ barrier

When data arrives to closed latch

Data has to be ‘re-launched’

Single-Phase Clock with Latches

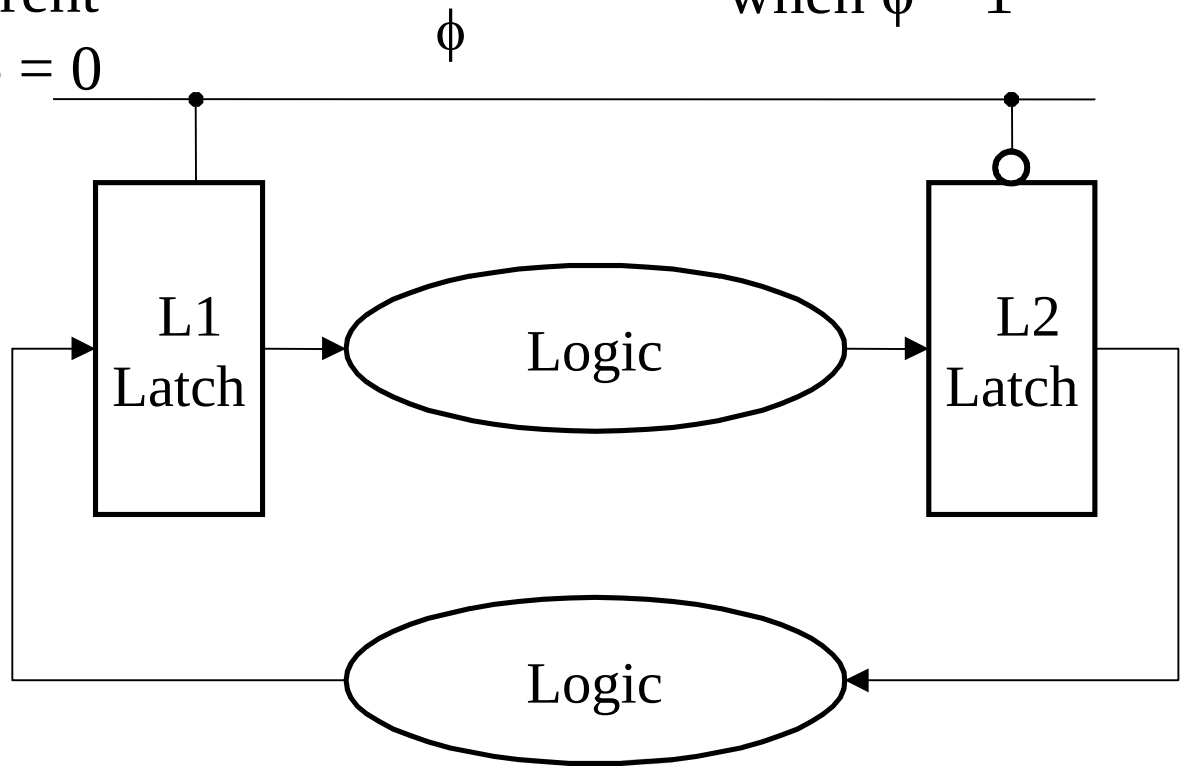




Latch-Based Design

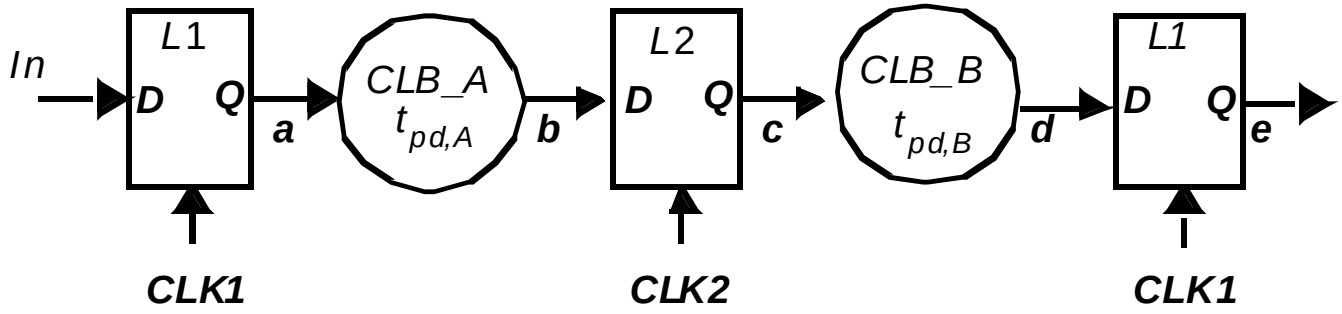
L1 latch is transparent when $\phi = 0$

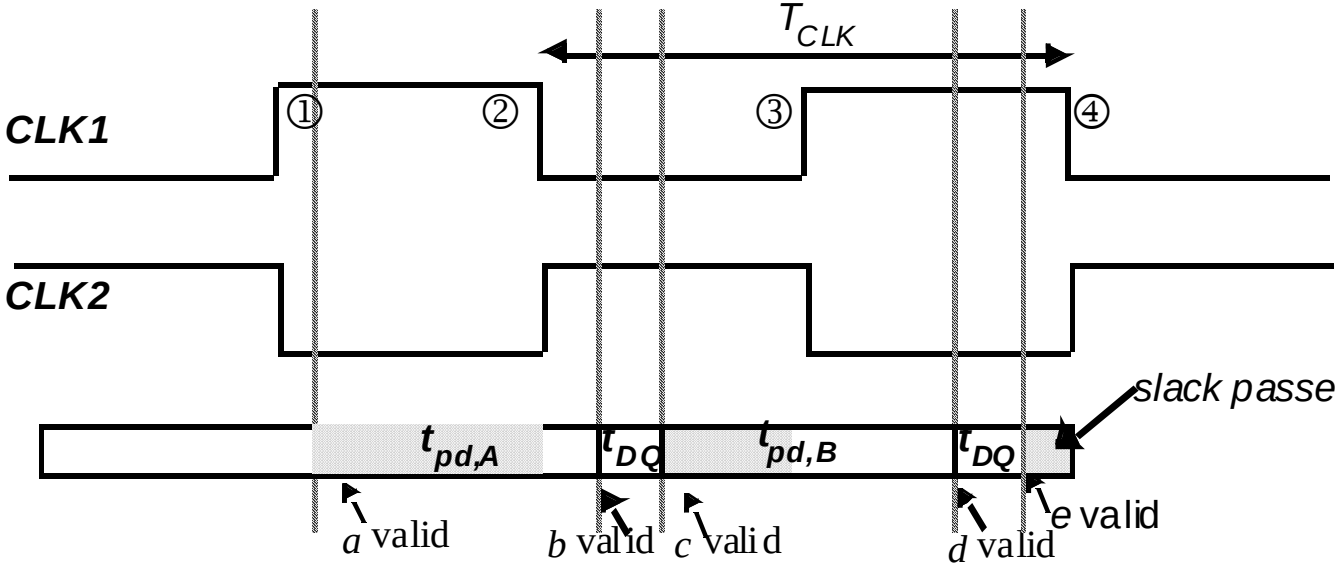
L2 latch is transparent when $\phi = 1$



Slack-borrowing

Slack-passing results from level sensitivity of latches and takes place if: $T_{CLK} < t_{pd,A} + t_{pd,B}$ and the logic functions correctly.



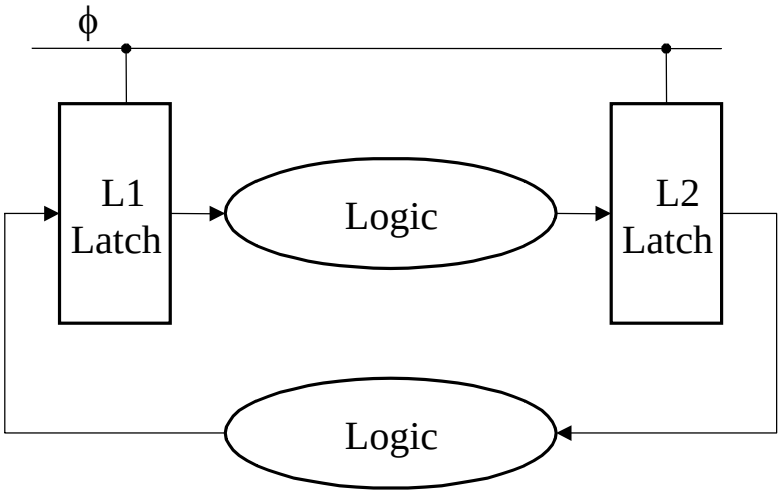


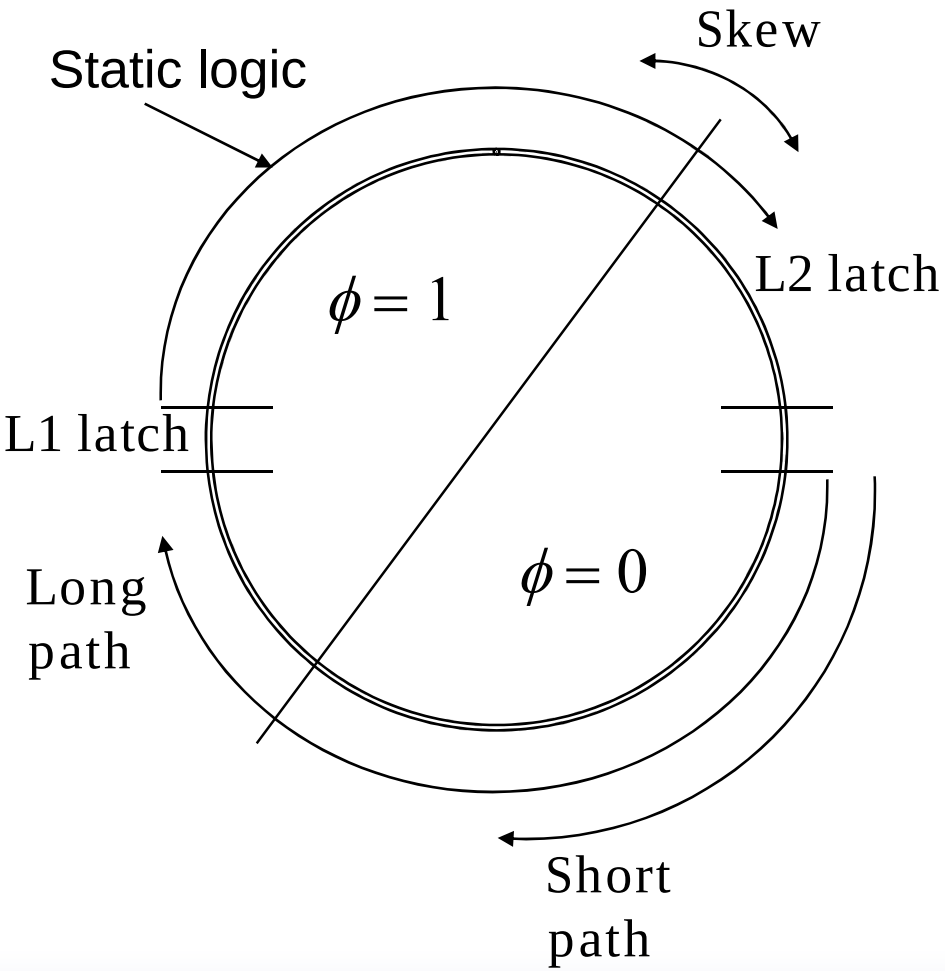
Max. time that can be borrowed = $T_{CLK}/2$

Max. logic cycle delay = $1.5XT_{CLK}$

"a" valid well before edge 2, since previous block did not use its entire time allotment, producing slack time denoted by the shaded area. Now CLB_B starts computing using the slack provided by CLB_A, and completes before its allocated time (edge 4) and passes a small slack to the next cycle.

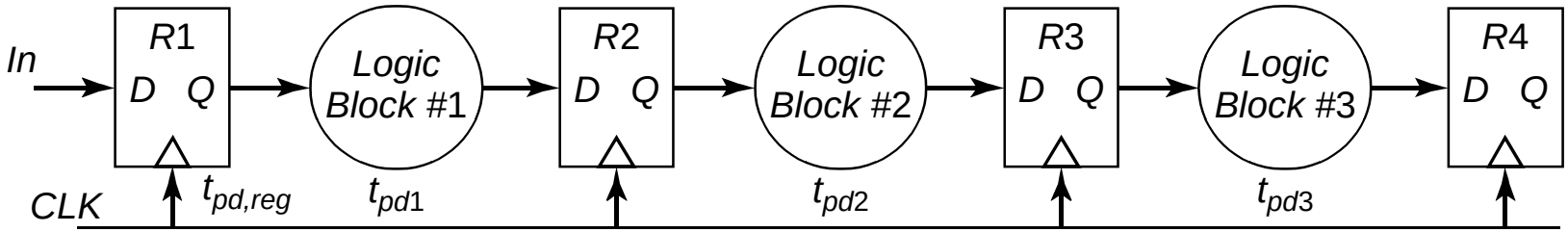
Latch-Based Timing





Can tolerate skew!

Self-timed and Asynchronous Design



Pipelined synchronous datapath

Functions of clock in synchronous design

1) Acts as completion signal: ensures that physical timing constraints are met

2) Ensures the correct ordering of events

Problems: skew, jitter, noise, system performance determined by slowest segment in pipeline

Alternatives to Synchronous Design

Truly asynchronous design

1) Completion is ensured by careful timing analysis

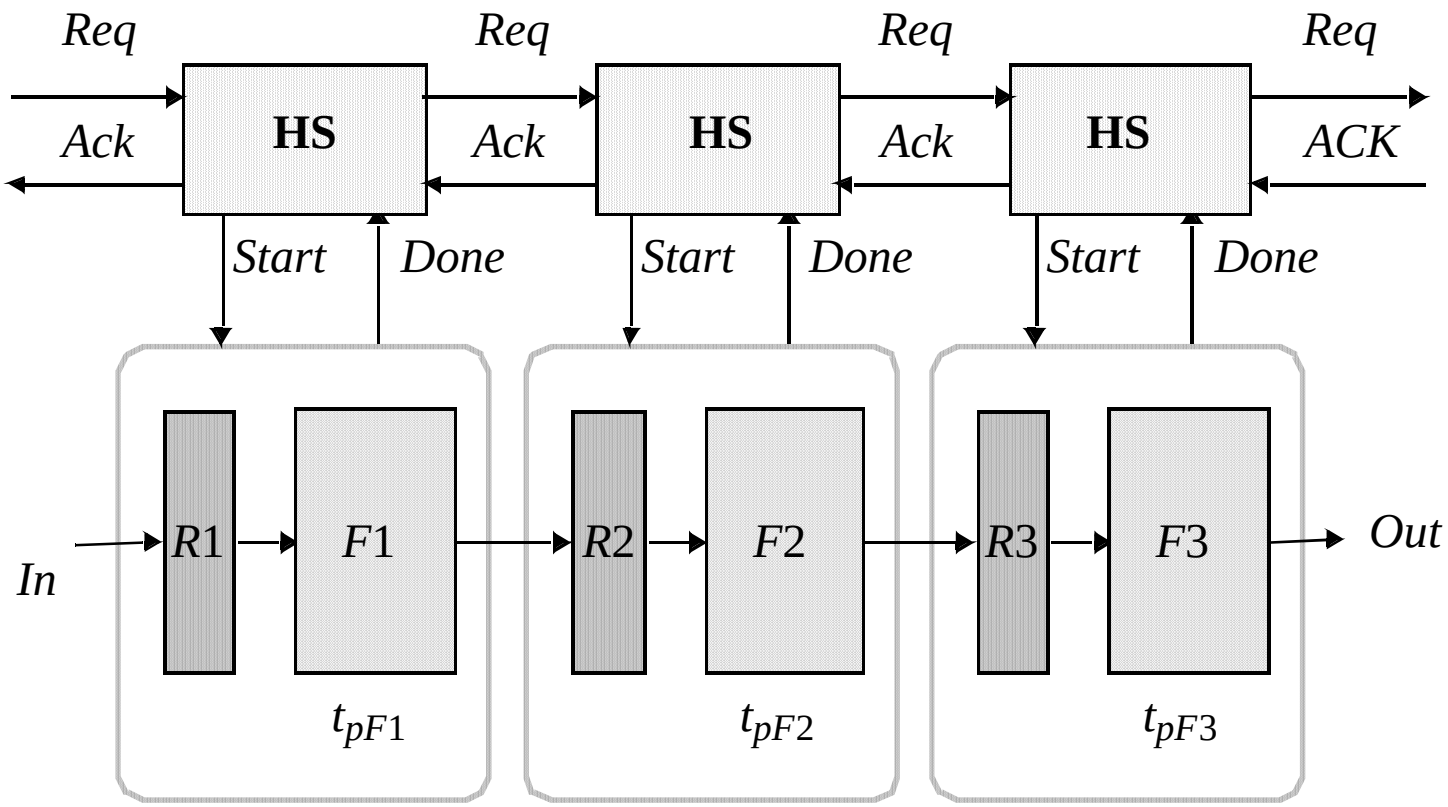
2) Ordering of events is implicit in logic

Self-timed design

1) Completion ensured by completion signal

2) Ordering imposed by handshaking protocol

Self-Timed Pipelined Datapath



The Handshaking protocol is used to communicate between adjacent combinational function that a particular computation has been completed

Self-Timed Designs

Features:

- Separates the physical and logical ordering functions implied in circuit timing
- The completion signal *Done* ensures that physical timing constraints are met
- Logical ordering of the operation is ensured by the *ack-req* scheme or the *HS* protocol
- Choice of protocol significantly impacts performance and robustness

Advantages:

- Timing signals generated locally—avoids overheads associated with distributing high-speed clocks
- Separating physical and logical ordering mechanisms results in potential increase in performance. In syn. systems the T_{CLK} must be large enough to account for the worst case delay
- Robust agaianst temperature variations

Cons:

- Substantial circuit-level overhead

Completion Signal Generation: Dual Rail Coding

Requires introduction of redundancy in the data representation in order to signal that a particular bit is in either a transition or a steady-state mode

Power dissipation can be a problem!

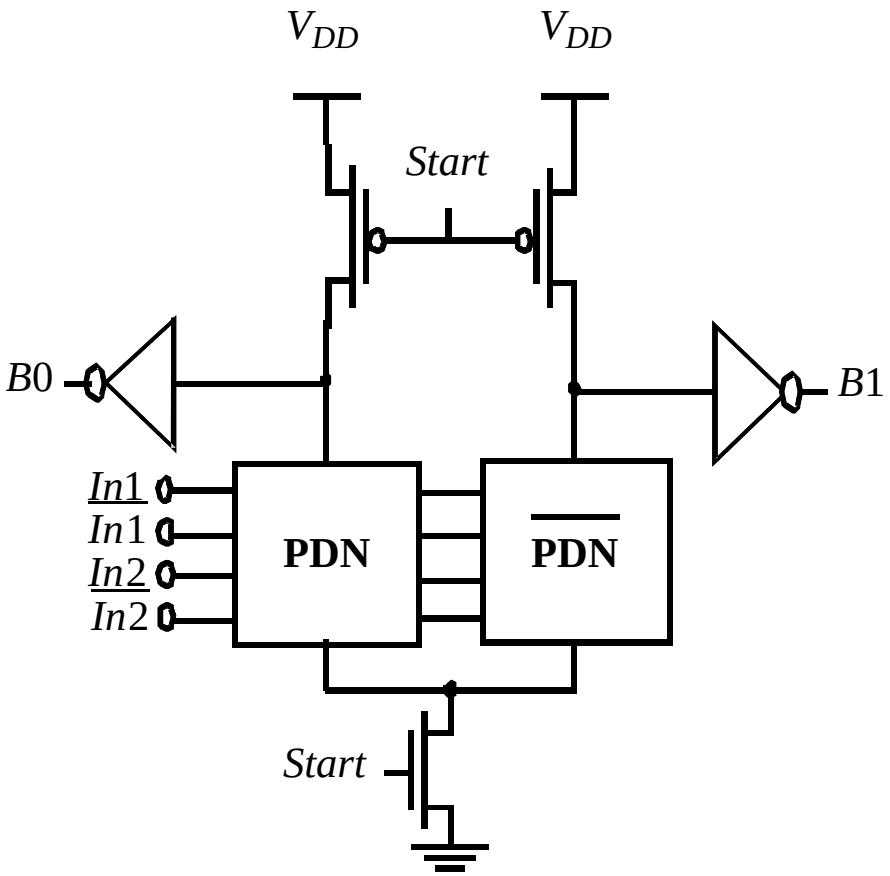
B	B0	B1
in transition (or reset)	0	0
0	0	1
1	1	0
illegal	1	1

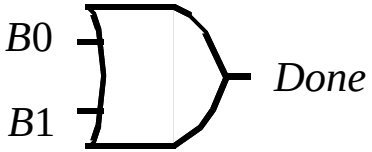
Two bits B0 and B1 are used to represent a single data bit B

Using Redundant Signal Encoding

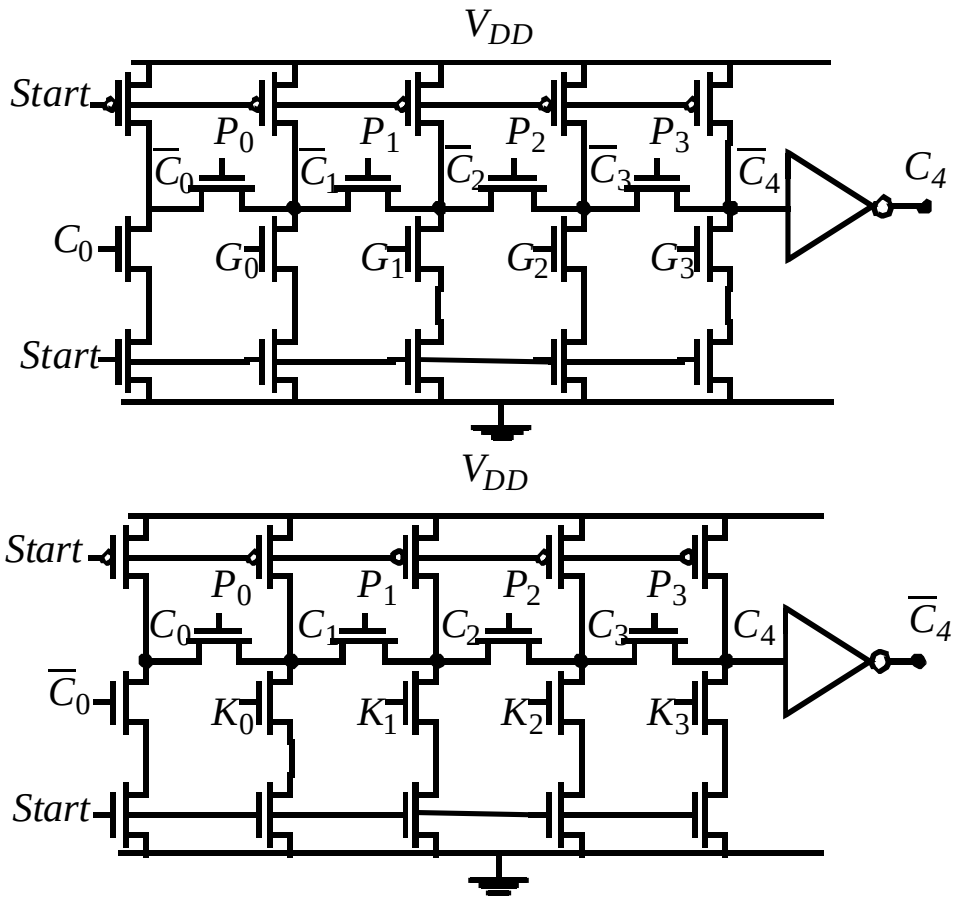
Key idea in all redundant signal representations is to capture the transition state to indicate that the circuit is in evaluation mode and the output data are not valid

Dual Rail Coding Using Dynamic DCVSL

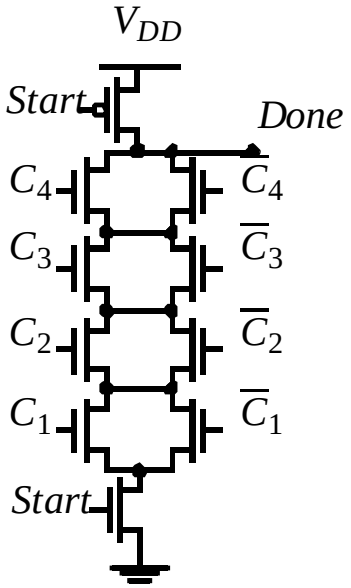




Example: A Self-Timed Adder

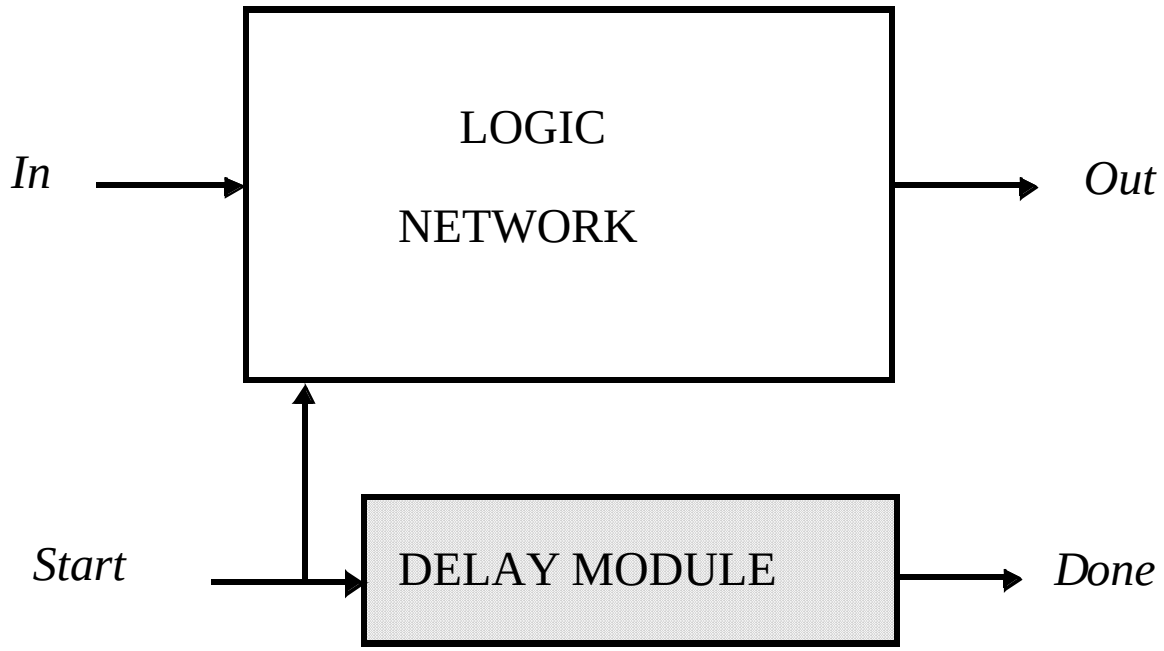


(a) Differential carry generation



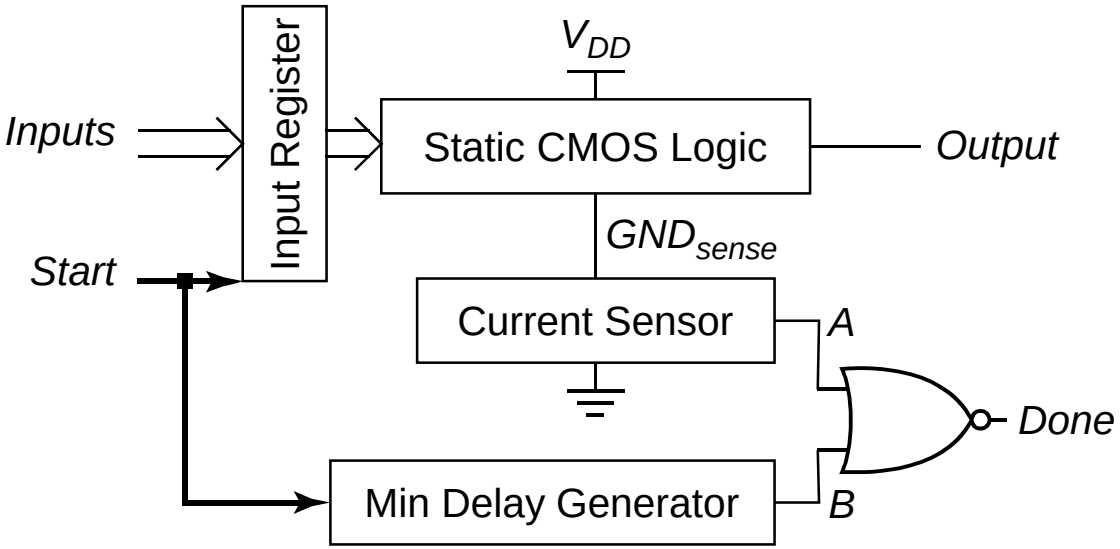
(b) Completion signal

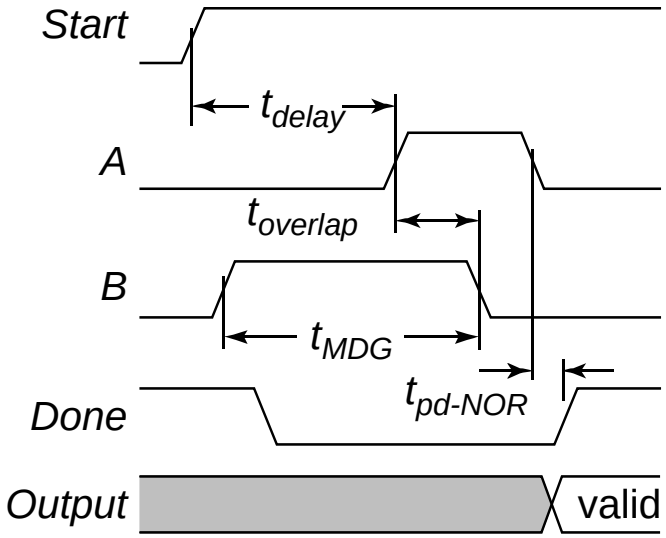
Completion Signal Generation



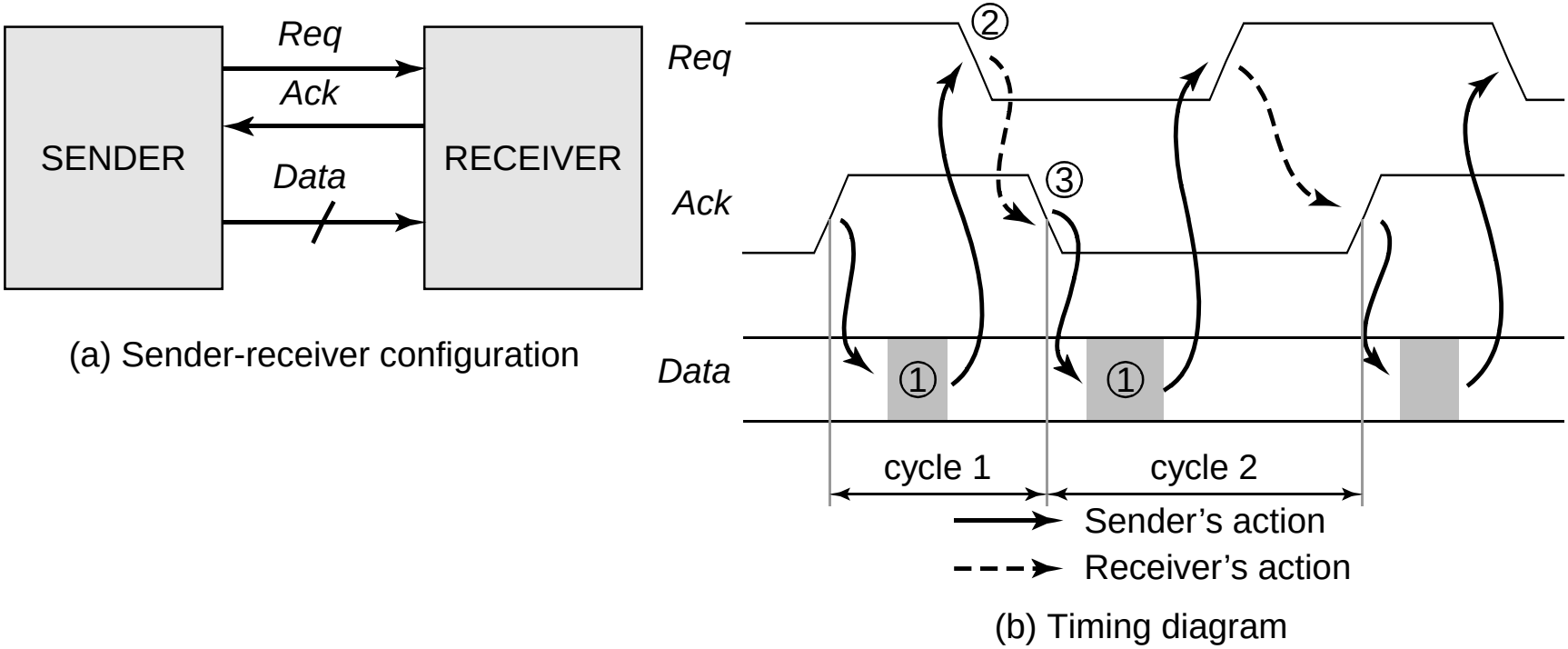
Widely used to generate the internal timing in memories

Completion Signal Using Current Sensing





Hand-Shaking Protocol

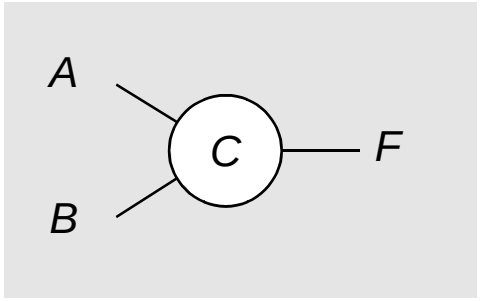


(a) Sender-receiver configuration

(b) Timing diagram

Two Phase Handshake

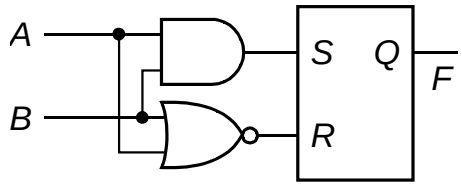
Event Logic – The Muller-C Element



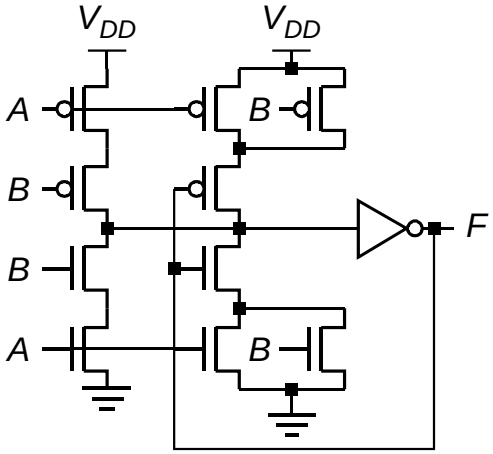
(a) Schematic

A	B	F_{n+1}
0	0	0
0	1	F_n
1	0	F_n
1	1	1

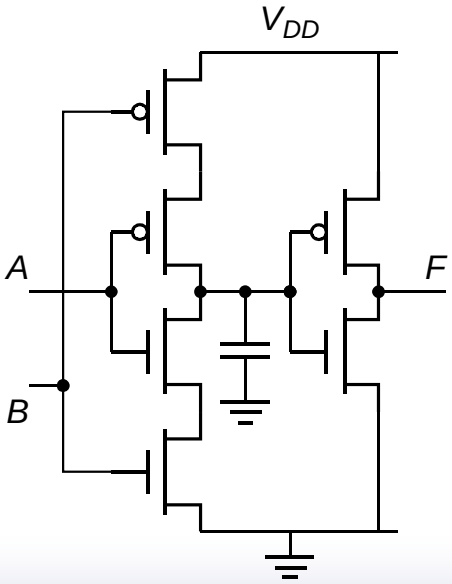
(b) Truth table



(a) Logic

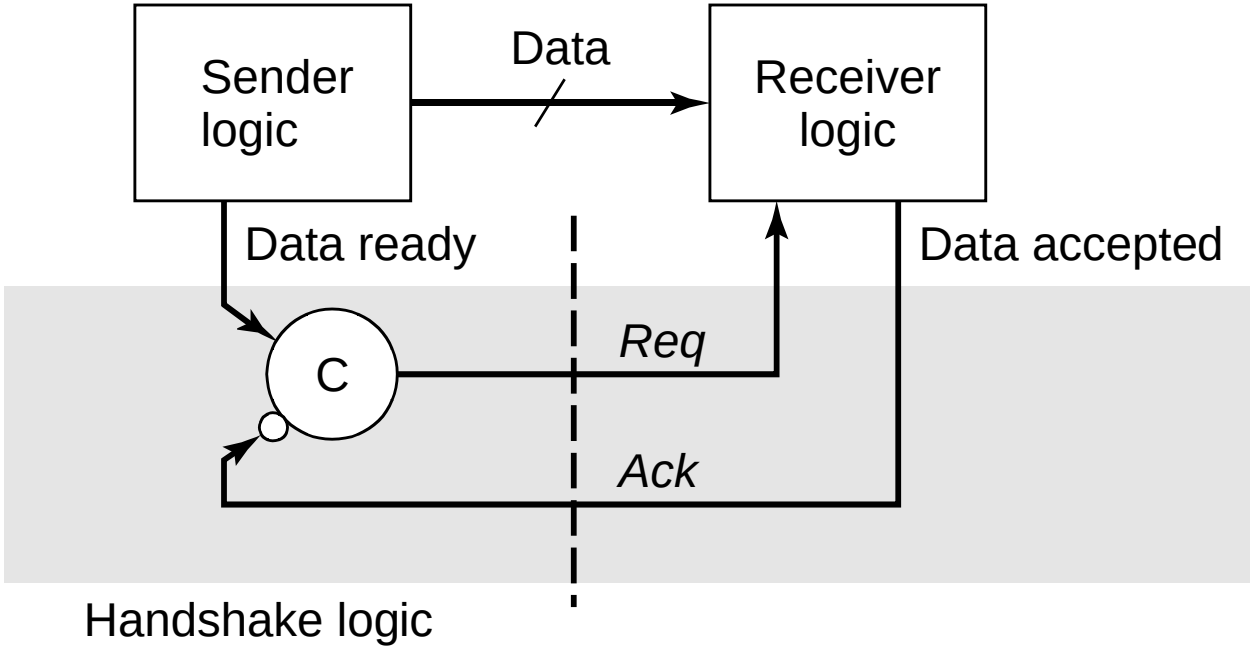


(b) Majority Function



(c) Dynamic

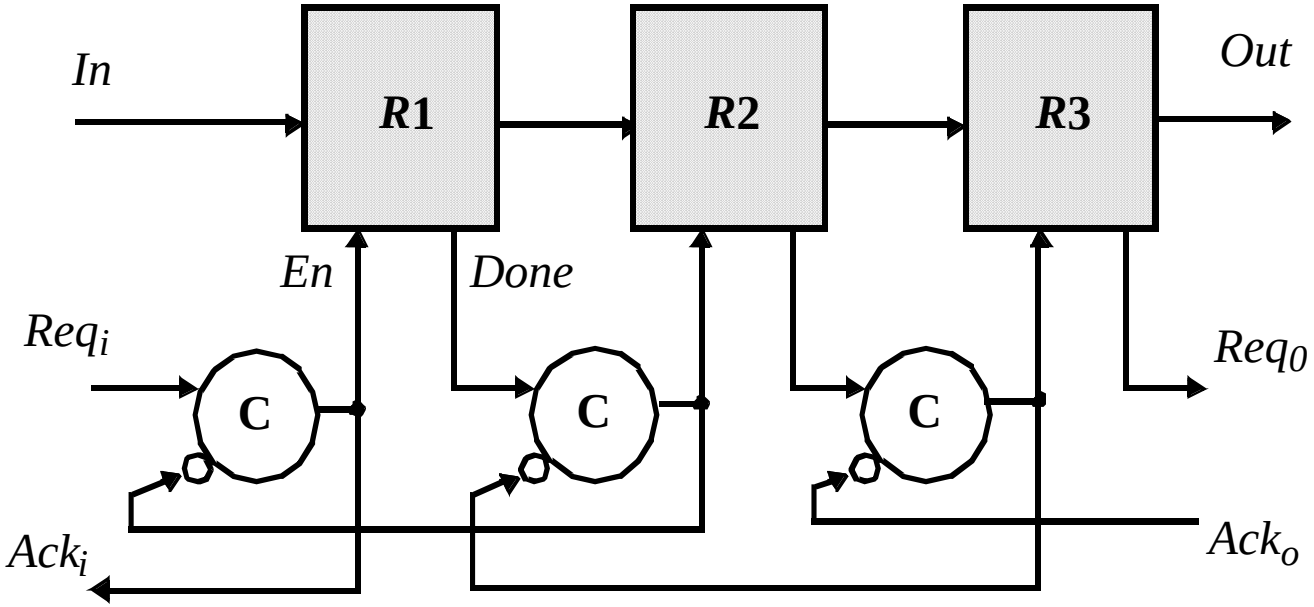
2-Phase Handshake Protocol



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

Disadvantage : edge - sensitive, has state

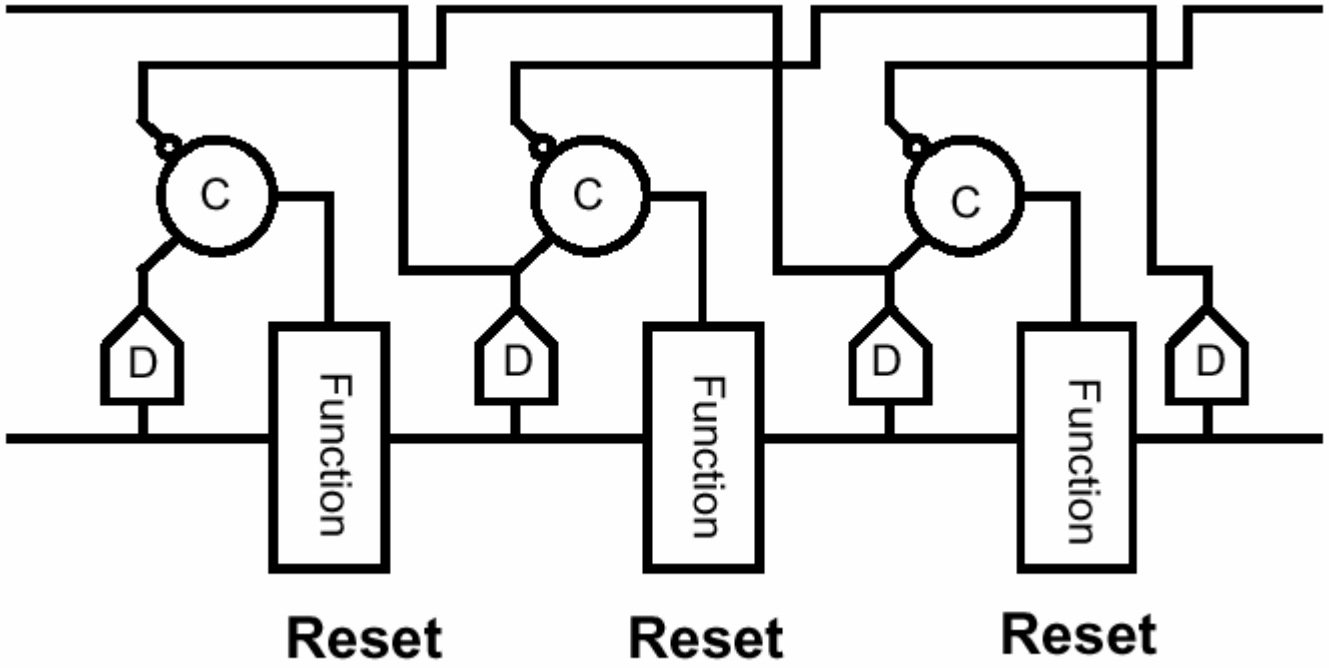
Example: Self-timed FIFO



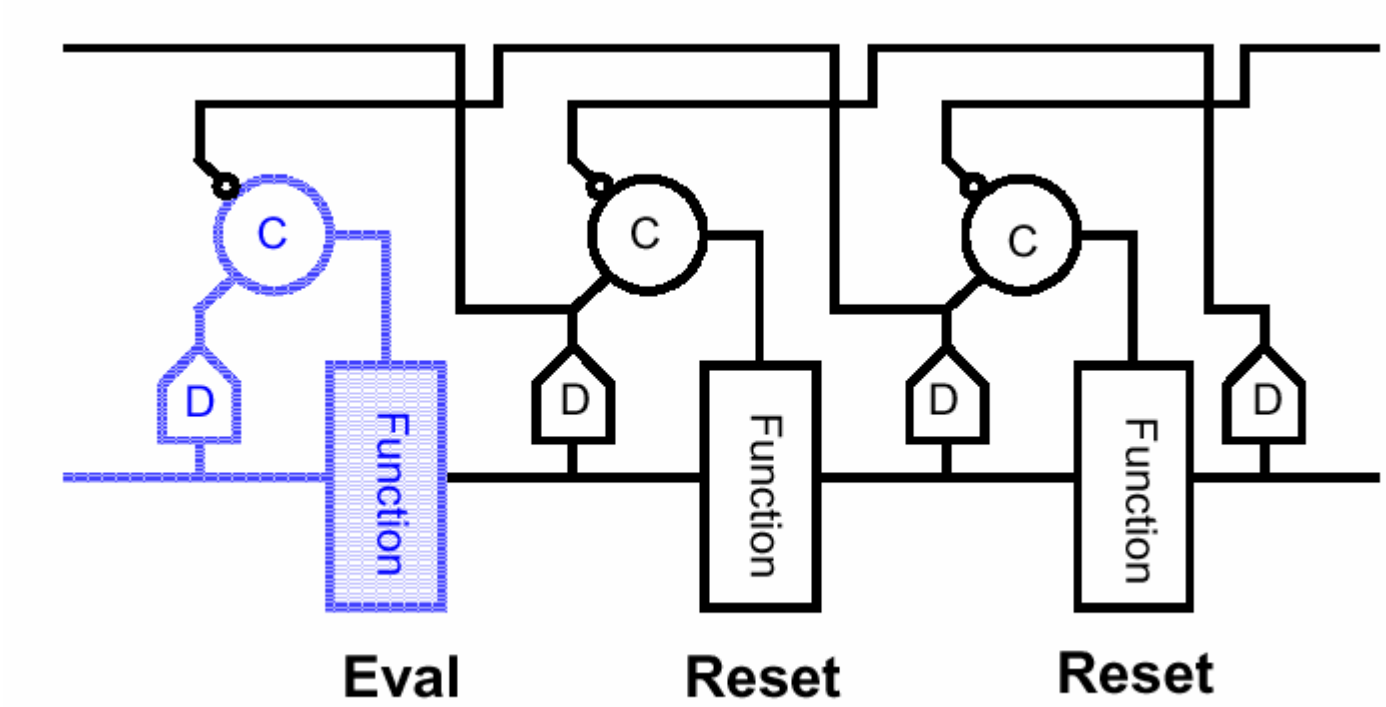
All 1s or 0s -> pipeline empty

Alternating 1s and 0s -> pipeline full

2-Phase Protocol

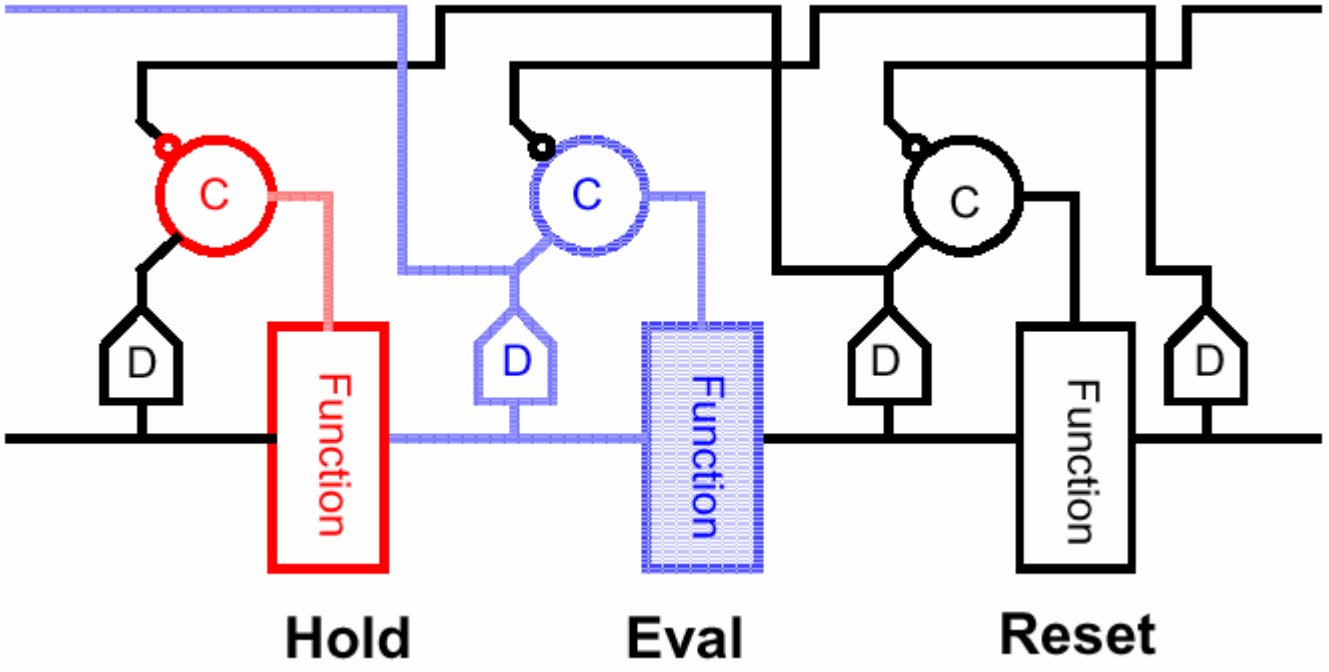


Example

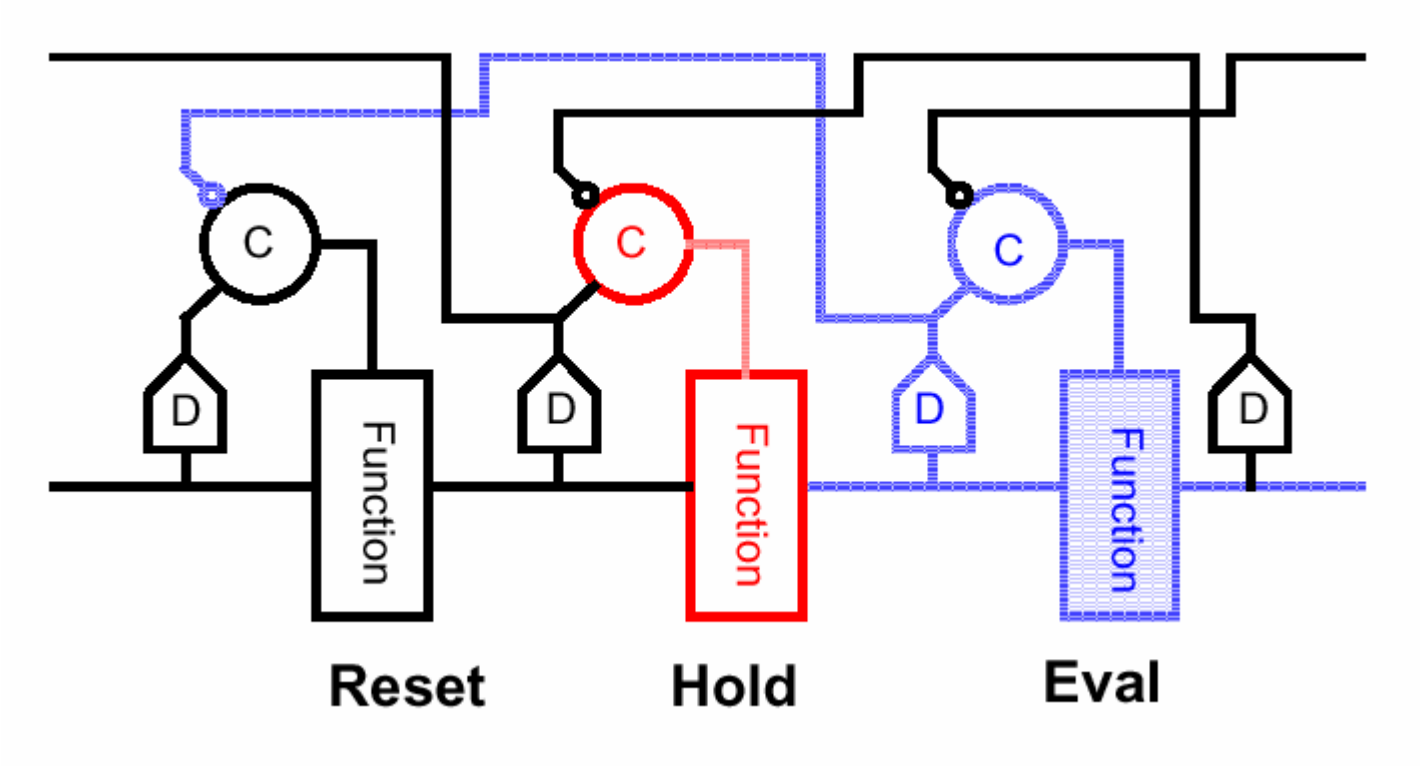


From [Horowitz]

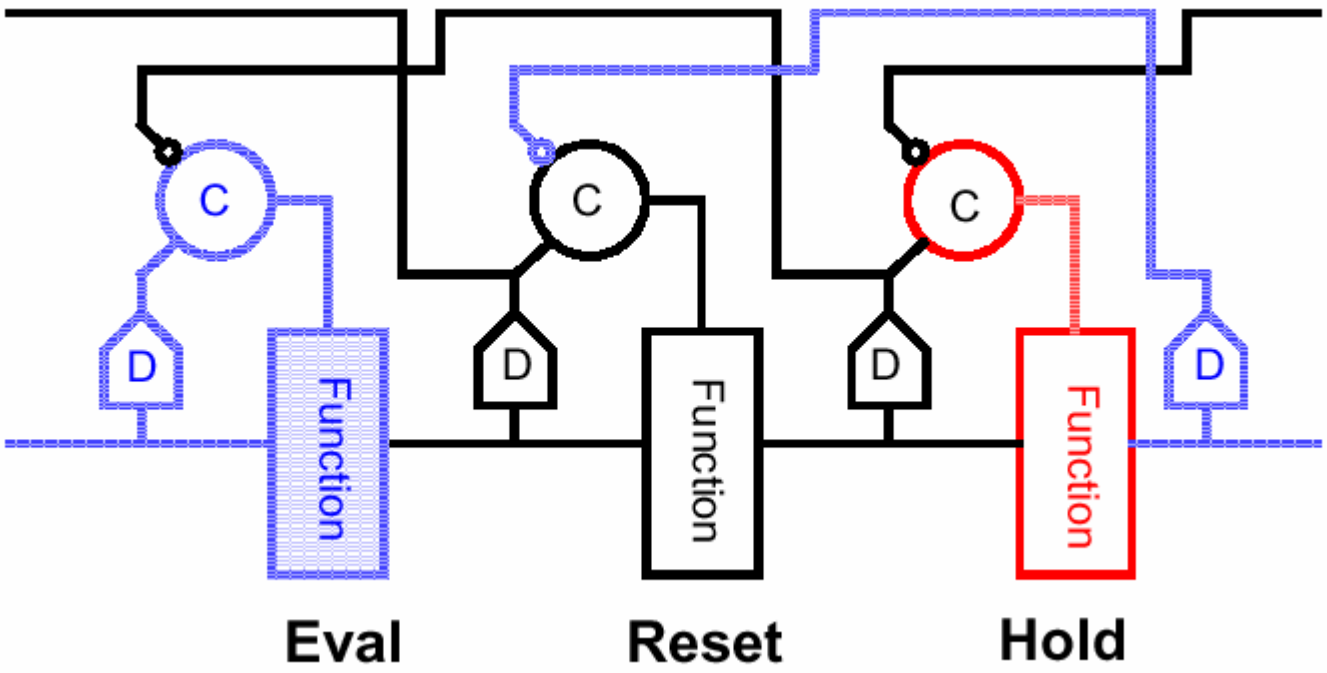
Example



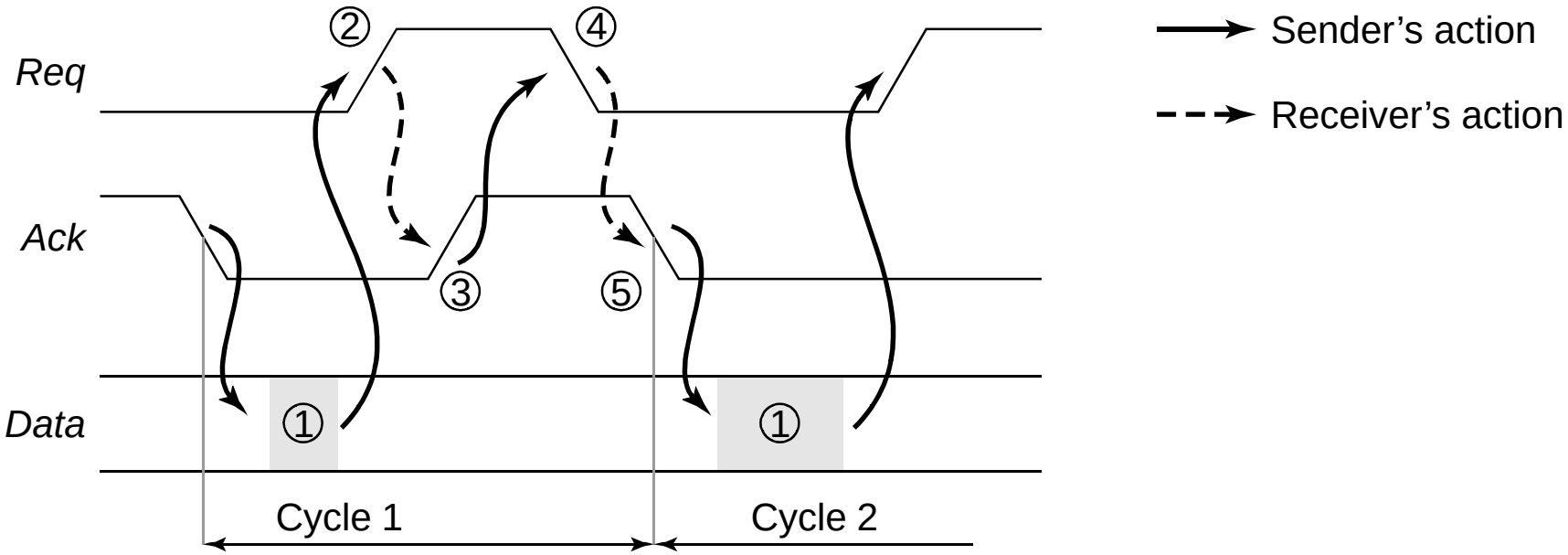
Example



Example



4-Phase Handshake Protocol

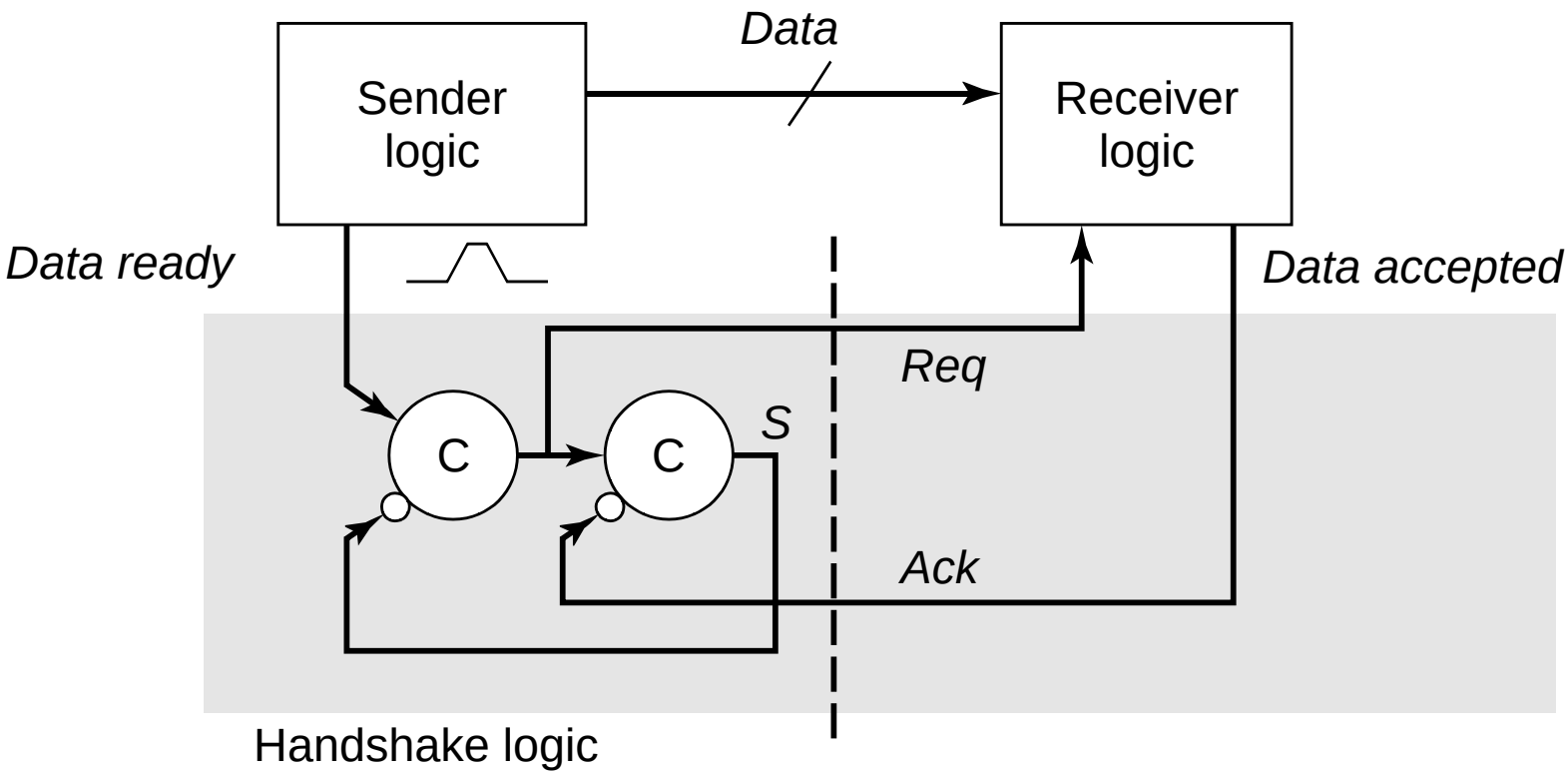


Also known as RTZ

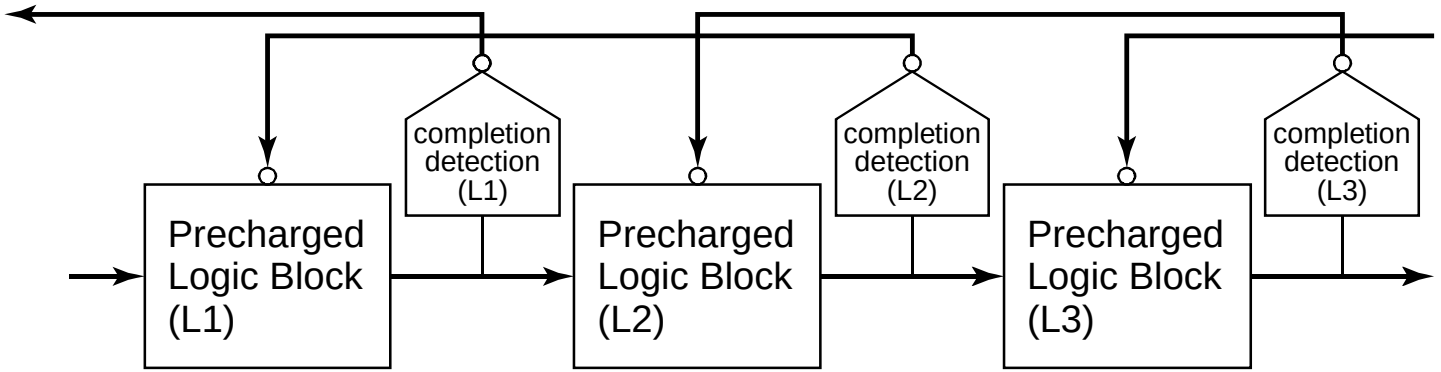
Slower, but unambiguous

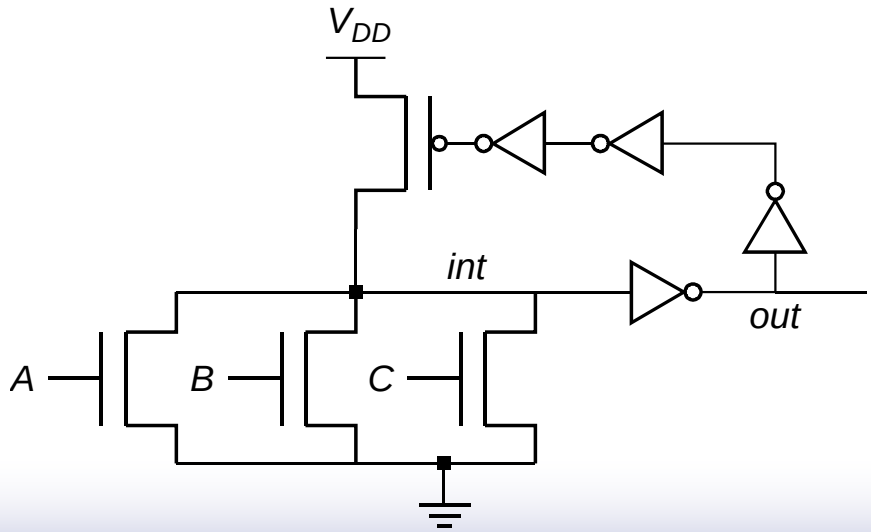
4-Phase Handshake Protocol

Implementation using Muller-C elements



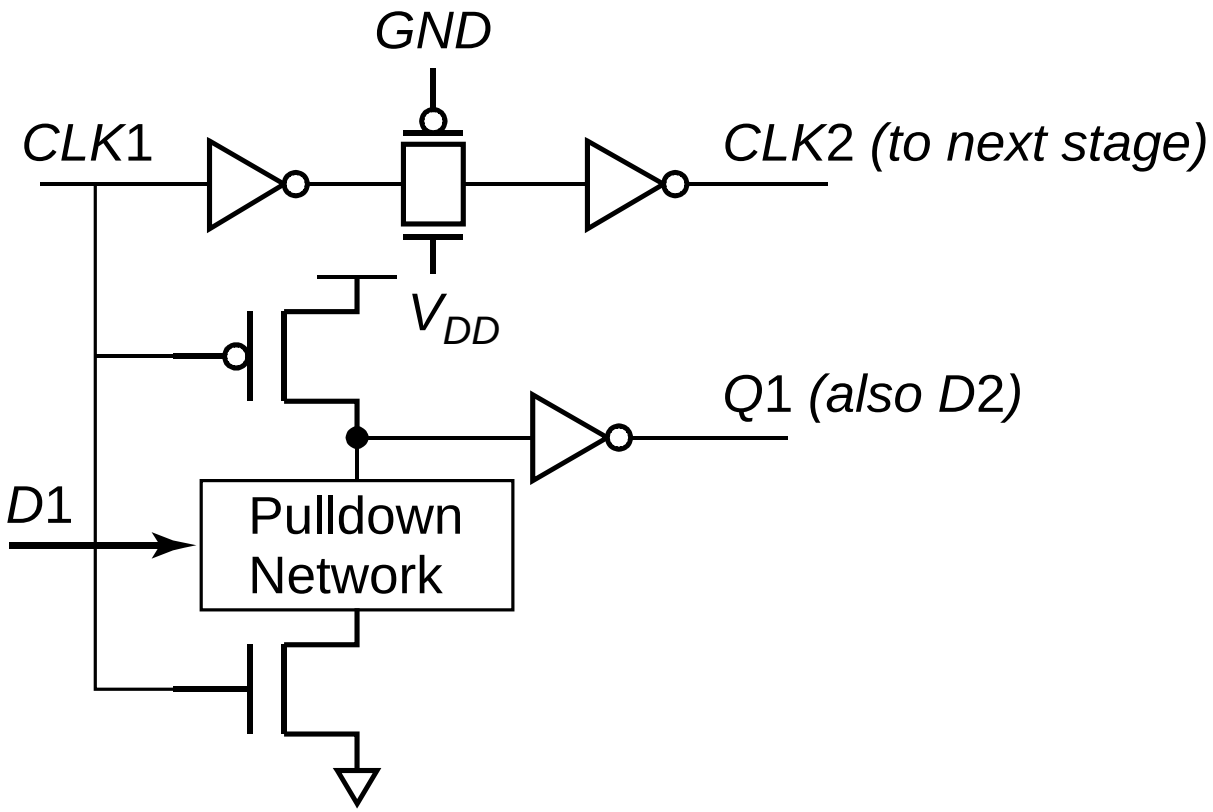
Self-Resetting Logic



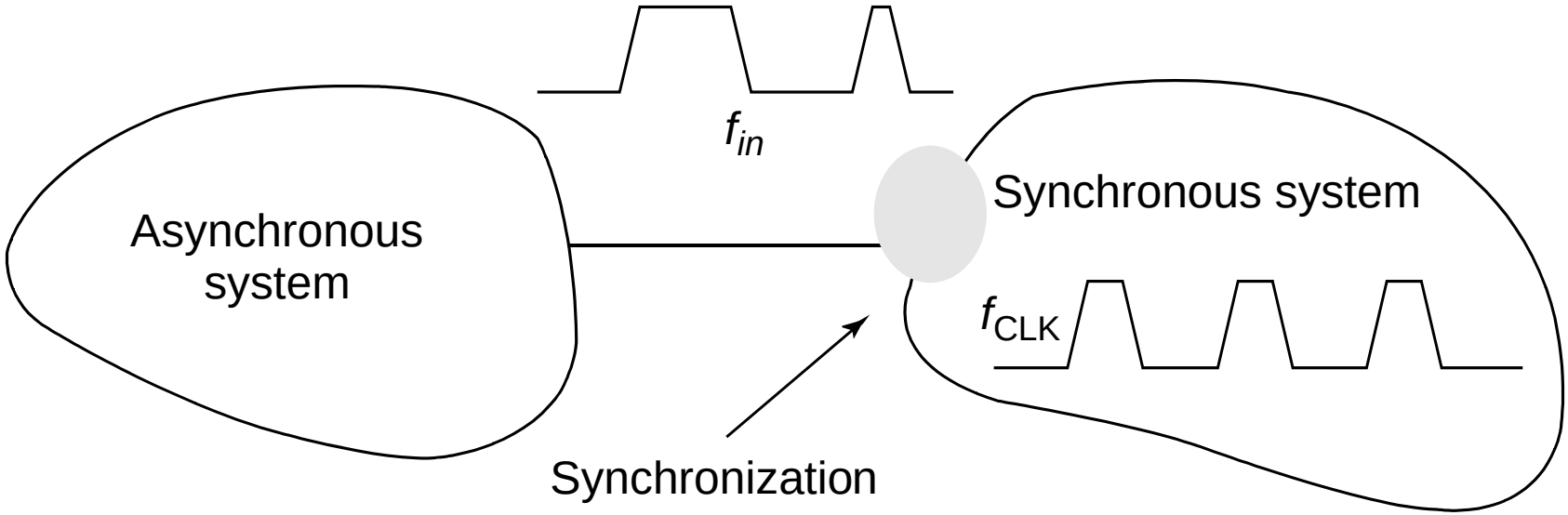


Post-charge logic

Clock-Delayed Domino



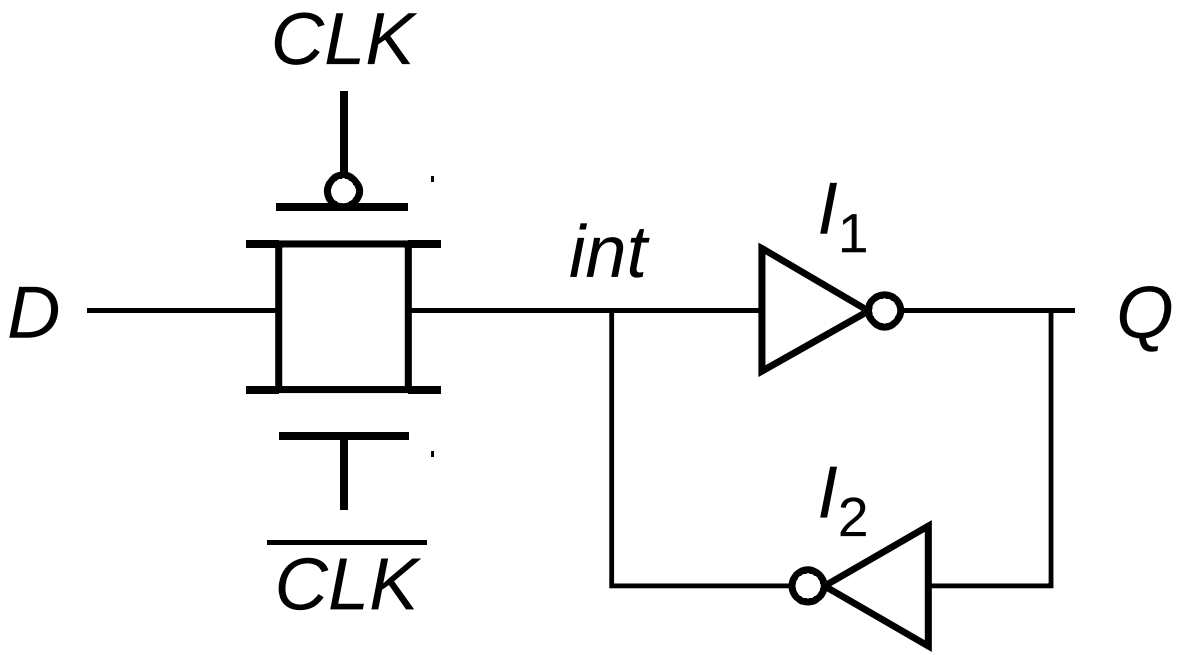
Asynchronous-Synchronous Interface



Synchronizers and Arbiters

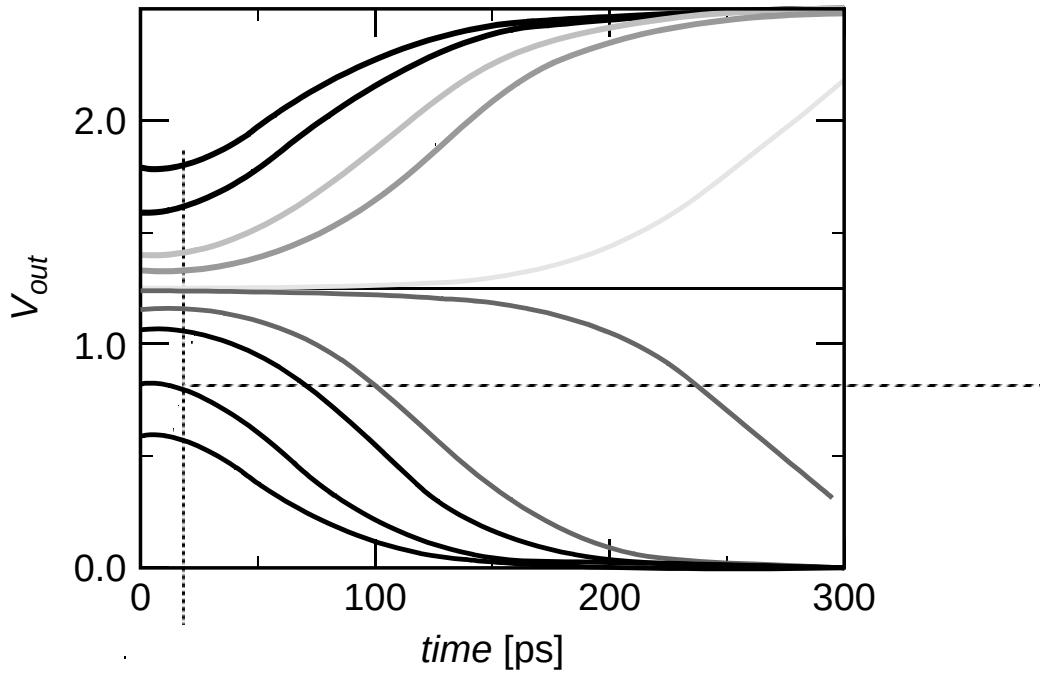
- Arbiter: Circuit to decide which of 2 events occurred first
- Synchronizer: Arbiter with clock ϕ as one of the inputs
- Problem: Circuit HAS to make a decision in limited time - which decision is not important
- Caveat: It is impossible to ensure correct operation
- But, we can decrease the error probability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

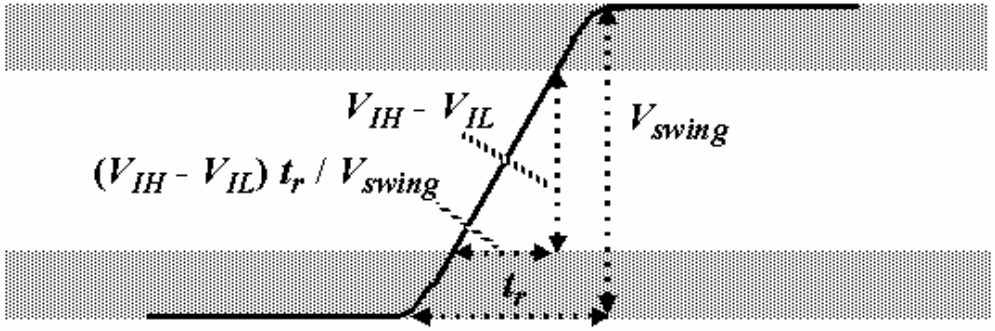
Synchronizer: Output Trajectories



Single-pole model for a flip-flop

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

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

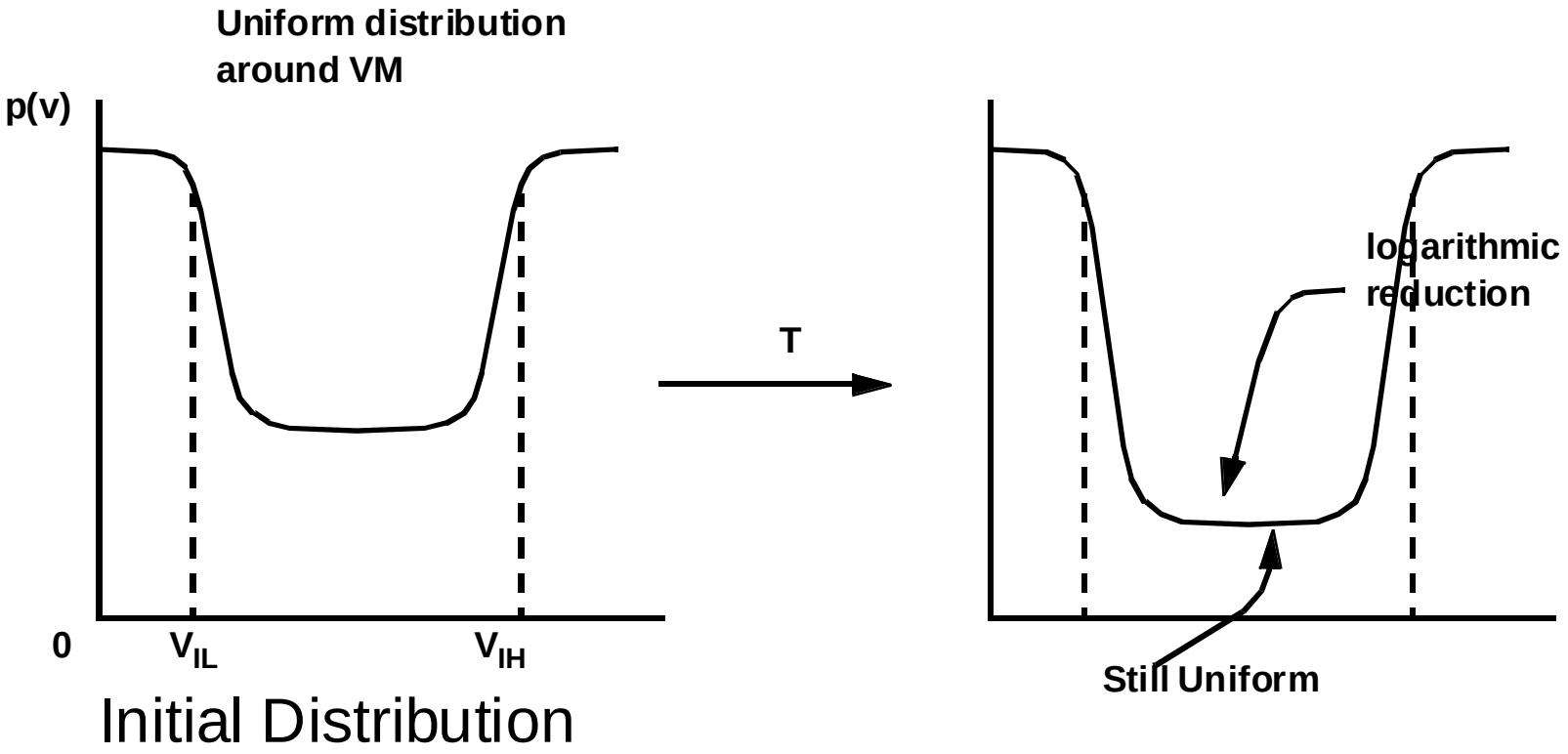
$V_{IH} - V_{IL} = 1$ V ($V_{DD} = 5$ V)

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

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

MTF (0) = 2.5 μsec

Influence of Noise

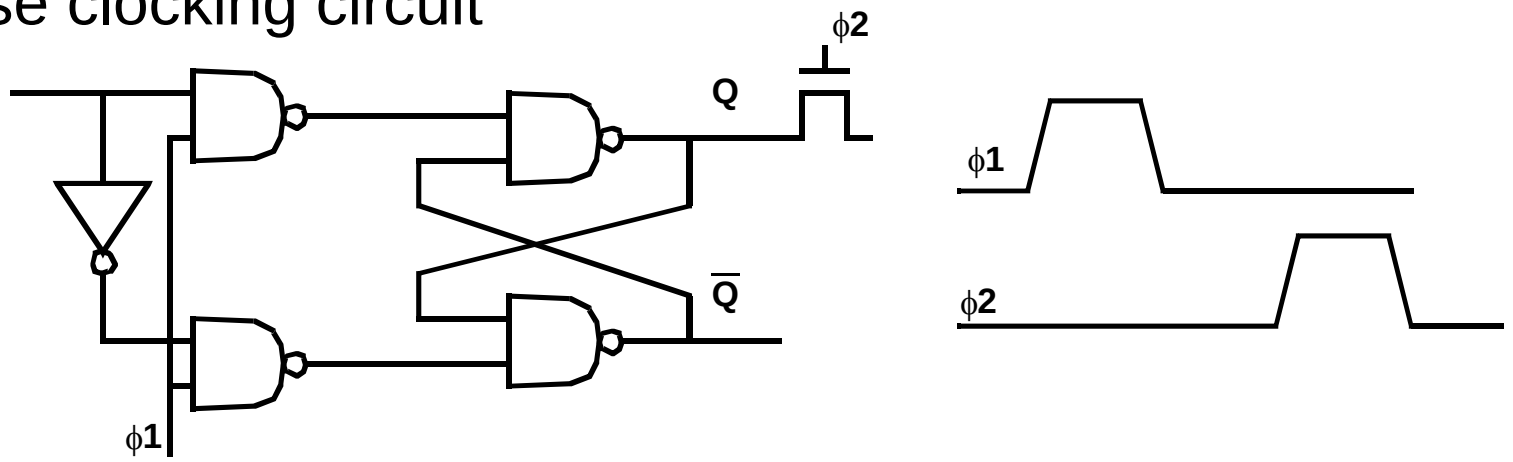


Initial Distribution

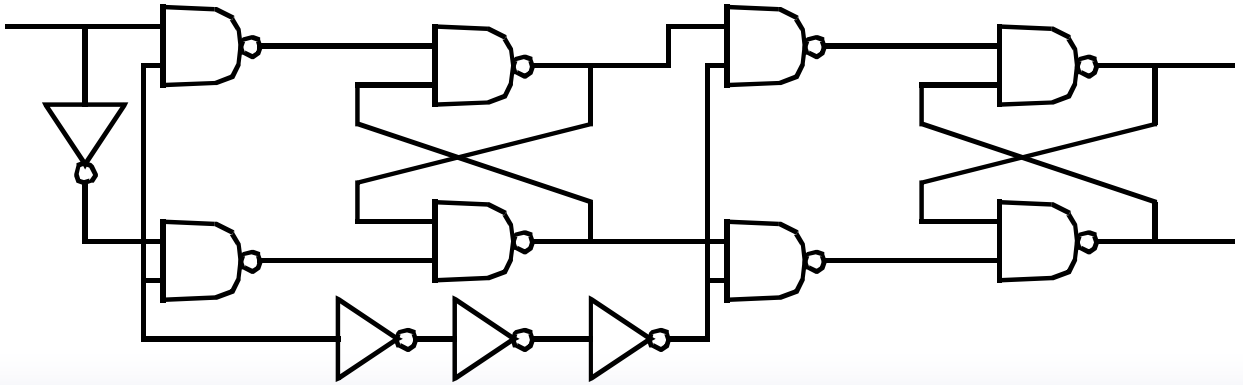
Low amplitude noise does not influence synchronization behavior

Typical Synchronizers

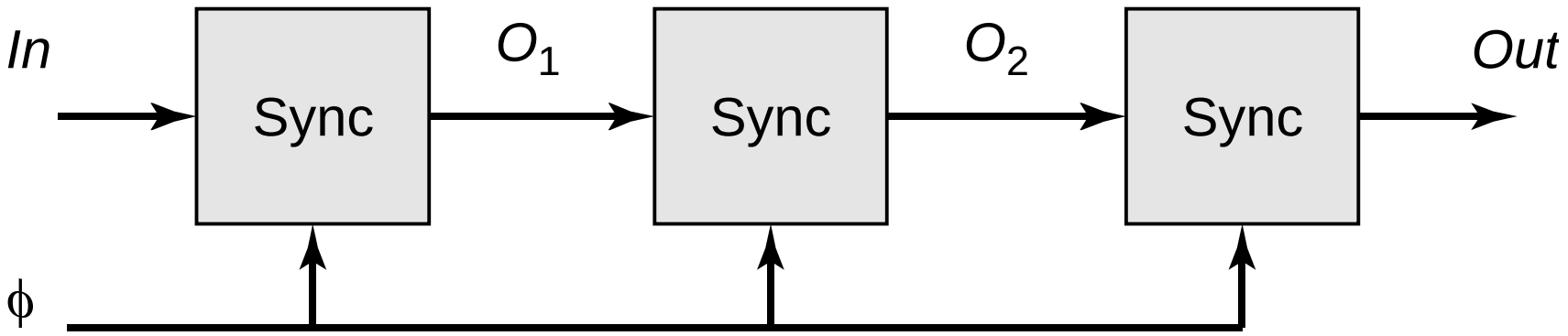
2 phase clocking circuit



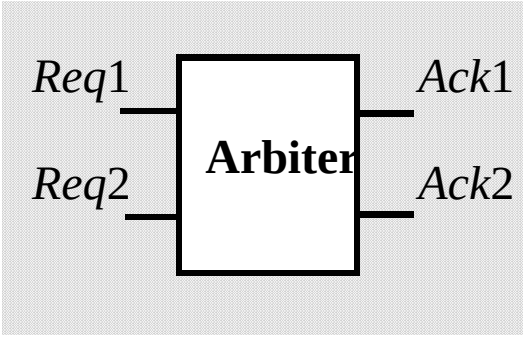
Using delay line



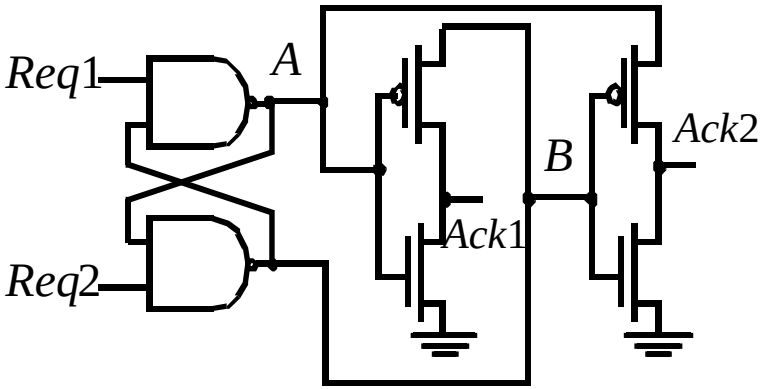
Cascaded Synchronizers Reduce MTF



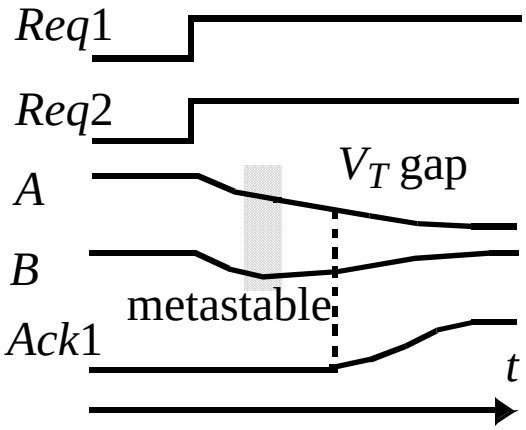
Arbiters



(a) Schematic symbol

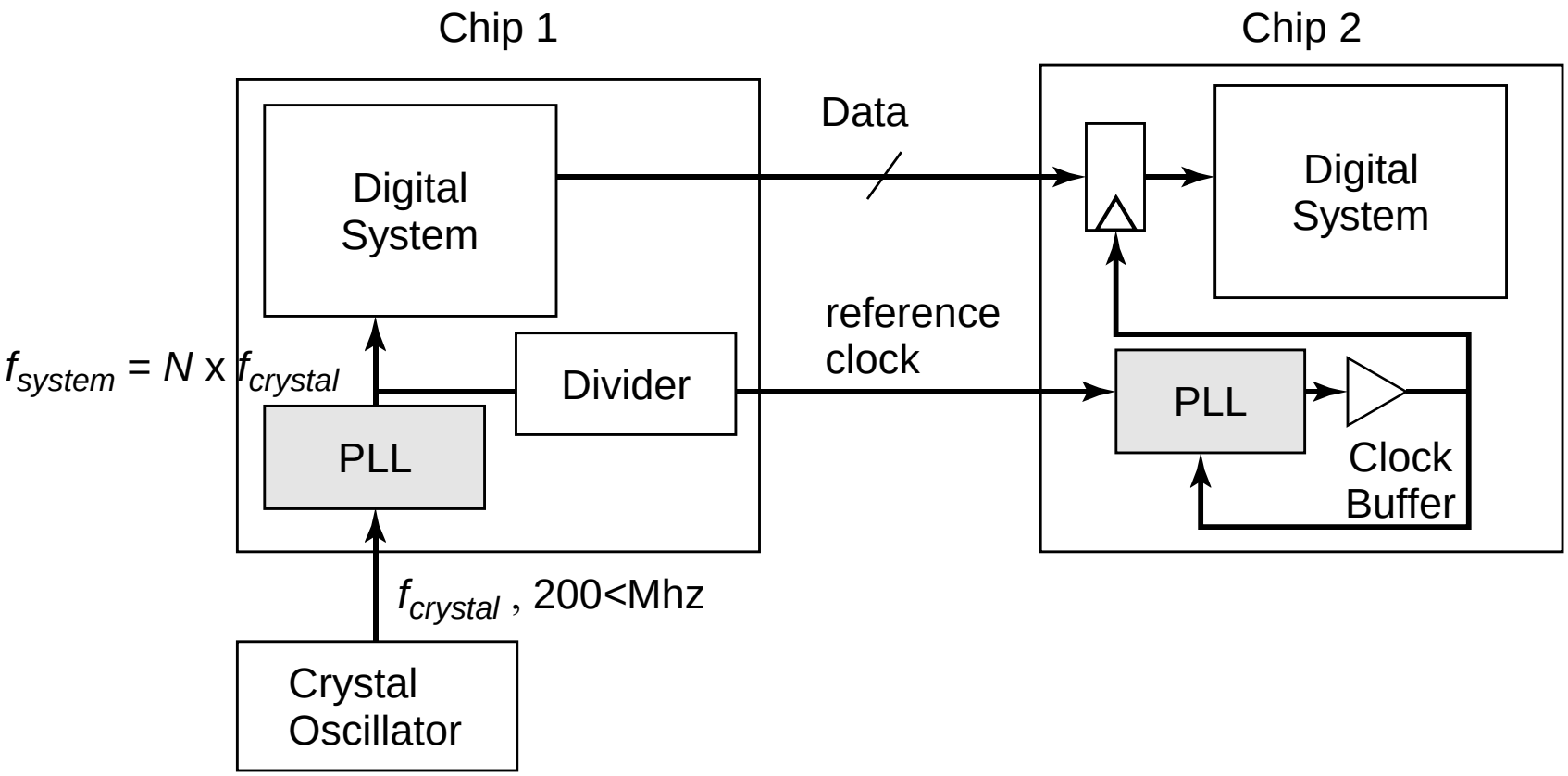


(b) Implementation

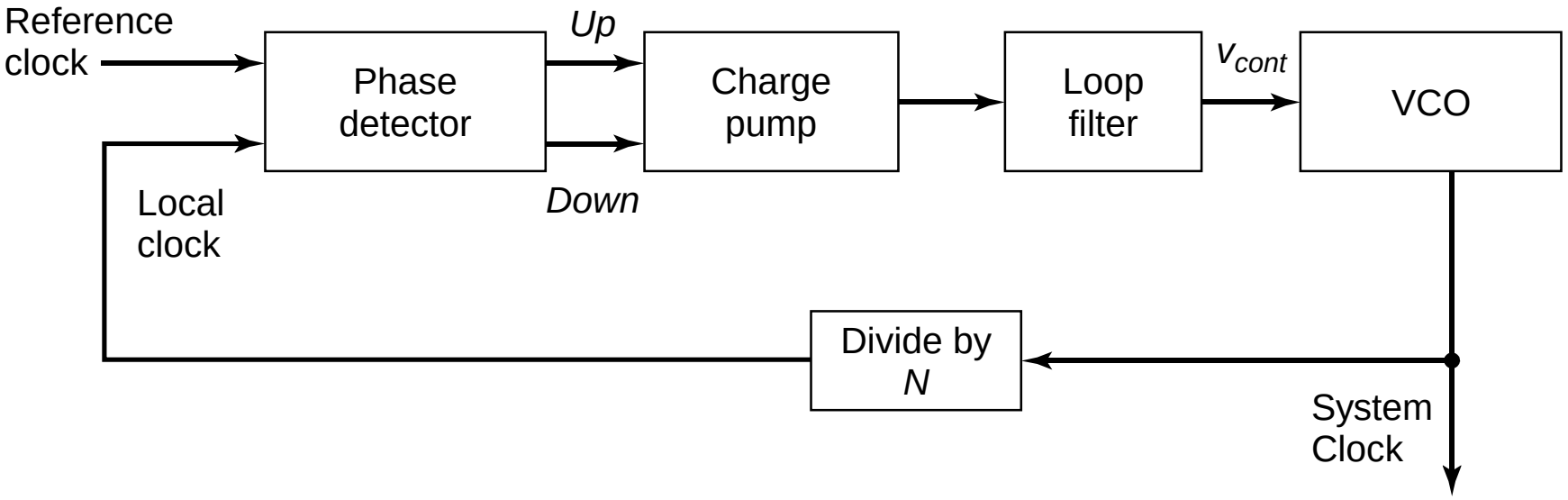


(c) Timing diagram

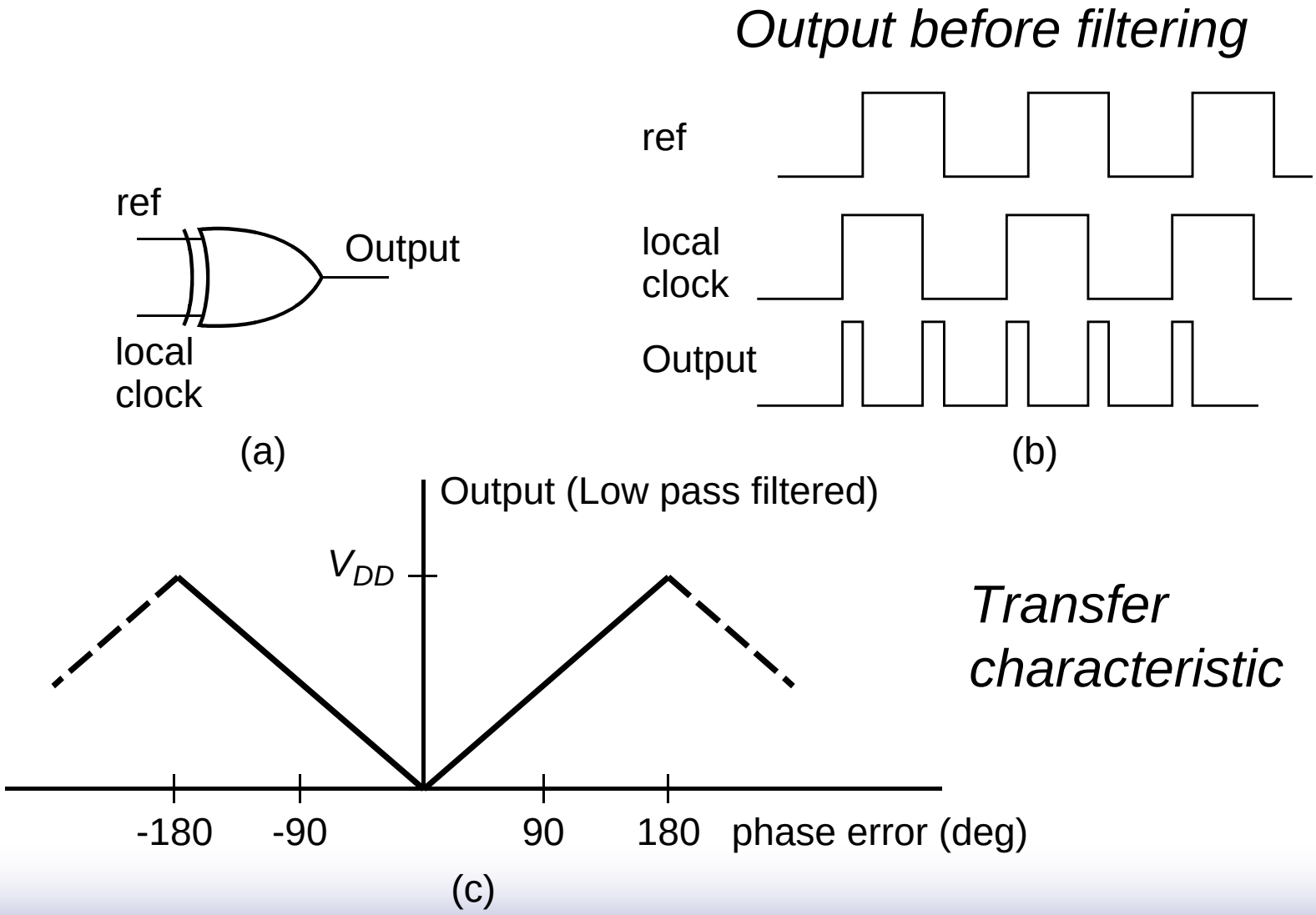
PLL-Based Synchronization



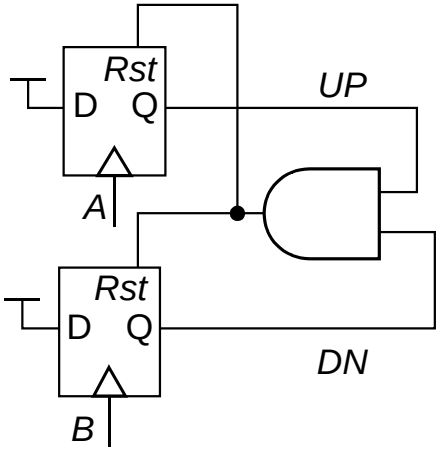
PLL Block Diagram



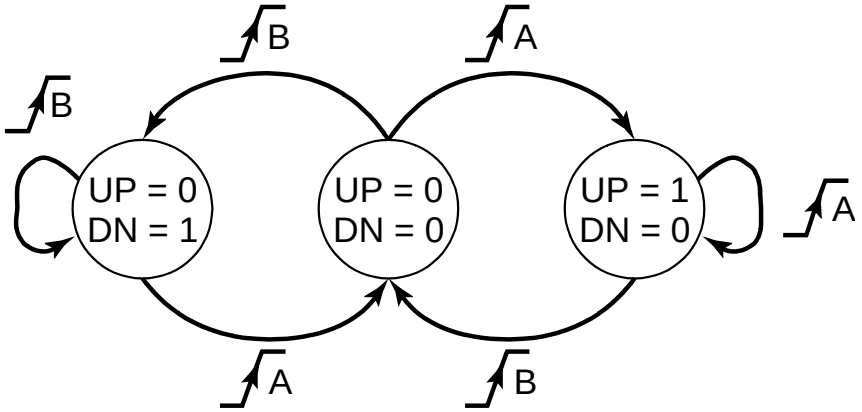
Phase Detector



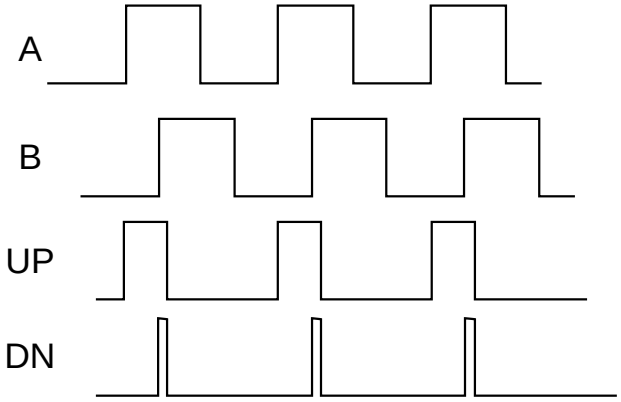
Phase-Frequency Detector

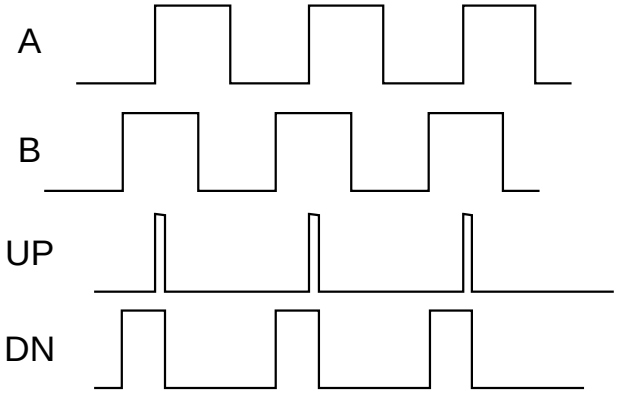


(a) schematic



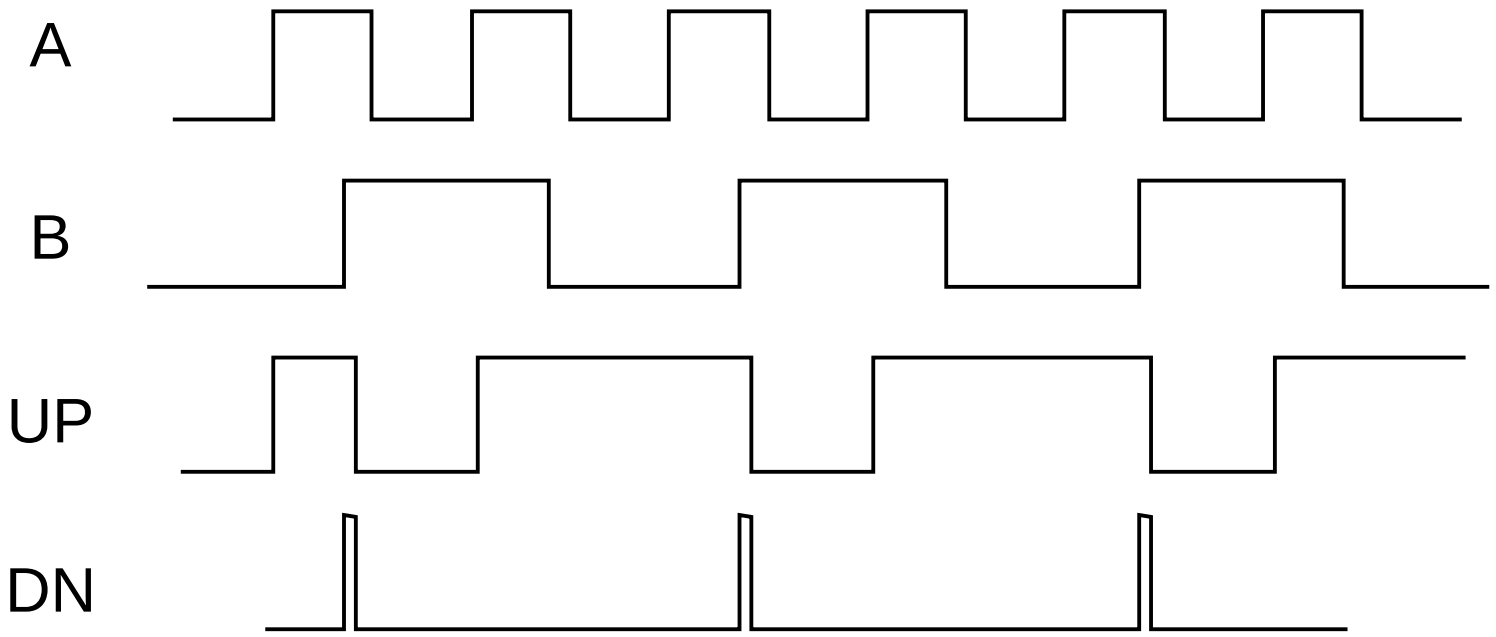
(b) state transition diagram



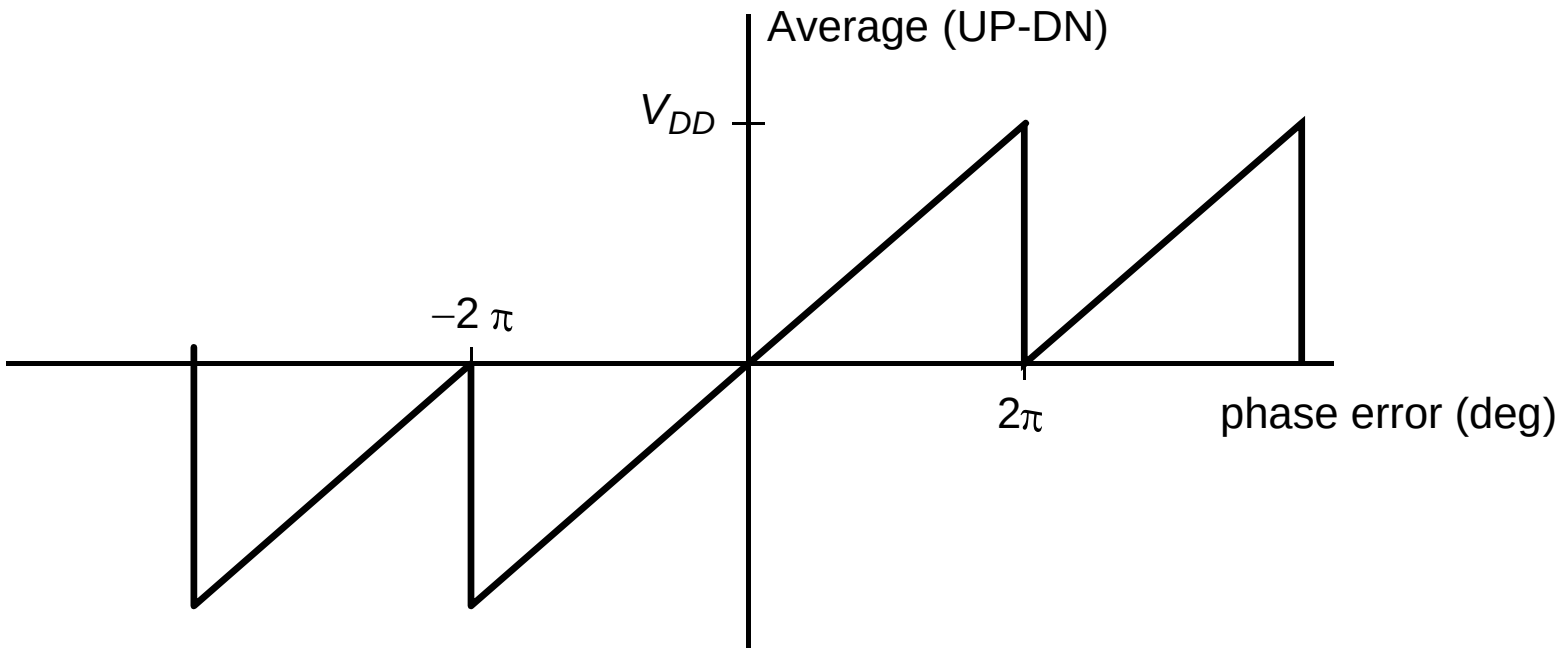


(c) Timing waveforms

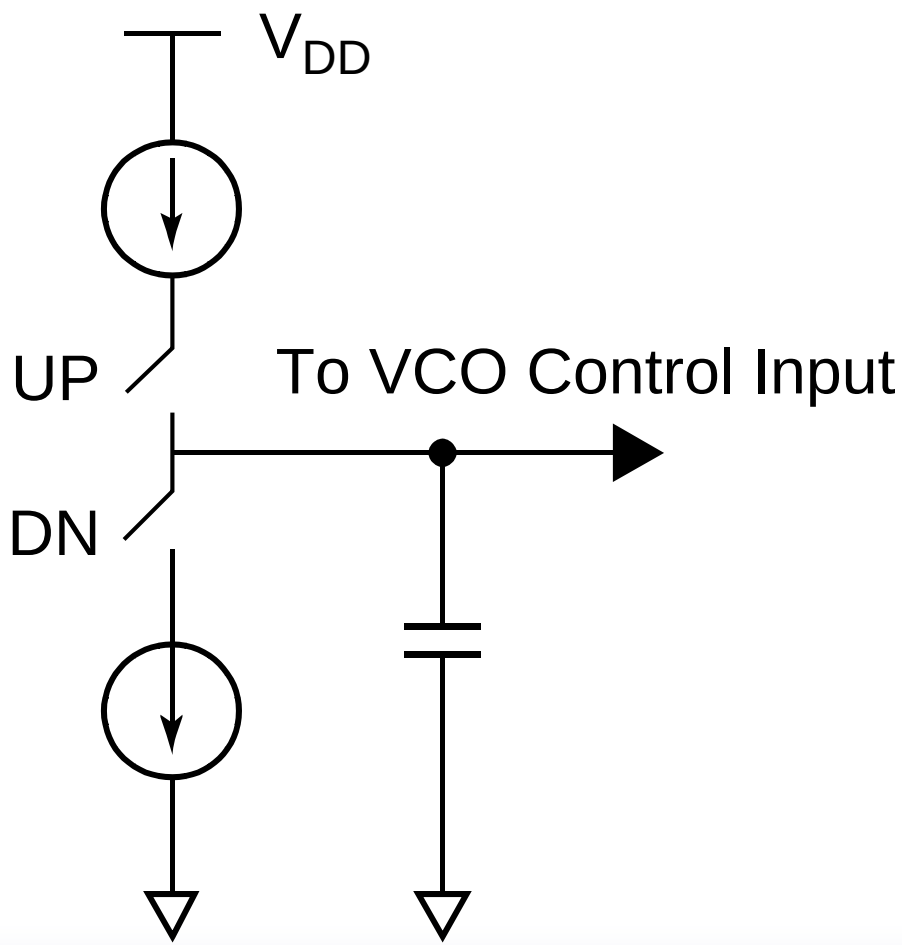
PFD Response to Frequency



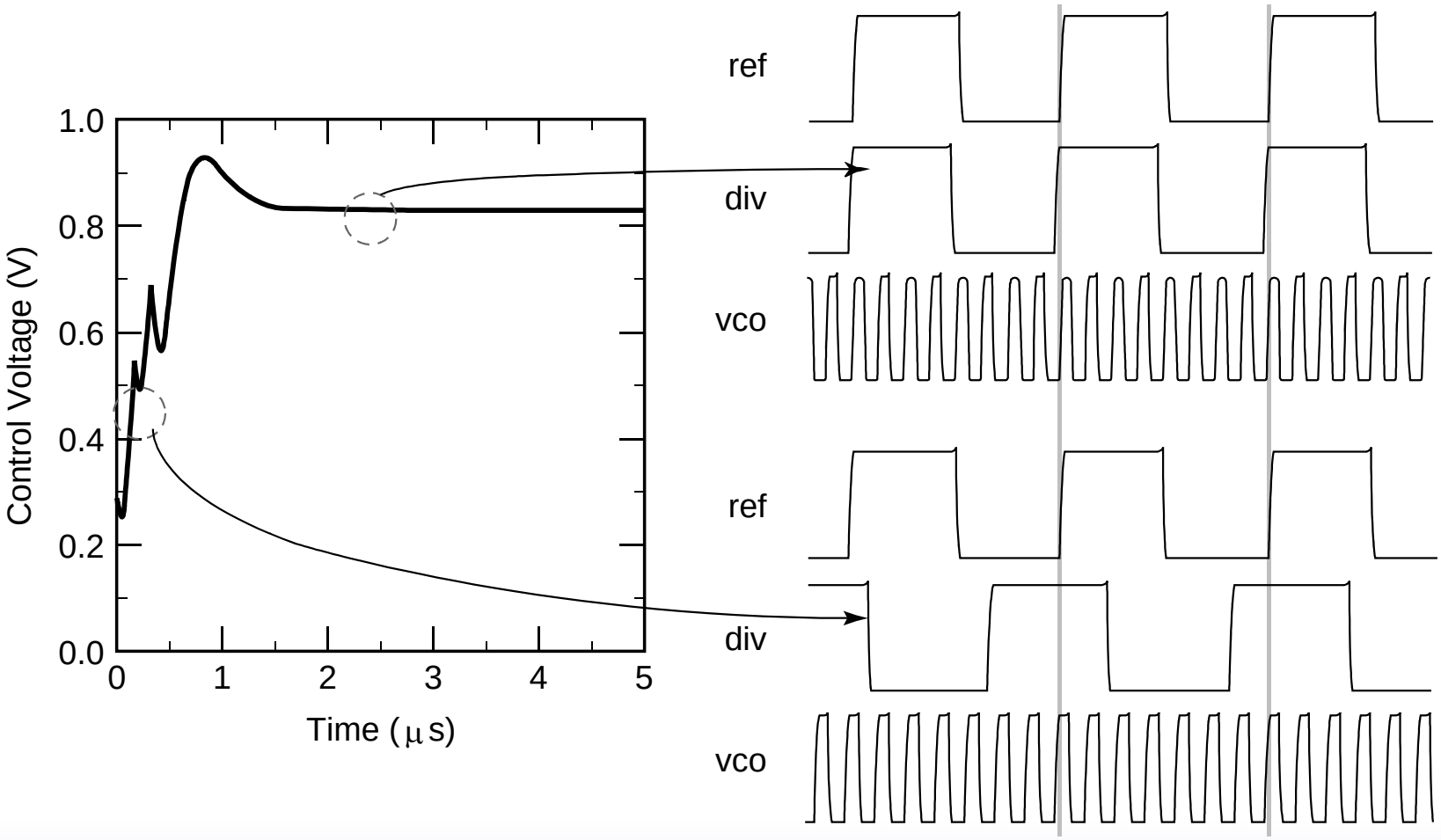
PFD Phase Transfer Characteristic



Charge Pump

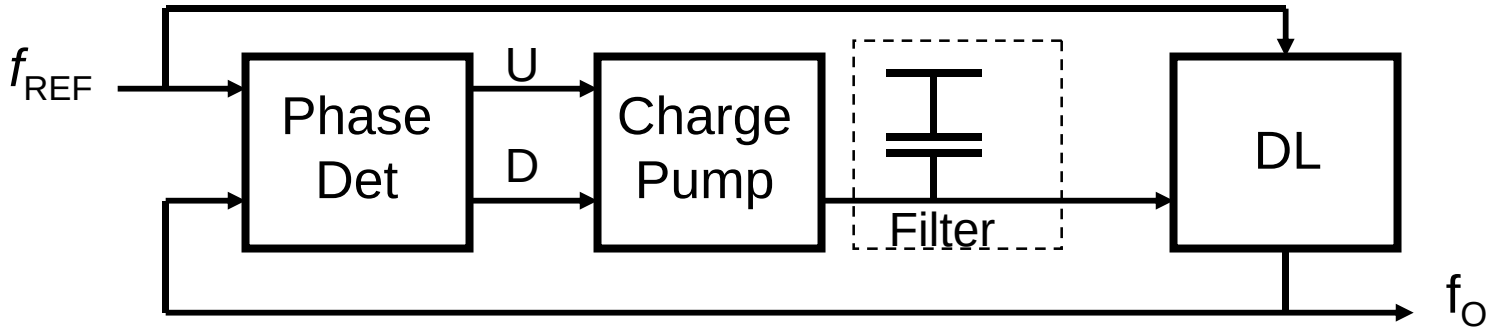


PLL Simulation

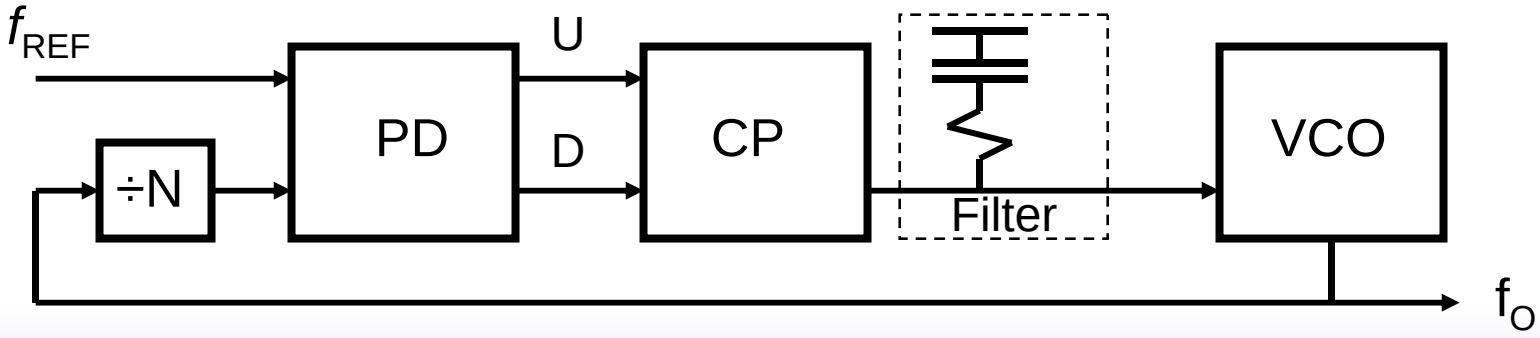


Clock Generation using DLLs

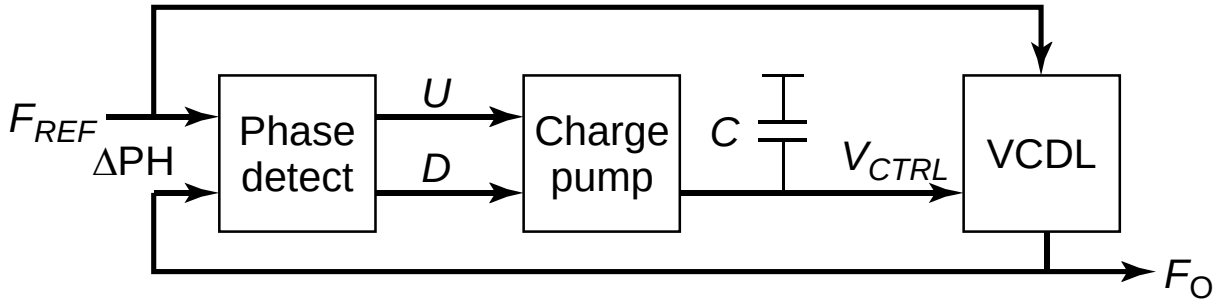
Delay-Locked Loop (Delay Line Based)



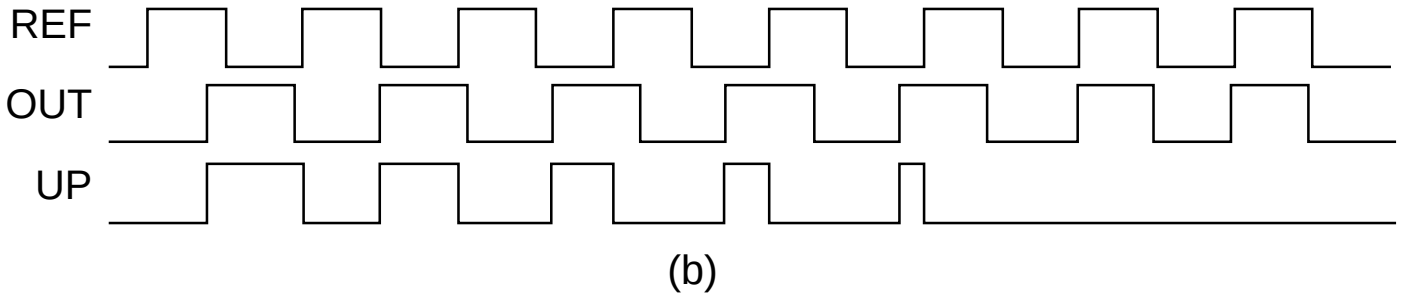
Phase-Locked Loop (VCO-Based)



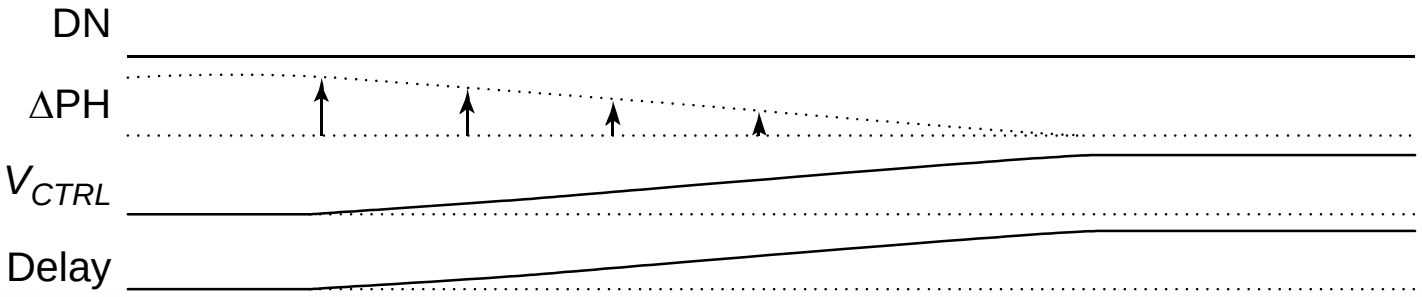
Delay Locked Loop



(a)



(b)



(c)

DLL-Based Clock Distribution