EITF35: Introduction to Structured VLSI Design

Part 4.2.1: Learn More …

Liang Liu
liang.liu@eit.lth.se

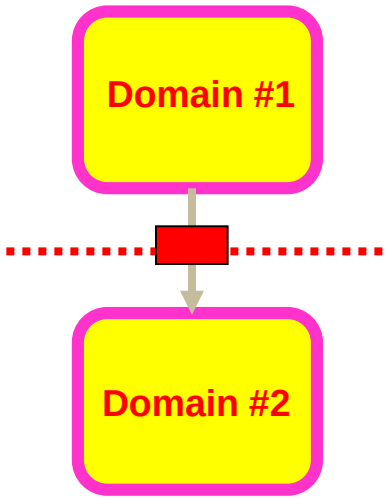
Outline

- Crossing clock domain
- Reset, synchronous or asynchronous?

Why two DFFs?

Crossing clock domain

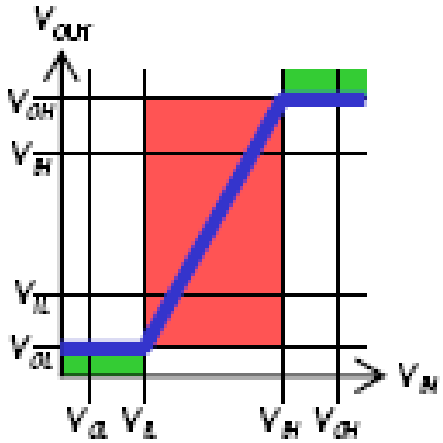
- **Multiple clock is needed in case:**
  - Inherent system requirement
    - ***Different clocks for sampling and processing***
  - Chip size limitation
    - ***Clock skew increases with the # FFs in a system***



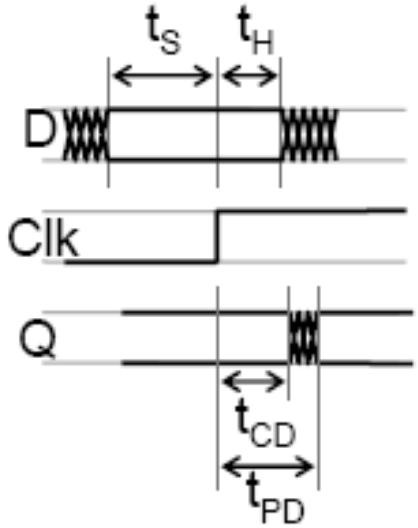
Multiple Clocks: Problems

- We have been setting very strict rules to make our digital circuits safe: using a forbidden zone in both voltage and time dimensions

Digital Values: distinguishing voltages representing "1" from "0"

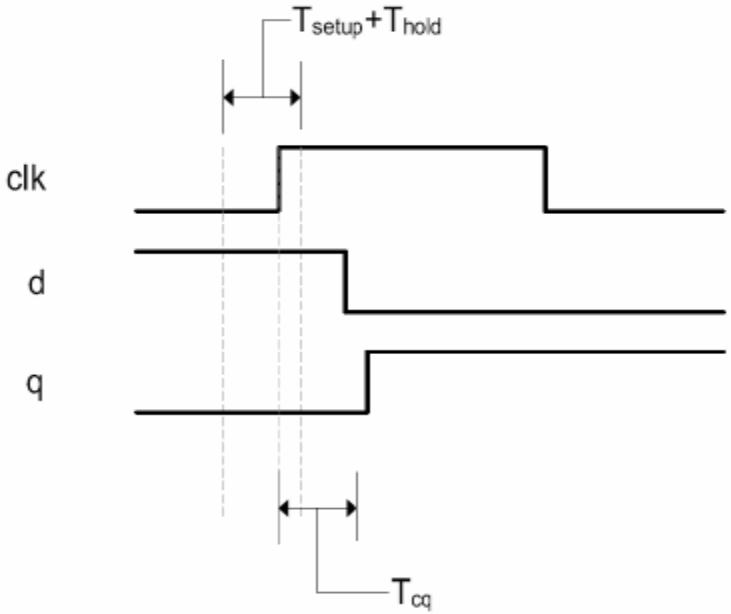


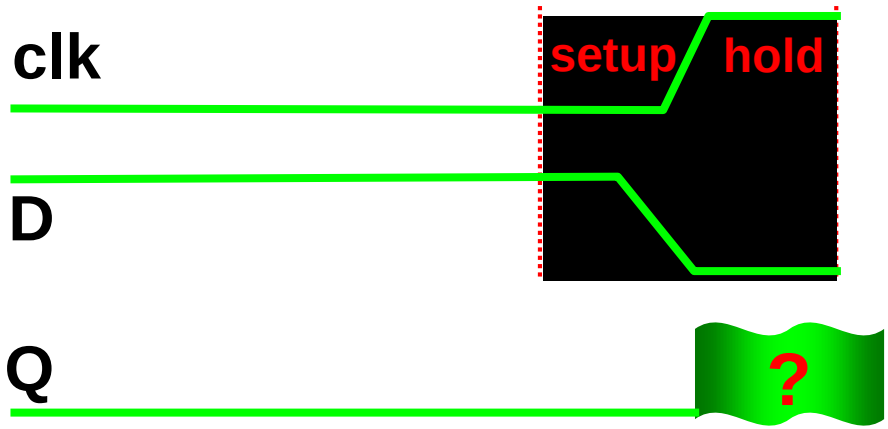
Digital Time: setup and hold time rules



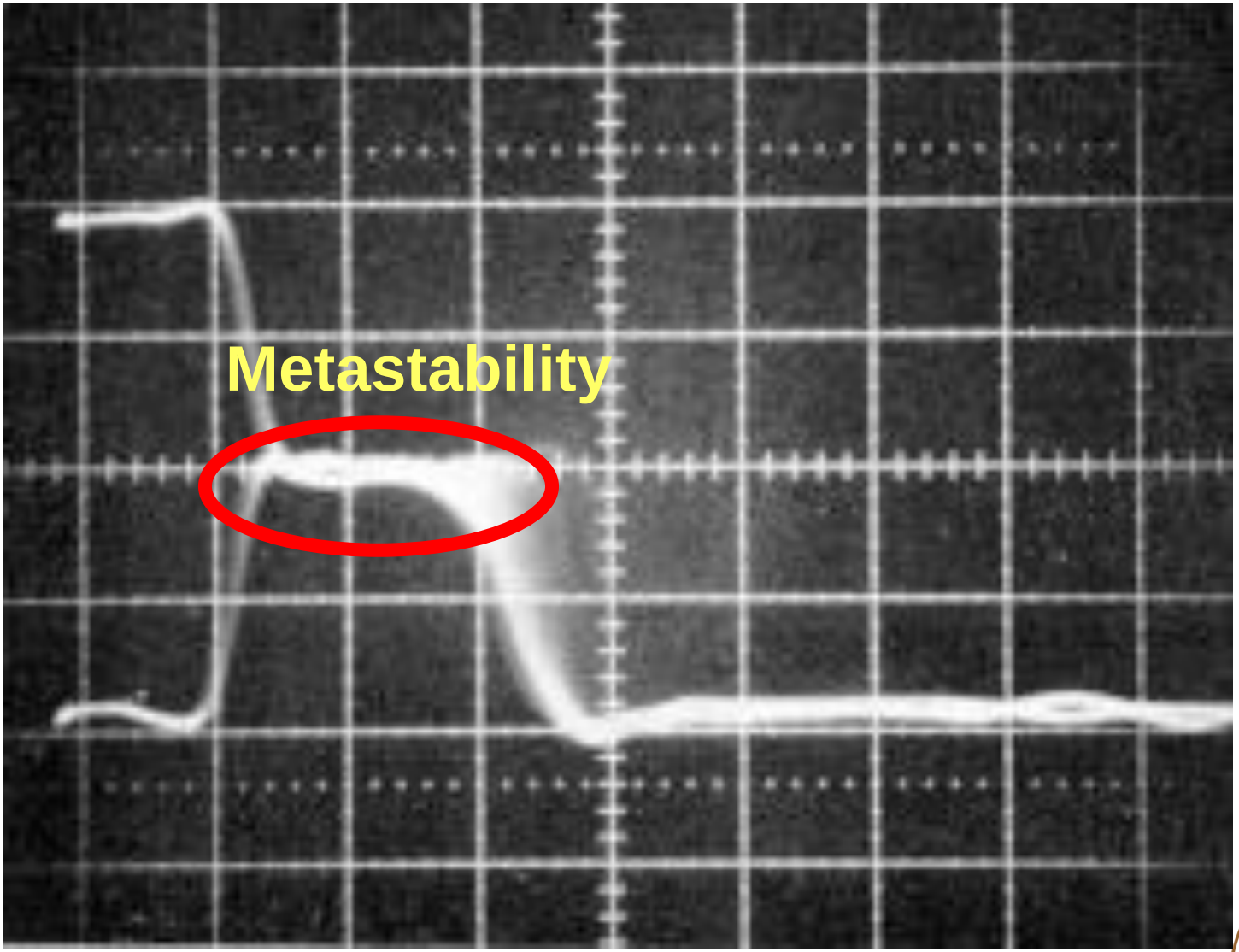
Metastability

- With asynchronous inputs, we have to break the rules: ***we cannot guarantee that setup and hold time requirements are met at the inputs!***
- What happens after timing violation?

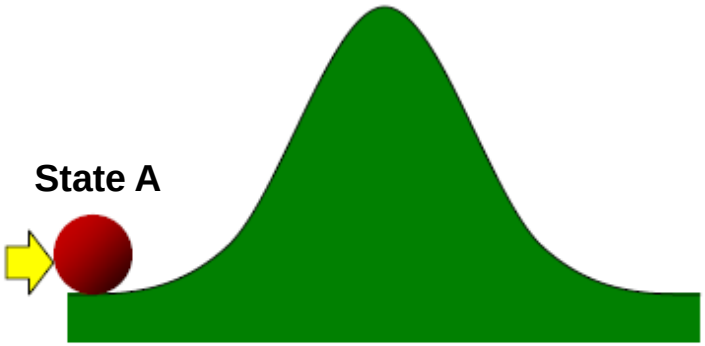




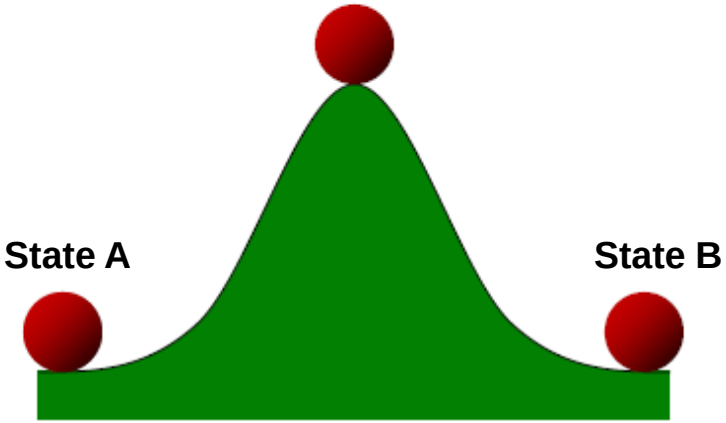
Metastability in Digital Logic



Mechanical Metastability



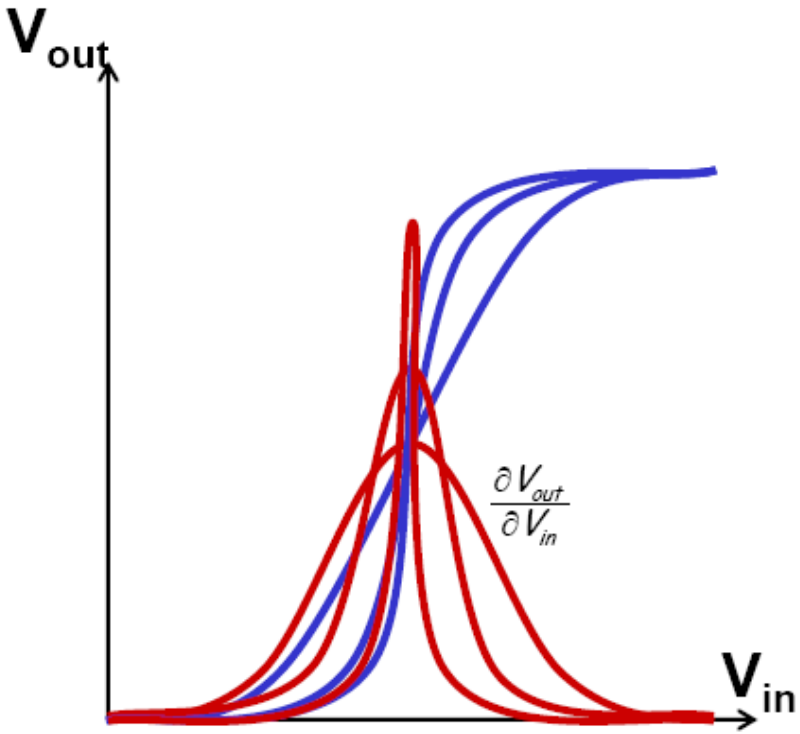
- **Launch a golf up a hill, 3 possible outcomes:**
  - Hit lightly: Rolls back
  - Hit hard: Goes over
  - Or: Stalls at the apex



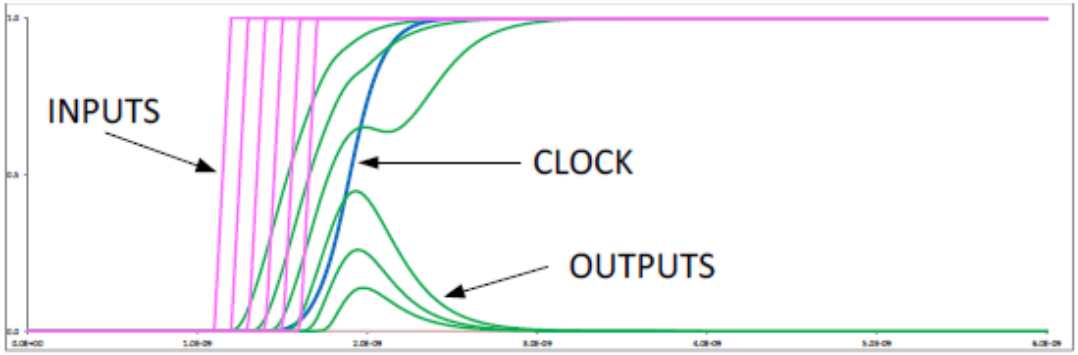
- **That last outcome is not stable:**
  - A gust of wind
  - Brownian motion
  - ***Can you tell the eventual state?***

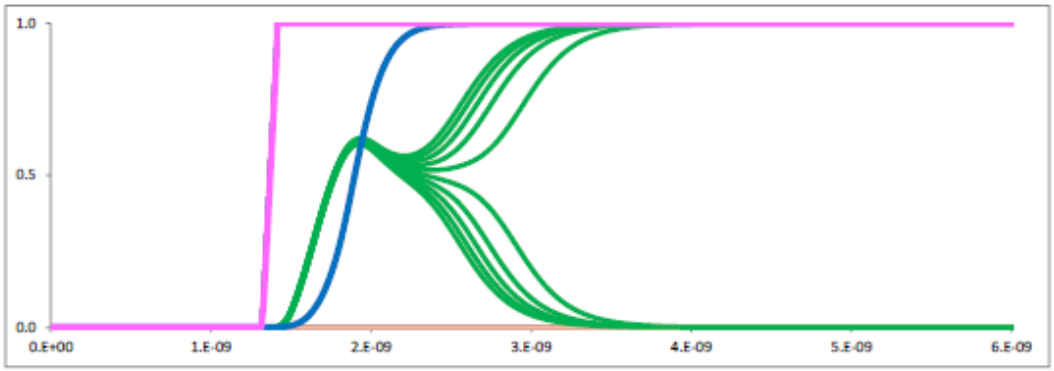
Metastability in Digital Logic

- **Our hill is related to the VTC (Voltage Transfer Curve).**
  - The higher the gain thru the transition region
  - The steeper the peak of the hill
  - The harder to get into a metastable state.
- **We can decrease the probability of getting into the metastable state, but we can't eliminate it…**



Metastability in Digital Logic



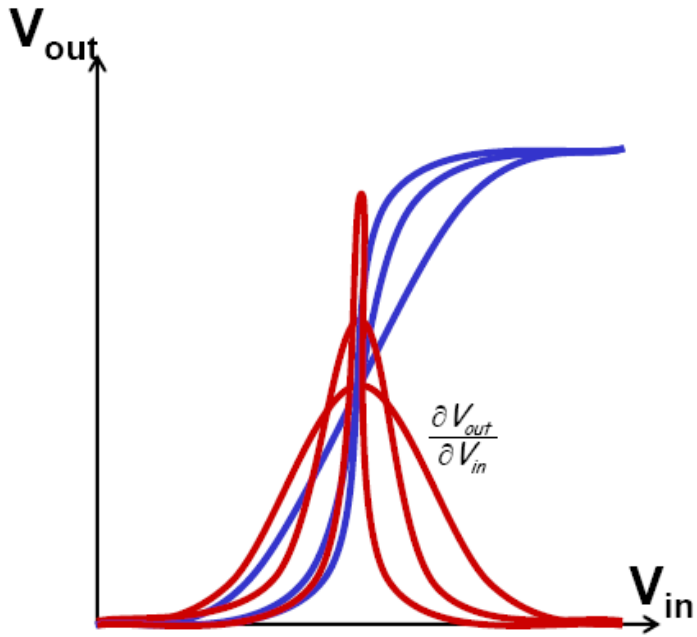


- Fixed clock edge
- Change the edge of inputs
- The input edge is moved in steps of 100ps and 1ps
- The behavior of outputs
  - 'Three' possible states
  - Will exit metastability

How long it takes to exit Metastability?

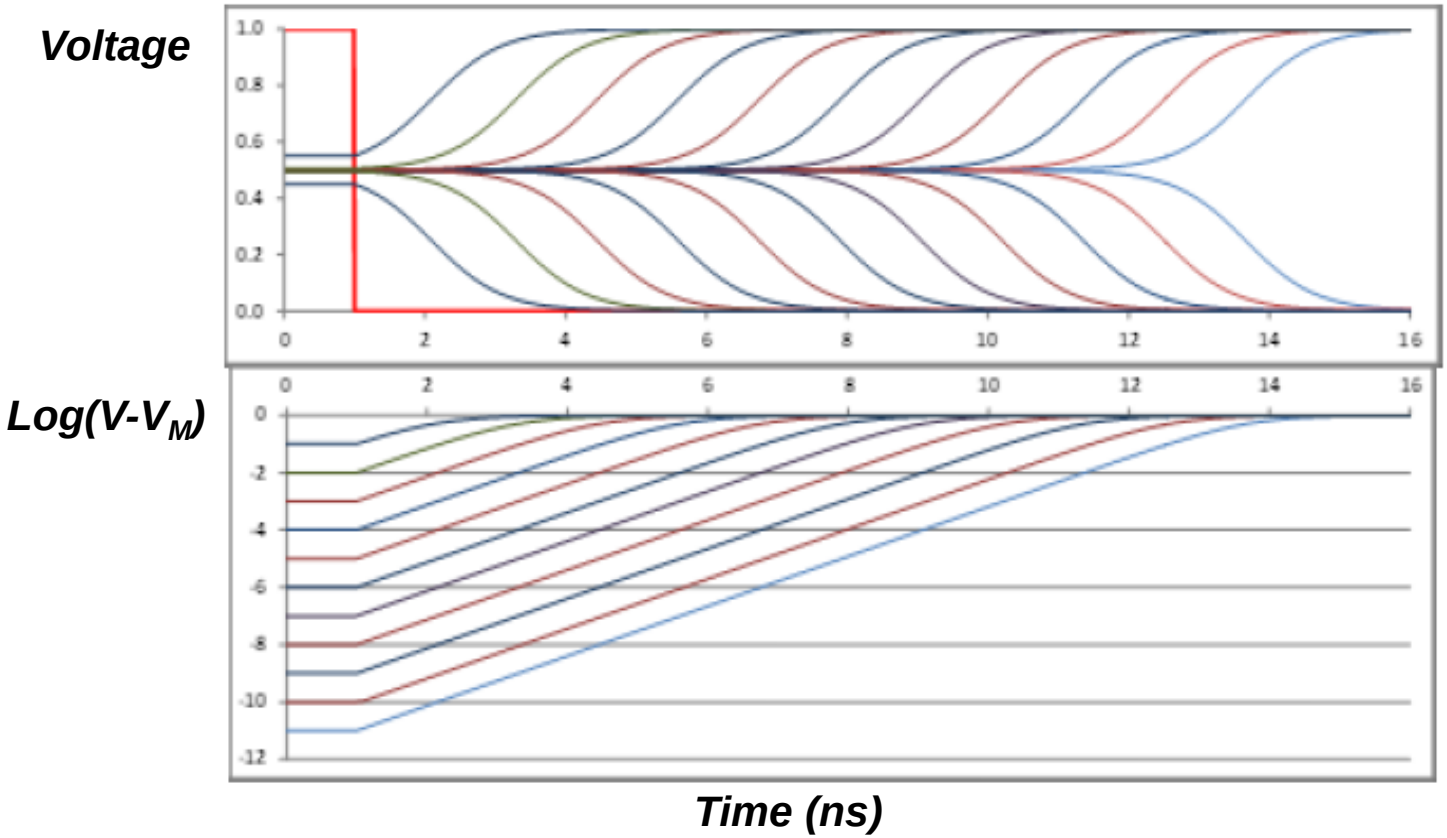
Exit Metastability

- Define a fixed-point voltage, V_M, (always have) such that $V_{IN} = V_M$ implies $V_{OUT} = V_M$
- Assume the device is sampling at some voltage V_0 near V_M
- The time to settle to a stable value depends on $(V_0 - V_M)$; its theoretically infinite for $V_0 = V_M$



Exit Metastability

- The time to exit metastability depends *logarithmically on (V_0 -V_M)*
- The *probability* of remaining metastable at time T is $e^{-T/\tau}$



MTBF: The probability of being metastable at time S?

- **Two conditions have to be met concurrently**
  - An FF enters the metastable state
  - An FF cannot resolve the metastable condition within S
- **The rate of failure** $p(failure) = p(enter\ MS) \times p(time\ to\ exit > S)$

$$Rate(failures) = T_W F_C F_D \times e^{-S/\tau}$$

  - T_W: time window around sampling edge incurring metastability
  - F_C: clock rate (assuming data change is uniformly distributed)
  - F_D: input change rate (input may not change every cycle)
- **Mean time between failures (MTBF)**

$$MTBF = \frac{e^{S/\tau}}{T_W F_C F_D}$$

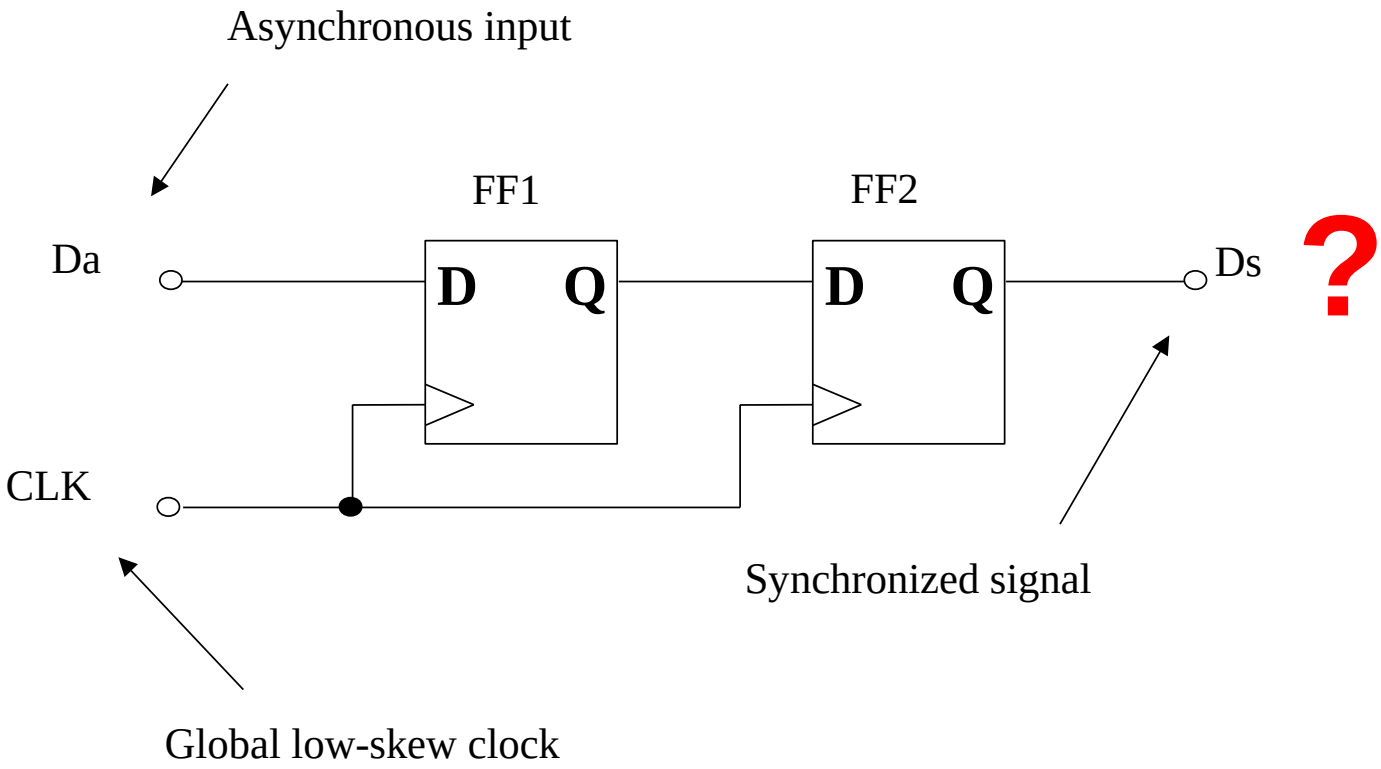
MTBF (Mean Time Between Failure)

- **Let's calculate an ASIC for 28nm CMOS process**
  - τ: 10ps (different FFs have different τ)
  - T_W=20ps, F_C=1GHz
  - Data changes every ten clock cycles
  - Allow 1 clock cycle to resolve metastability, **$S=T_C$**

$MTBF=4\times10^{29}$ year !

[For comparison:
Age of oldest hominid fossil: $5x10^6$ years
Age of earth: $5x10^9$ years]

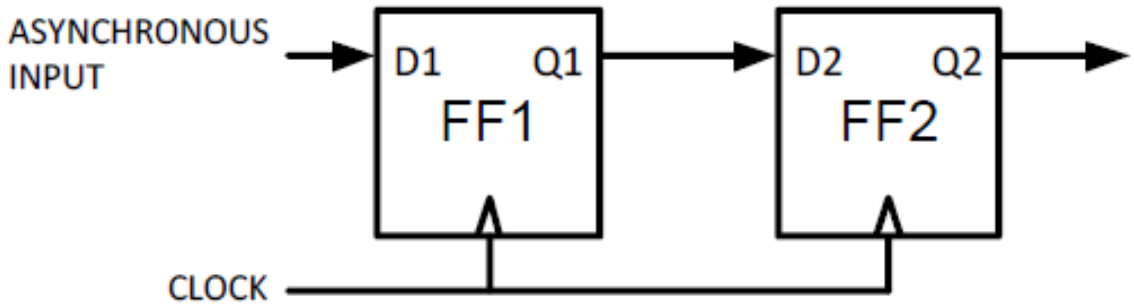
The Two-Flip-Flop Synchronizer

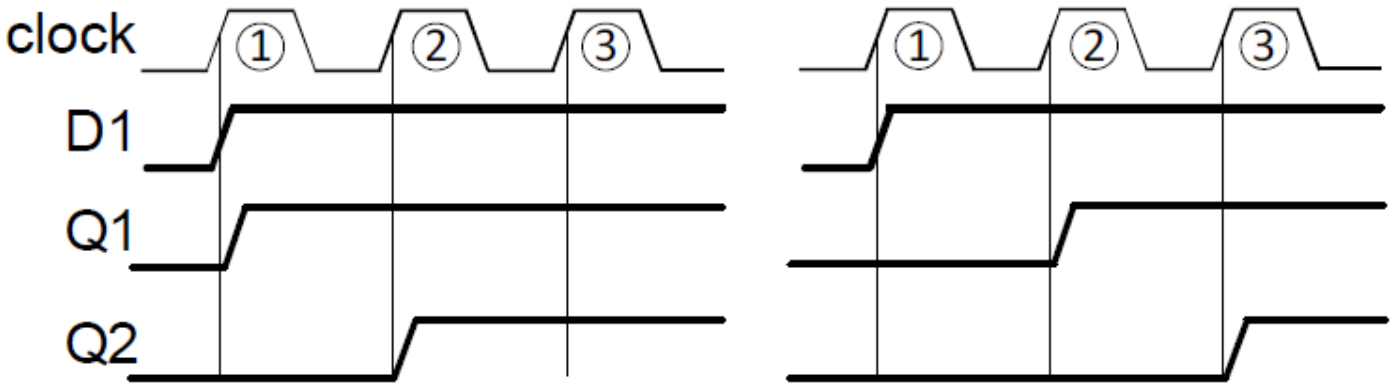


$$S = T_C$$

The Two-Flip-Flop Synchronizer

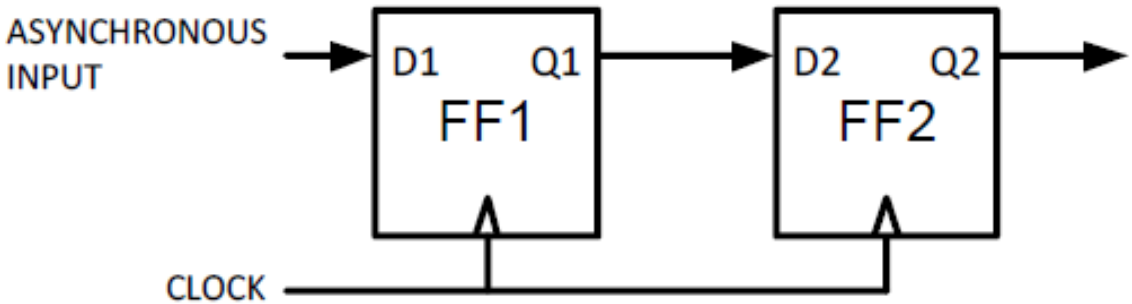
- **Possible Outcomes**

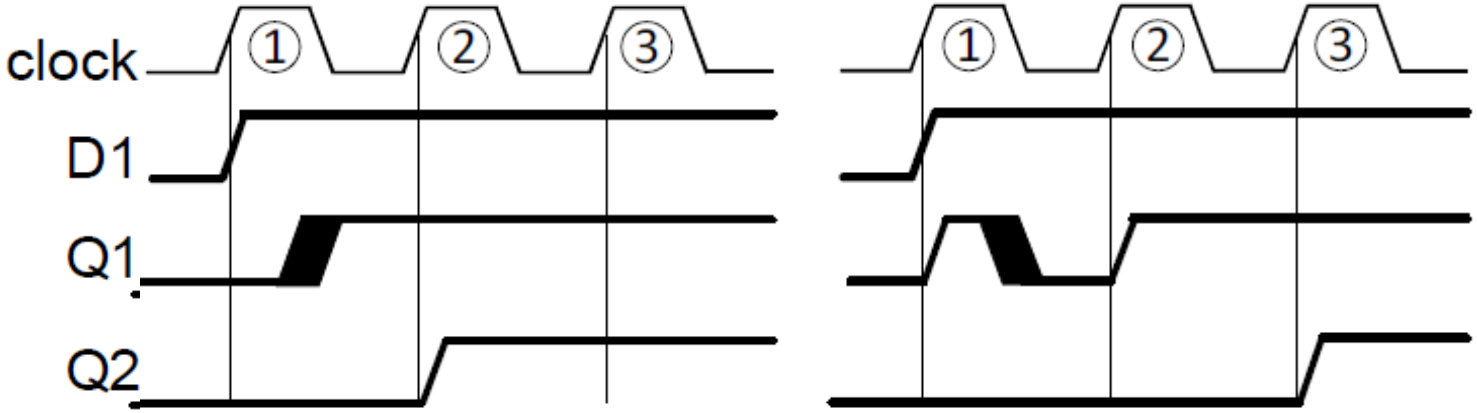




The Two-Flip-Flop Synchronizer

- **Possible Outcomes**



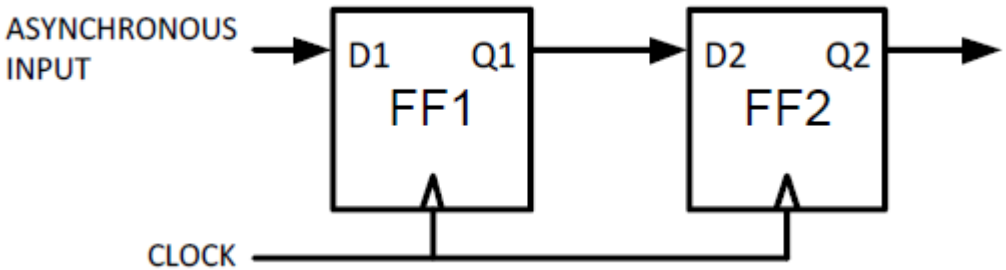


Open Question: What is the limitation?

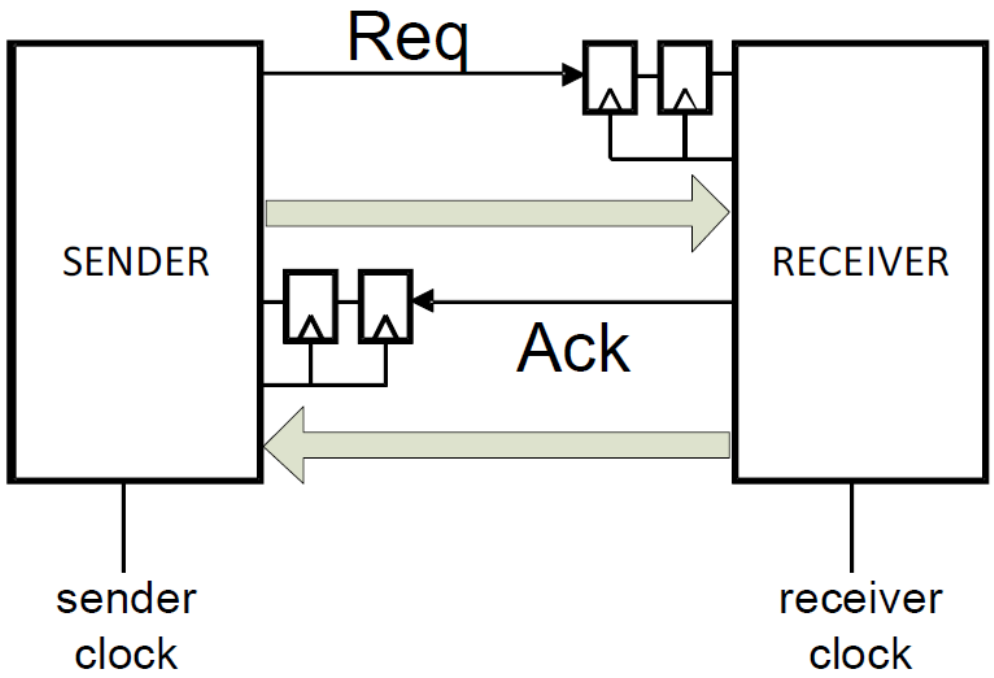
The Two-Flip-Flop Synchronizer

Problems

- Just ensures that the receiving system does not enter a metastable state
- Not guarantee the **"function"** of the received signal
  - ***Uncertainty Remains: Q2 goes high either one or two cycles later than the input***
- D1 mush stay high for at *least two cycles*.
- How about data bus (***multiple bits***) crossing clock domain?
  - ***Some bits may pass through the synchronizer after one cycle while others may take two cycles.***



A Complete Synchronizer

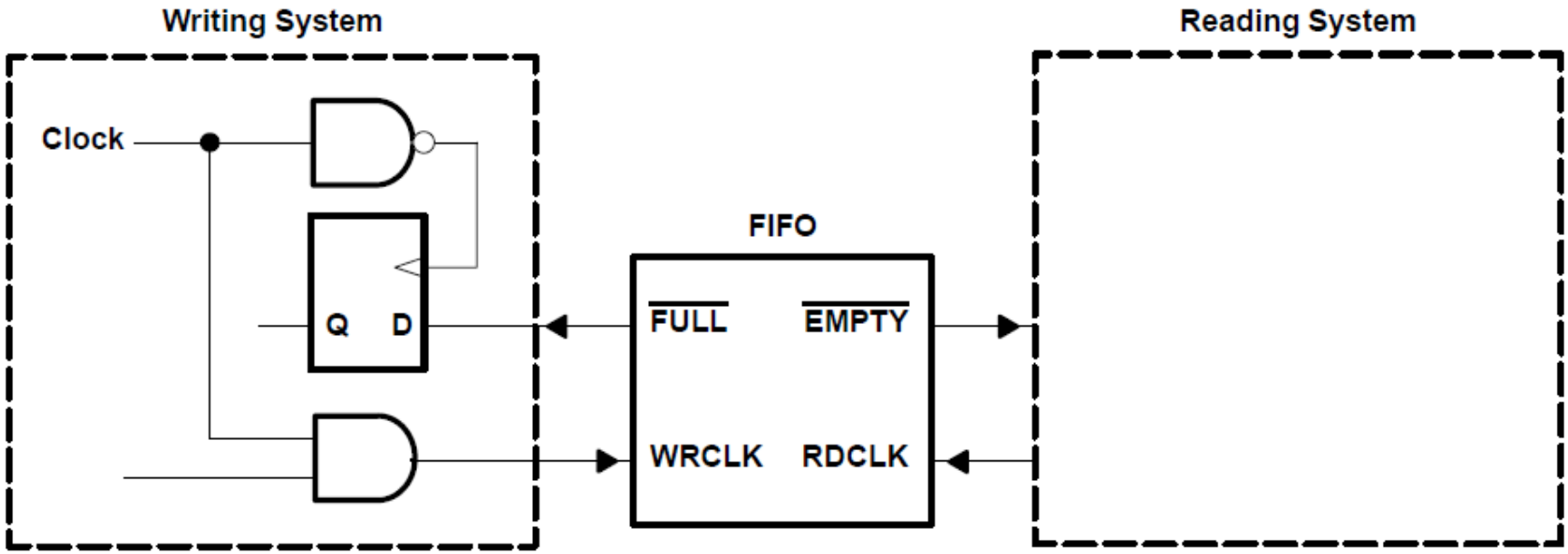


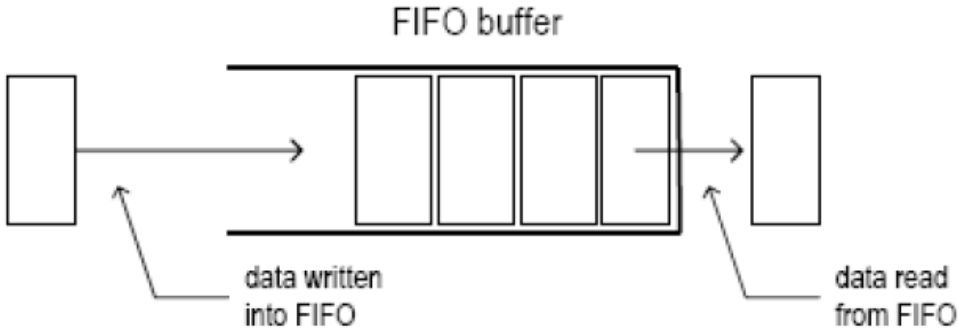
- The sender place data on the bus
- The sender sends Req, Req gets synchronized by the top synchronization circuits
- The receiver gets data and sends back ACK
- Ack gets synchronized by the sender, and only then is the sender allowed to start a new cycle again.

FIFO

FIFO (first in first out) Buffer

- "Elastic" storage between two subsystems



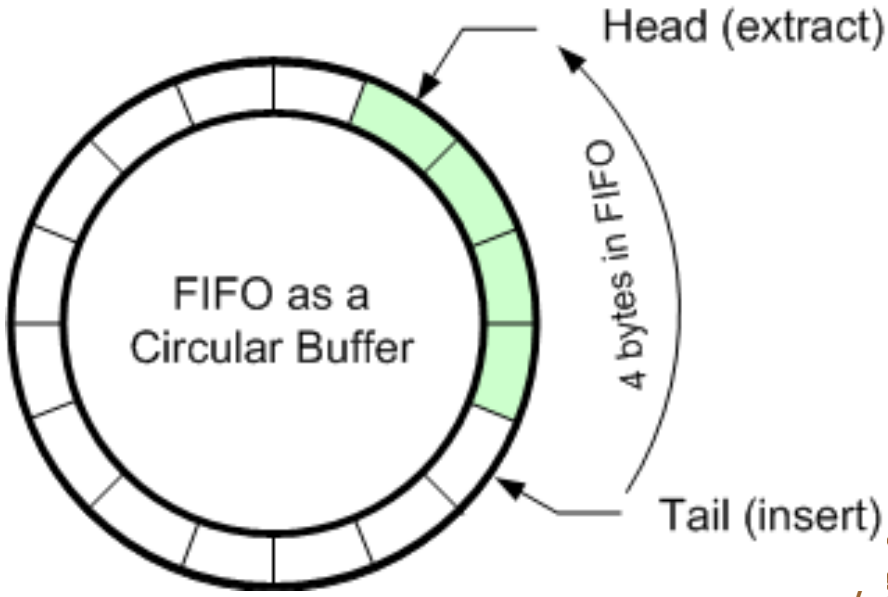


Circular FIFO

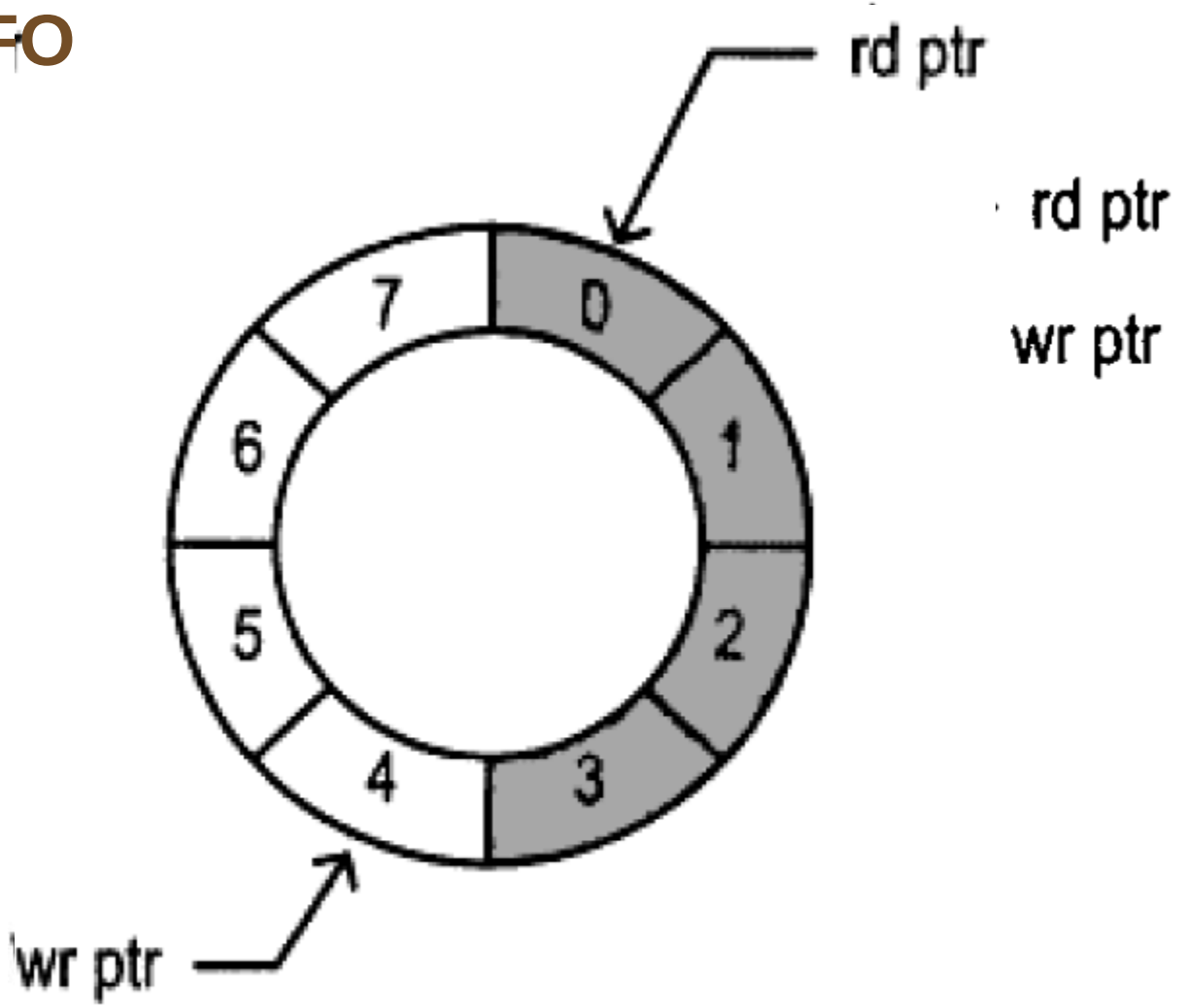
How to Implement a FIFO?

- Circular queue implementation
- Use two pointers and a “generic storage”
  - ***Write pointer*: point to the empty slot before the *head* of the queue**
  - ***Read pointer*: point to the *tail* of the queue**

"First in? First out!"



Circular FIFO

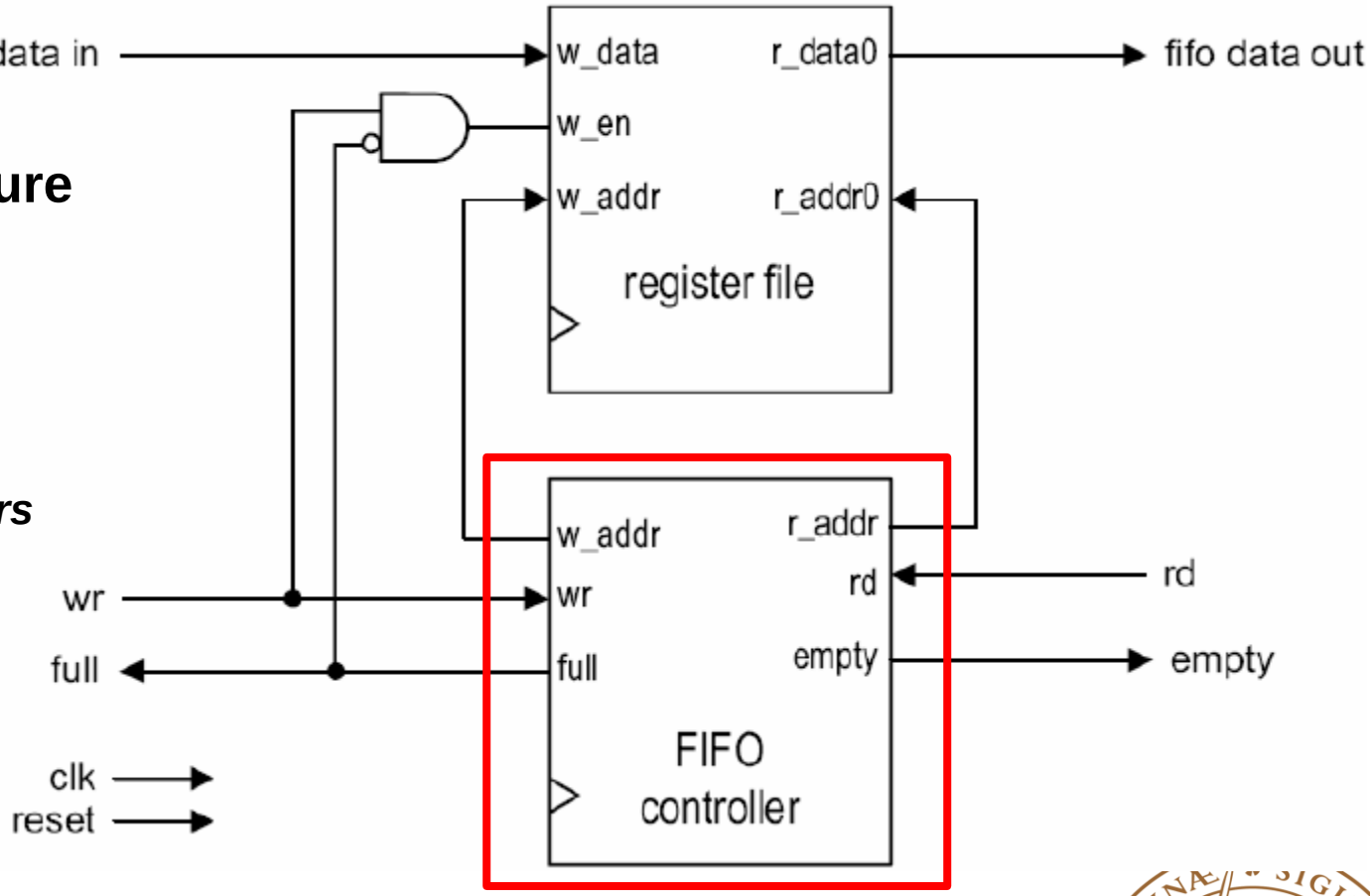


(c). 4 more writes

FIFO Implementation

- **Overall Architecture**
  - Storage Elements
    - ***Reg. file***
  - **FIFO Controller**
    - ***Read and write pointers: 2 counters***
    - ***Status circuit:***

      full, empty



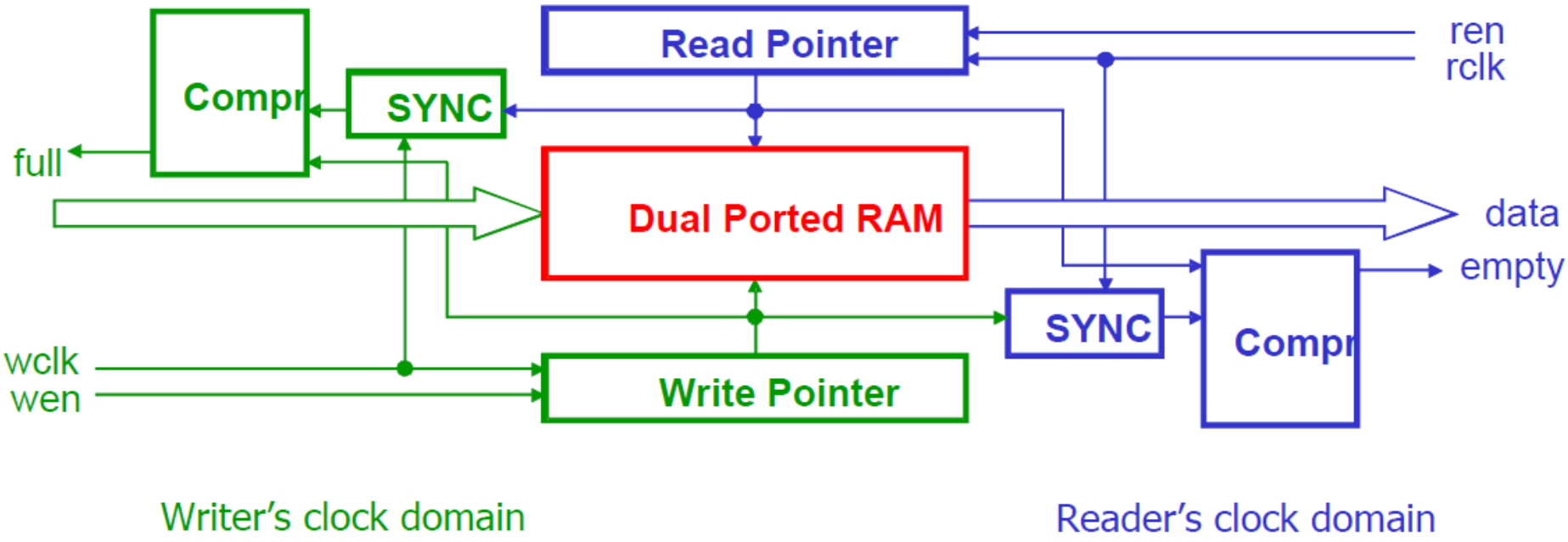
FIFO Implementation: Controller

- **Augmented binary counter:**
  - Increase the counter by 1 bits
  - Use LSBs for as register address
  - Use **MSB** to distinguish full or empty

Write pointer	Read pointer	Operation	Status
0 000	0 000	initialization	empty
0 111	0 000	after 7 writes	
1 000	0 000	after 1 write	full
1 000	0 100	after 4 reads	
1 100	0 100	after 4 writes	full
1 100	1 011	after 7 reads	
1 100	1 100	after 1 read	empty
0 011	1 100	after 7 writes	
0 100	1 100	after 1 write	full
0 100	0 100	after 8 reads	empty

0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000

A Complete Synchronizer



- ***Control signals*** **are 'synchronized',** ***data*** **pass through storage elements**
- **Assertion delay, it takes** ***two clock*** **cycles for the control signal passing through synchronizer**
- **Usually available in libraries**
- **The key question,** ***how large the RAM should be?***
- **"When in doubt, double it"**

Lecture

- **Sept. 26th Monday (13.15-15.00)**

Design for Test (DFT)

Erik Larsson

Associate Professor

- **Sept. 27th Tuesday (13.15-15.00)**

Stefan Lundberg AXIS COMMUNICATIONS

Lecture

- Oct. 3th Monday No Lecture

- Oct. 4th Tuesday (13.15-15.00)

Charlotte Sköld
Sadat Rahman