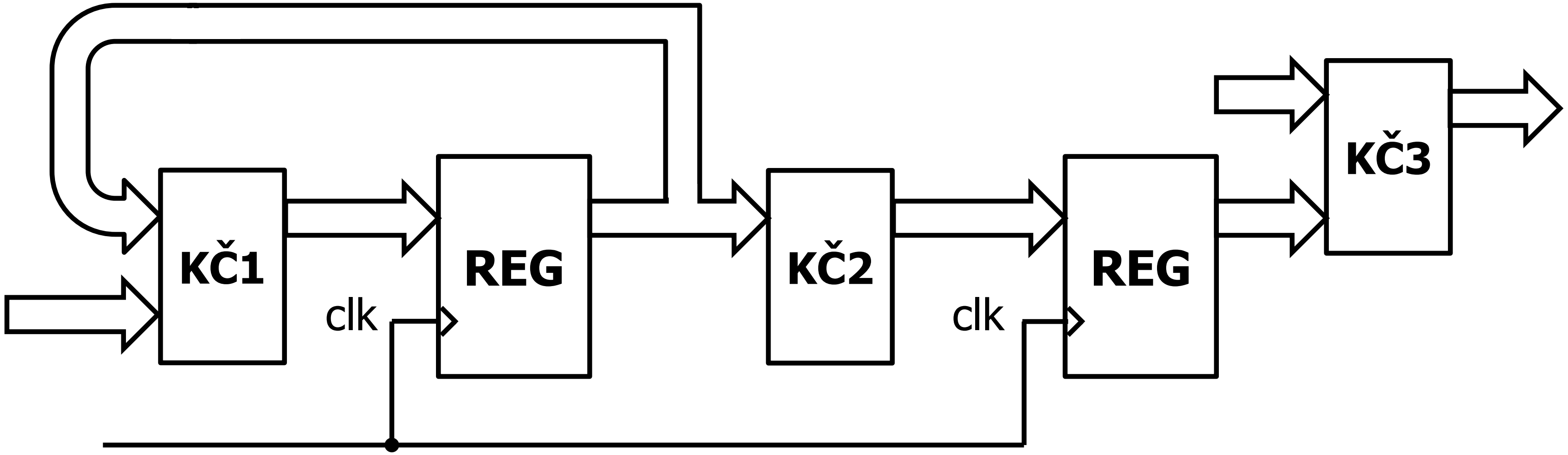
PA221: Clock Domain Crossing (CDC): Selected Parts

jan.kral@fi.muni.cz

Clock Domains in FPGAs

Clock Domain

- ❑ A group of Flip-Flops (registers) sensitive to the same clock edge (usually rising one) of the same clock signal.
- ❑ A minimal difference of clock signal propagation delay to individual Flip-Flops is expected (minimal skew); i.e. all the registers flip at exactly same time instant.

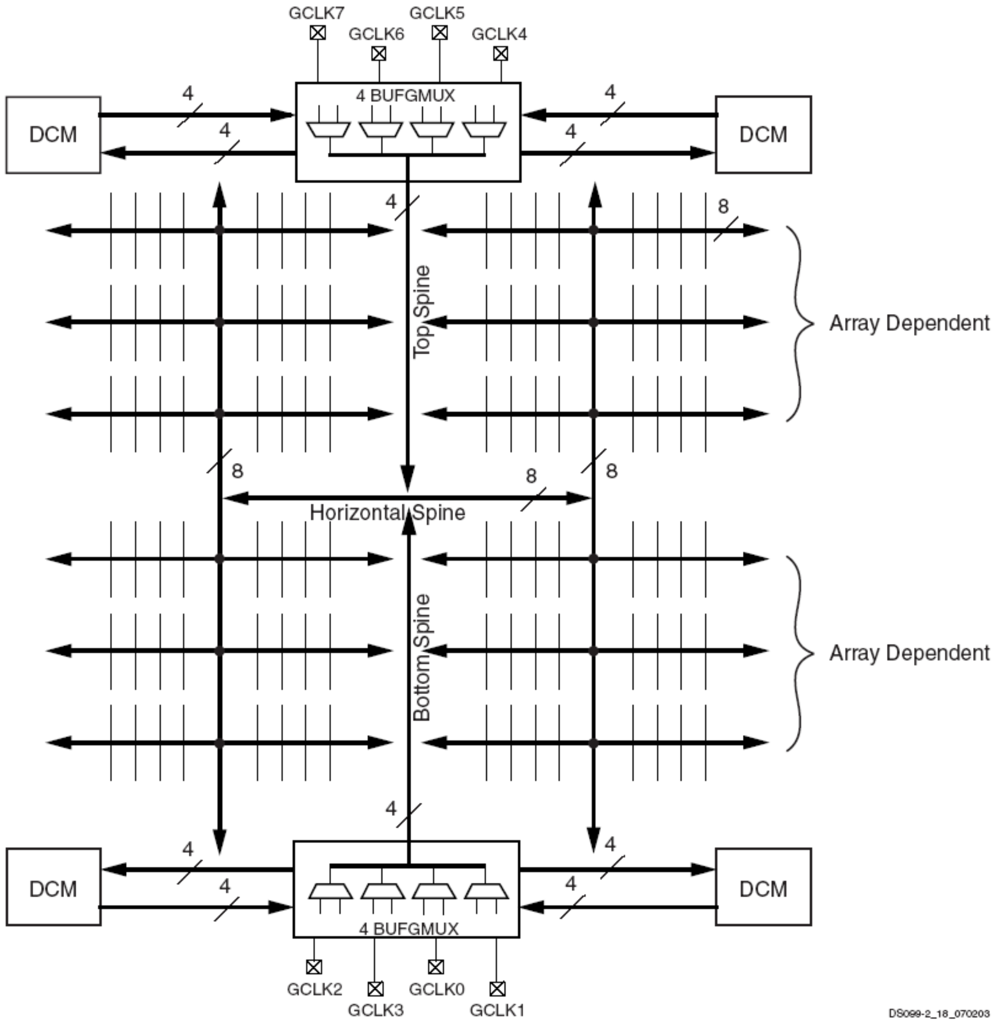


Clock Signal Distribution

- ☐ Goal:

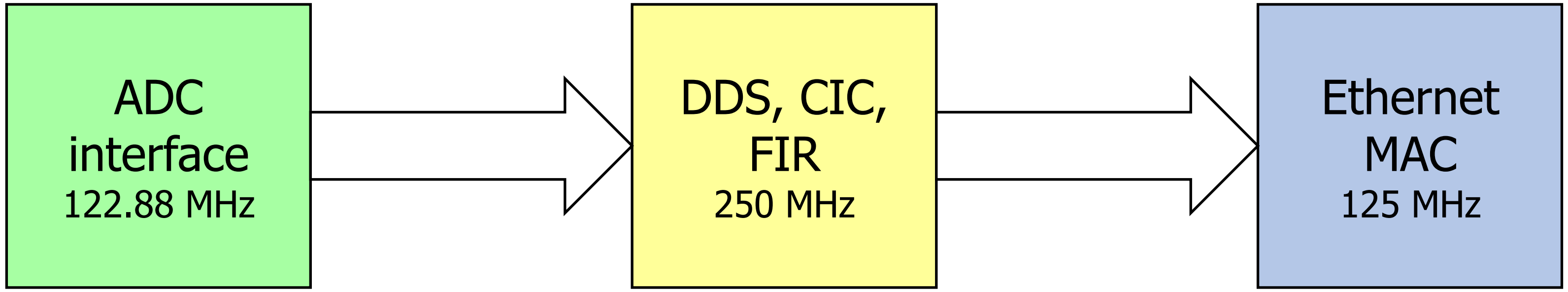
  to **minimize difference** of clock signal propagation delay to individual Flip-Flops. (*to minimize* ***skew***)

Xilinx Spartan-3



Why to use more clock signals?

- ❑ In real designs, more clock signals are enforced by system requirements. For example, data acquisitions requires different clock frequency than a standard communication interface.
- ❑ Formally: Globally asynchronous, locally synchronous sequential systems



ADC interface 122.88 MHz

DDS, CIC, FIR 250 MHz

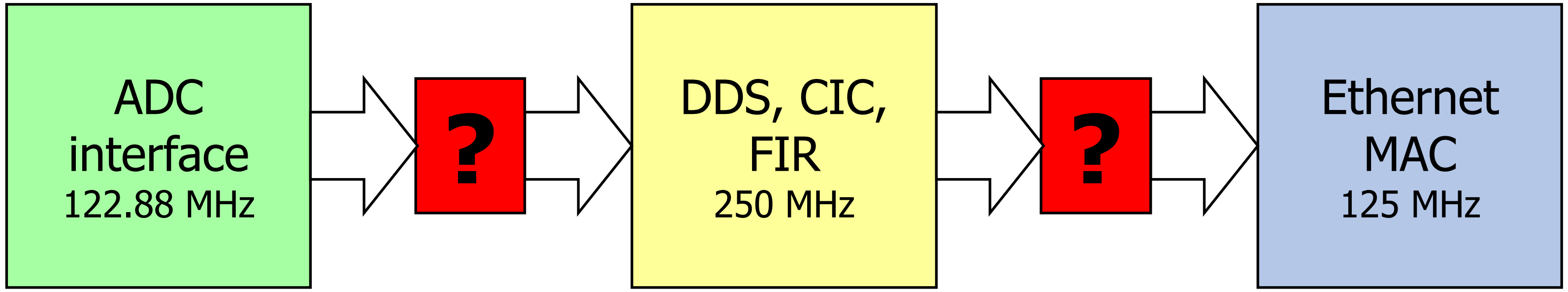
Ethernet MAC 125 MHz

Why to use more clock signals?

- ❑ **High clock frequency** is only suitable for critical parts of the design (fast interfaces, fast data processing blocks) because:
  - ❑ **Lower clock frequency** puts **lower requirements** on the programmable interconnect → faster Place and Router procedure (relaxed requirements)
  - ❑ **Lower clock frequency** → **lower** dynamic **power consumption** (lower thermal dissipation)

Problem: Interfacing between clock domains

- There are usually many data and/or control signals that must pass from one clock domain to another. As clock domains are **asynchronous** to each other, a special care must be taken to avoid **metastability** and **data loss/duplicity**.

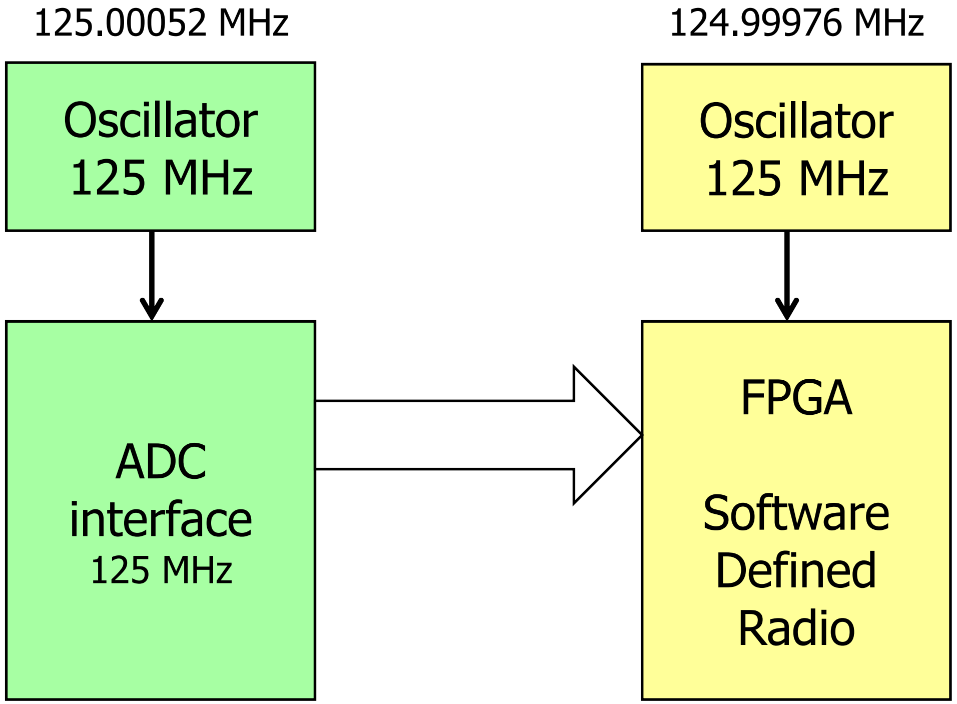


MUNI
FI

Signal Crossing over Clock Domain Boundary

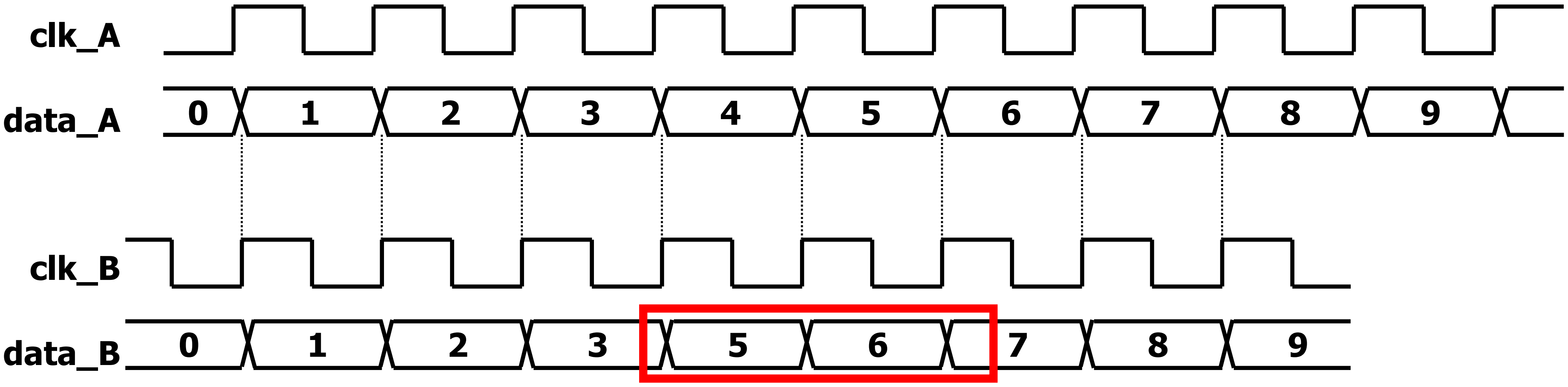
CDC: Common Mistake

- ❑ Data interface between two clock domains having same reference frequency but derived from their own oscillators.



- ❑ The oscillators may have same nominal frequency, but the actual frequency is always slightly different.
- ❑ The difference is not stable over time – clock jitter / wander.
- ❑ When reading data on the receiver side (in FPGA on the example shown) some data words may be duplicated or skipped.

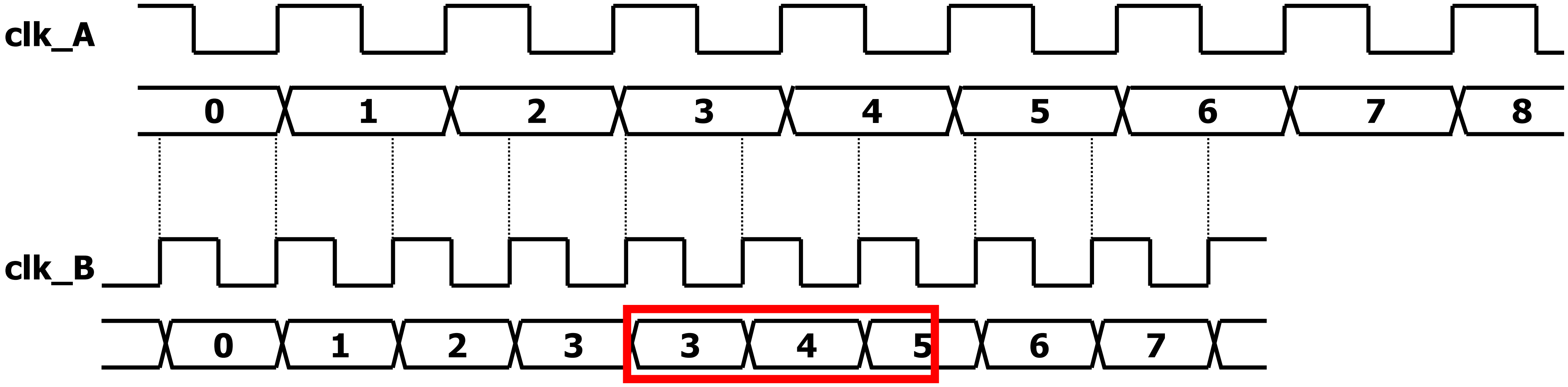
Possible Effects



Skipping some data words

$F_{clkA} > F_{clkB}$

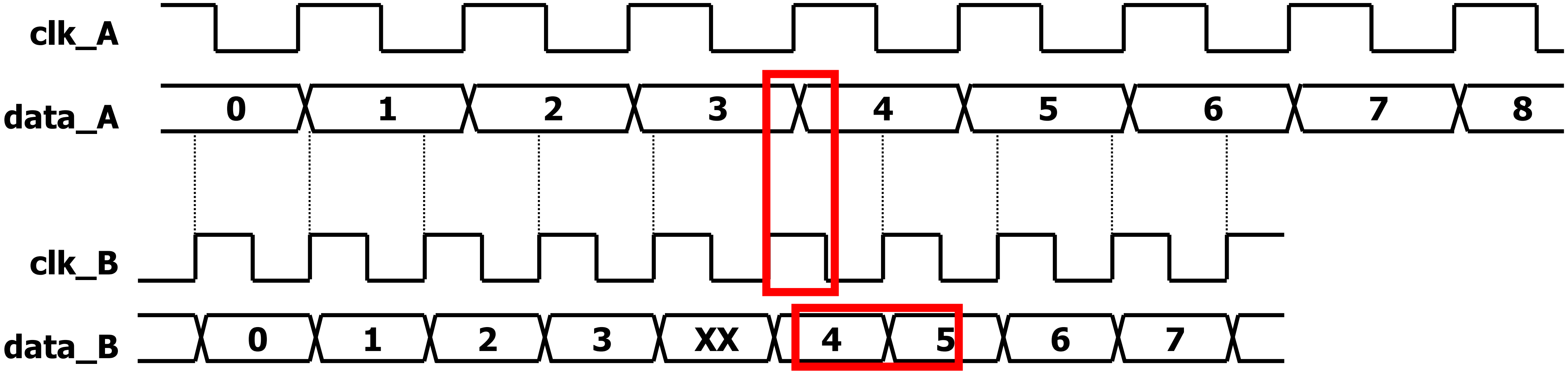
Possible Effects



Duplicity of some data words

$F_{clkA} < F_{clkB}$

Possible Effects



A metastability can occur when setup/hold time is violated. Invalid (random) data read.

$F_{clkA} < F_{clkB}$

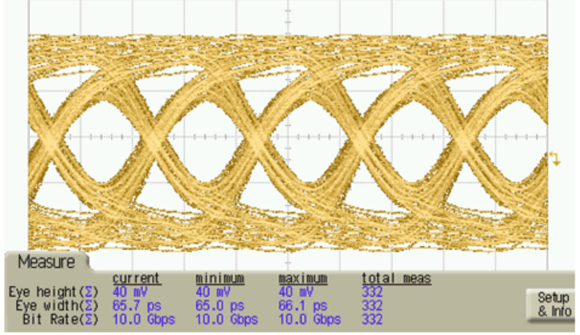
Possible Effects: Incorrect Bus Data

clk_A

data_A

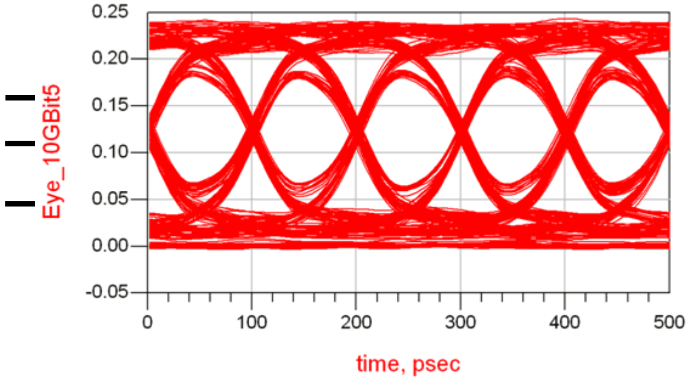
clk_B

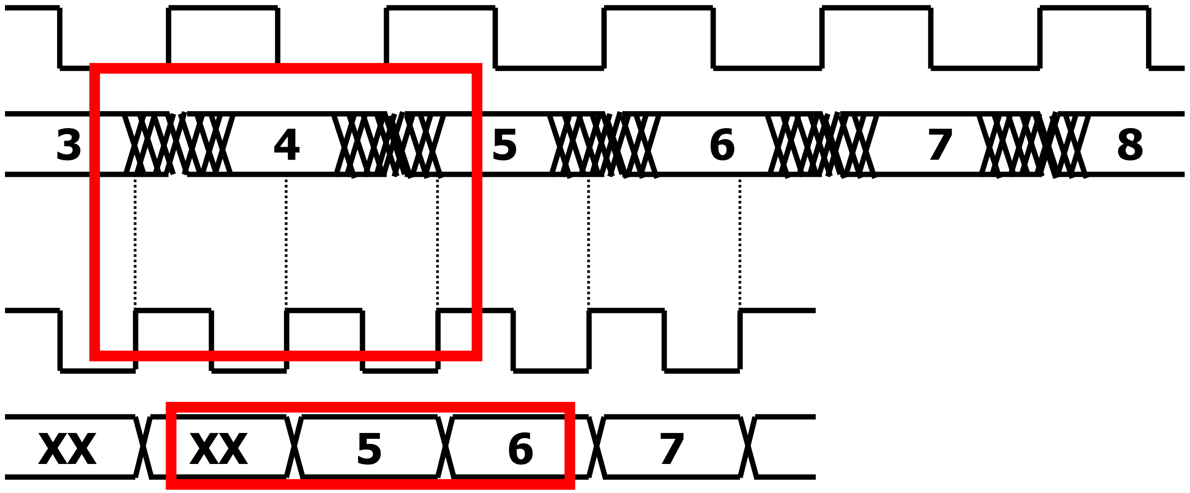
data_B



Measure

	current	minimum	maximum	total meas
Eye height(Σ)	40 mV	40 mV	40 mV	332
Eye width(Σ)	65.7 ps	65.0 ps	66.1 ps	332
Bit Rate(Σ)	10.0 Gbps	10.0 Gbps	10.0 Gbps	332

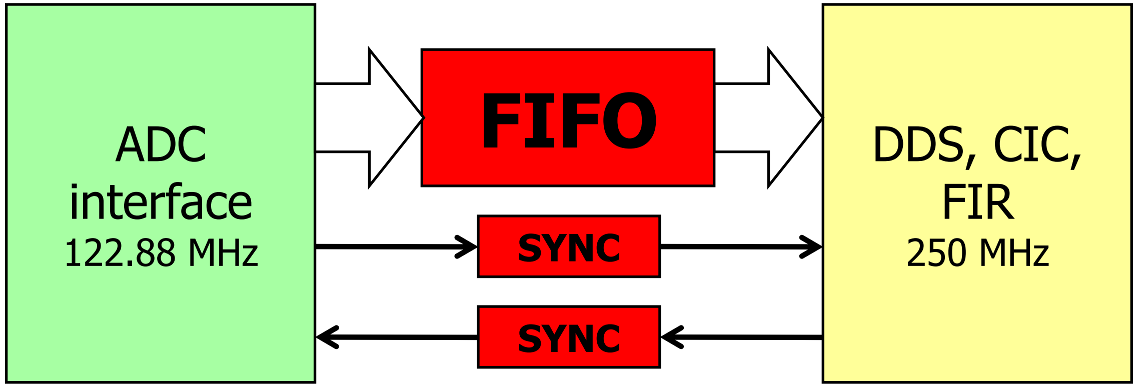




Solution

- ❑ **Simple synchronizer** – suitable for control signals or slow data
- ❑ **Asynchronous FIFO buffer** – with independent clocks at write and read side; suitable for high-speed data transfers

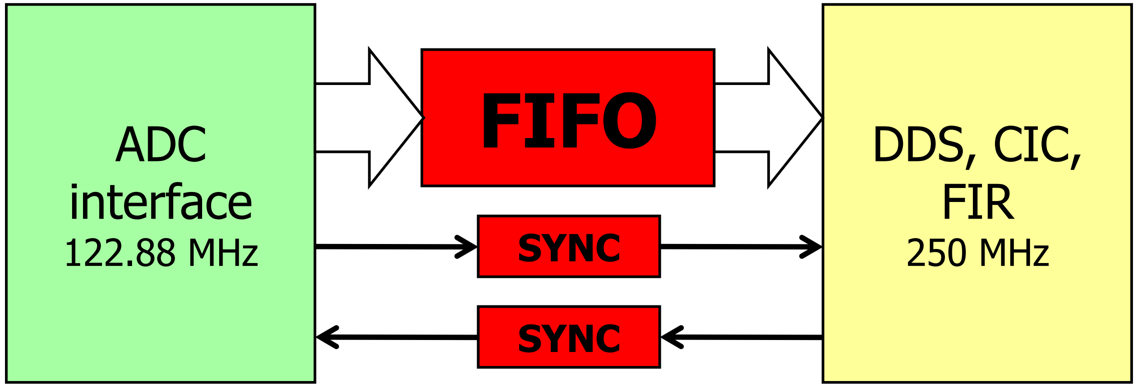
Both solutions can deal with data transfer in one direction only; for bidirectional transfer it is necessary to double each structure.



Solution

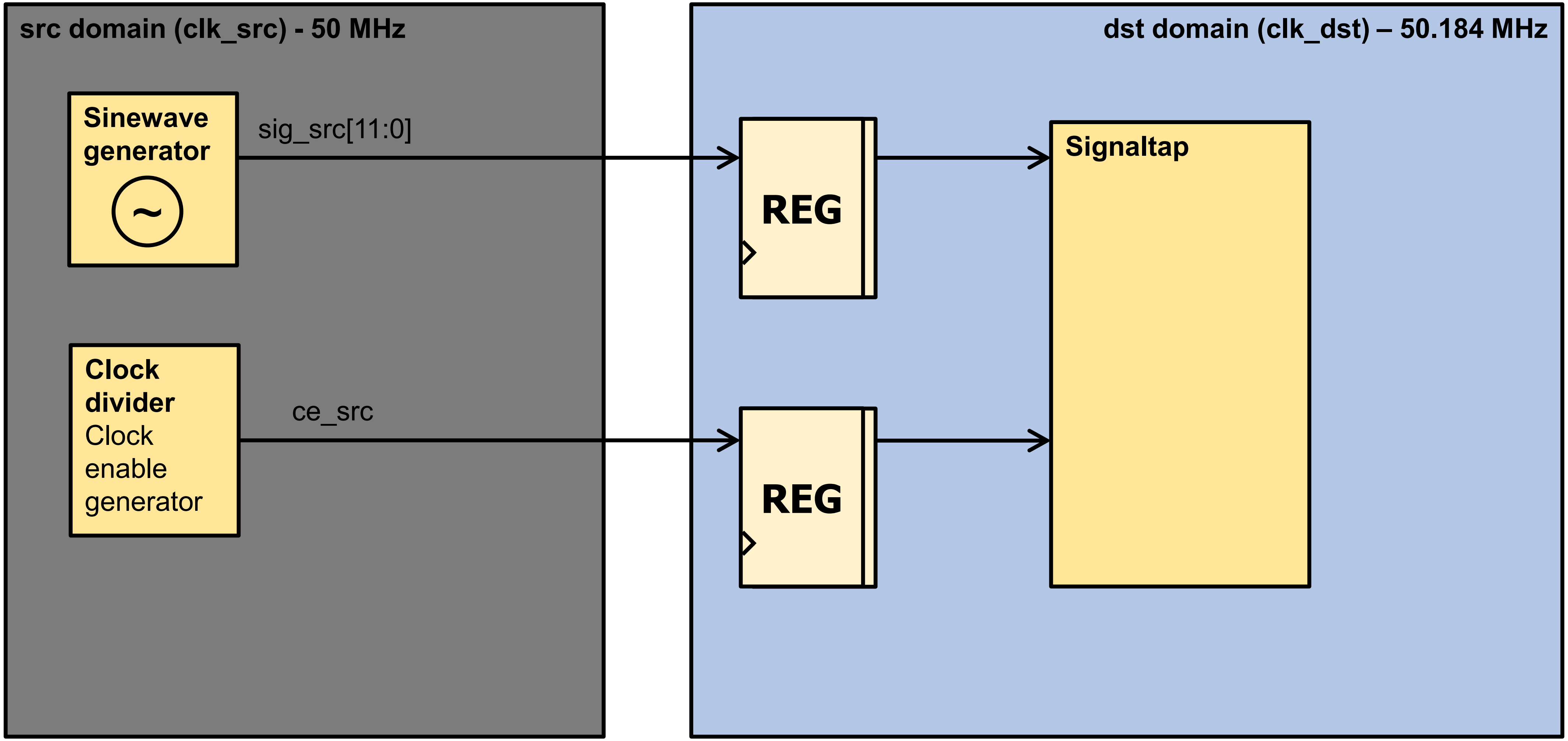
- ❑ **The actual solution depends on:**
  - ❑ Single bit or multiple-bit signal
  - ❑ Clock frequency of source and destination domains
  - ❑ Character of the signal (long or short impulse)
  - ❑ Is a transfer required every clock cycle?

http://www.fpga4fun.com/CrossClockDomain.html



Assignment

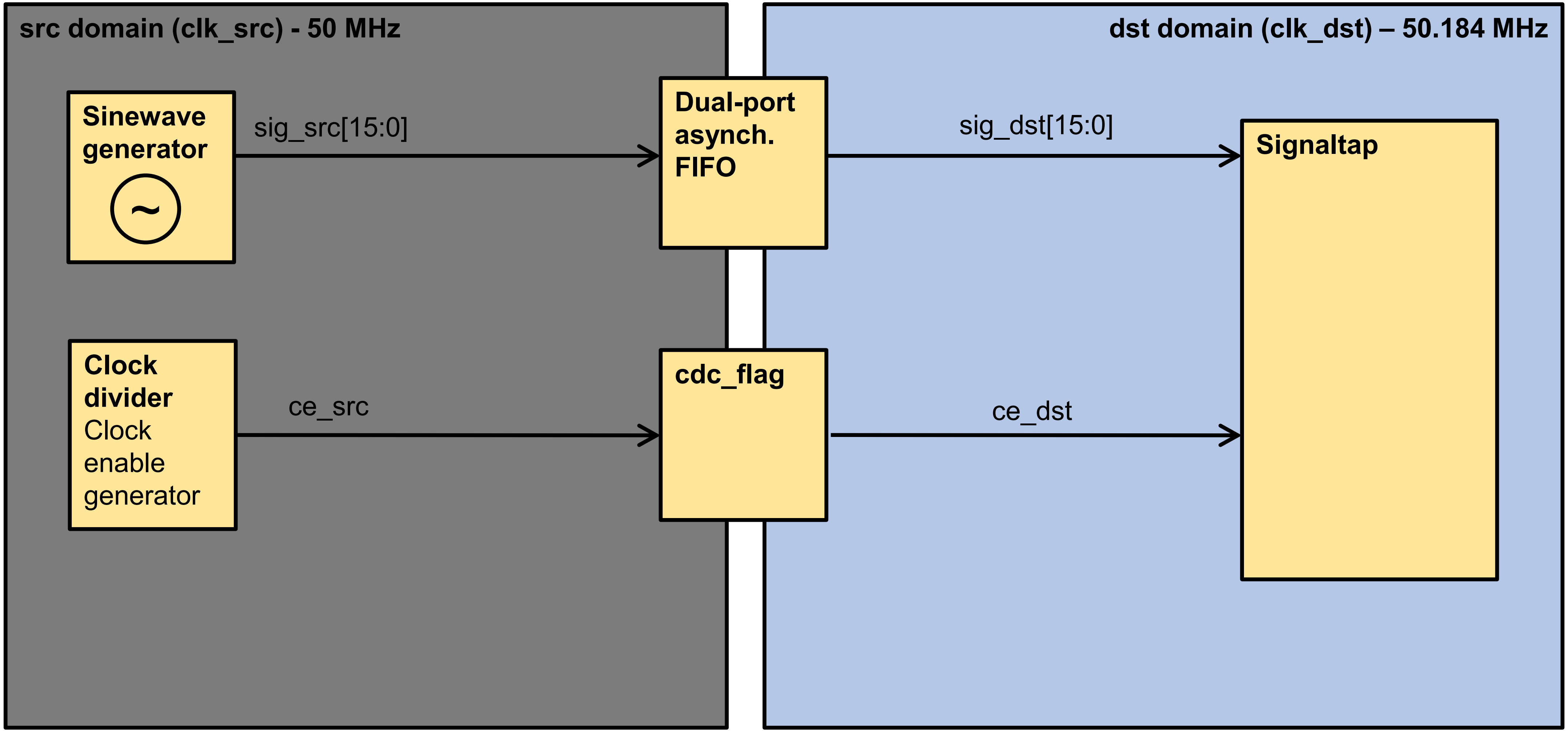
Template


src domain (clk_src) - 50 MHz
Sinewave generator
~
sig_src[11:0]
Clock divider
Clock enable generator
ce_src
dst domain (clk_dst) – 50.184 MHz
REG
REG
Signaltap

Assignment

- ❑ Observe the cross-domain effects in signaltap
- ❑ Add FIFO on sig_src to achieve a proper domain crossing
- ❑ Create your own synchronizer for clock enable signal (and signals with similar waveforms)

Solution



src domain (clk_src) - 50 MHz
dst domain (clk_dst) – 50.184 MHz
Sinewave generator
~
sig_src[15:0]
Dual-port asynch. FIFO
sig_dst[15:0]
Signaltap
Clock divider
Clock enable generator
ce_src
cdc_flag
ce_dst

cdc_flag

In source domain:

- ❑ Convert the short input pulses to level changes
  - ❑ each impulse will toggle a signal
  - ❑ prolonged signals can cross the clock-domain boundary

In destination domain:

- ❑ Add two D-flops to remove metastability
- ❑ Edge detector converts the level changes back one-clock-cycle impulses

cdc_flag

Recommended port naming:

clk_src

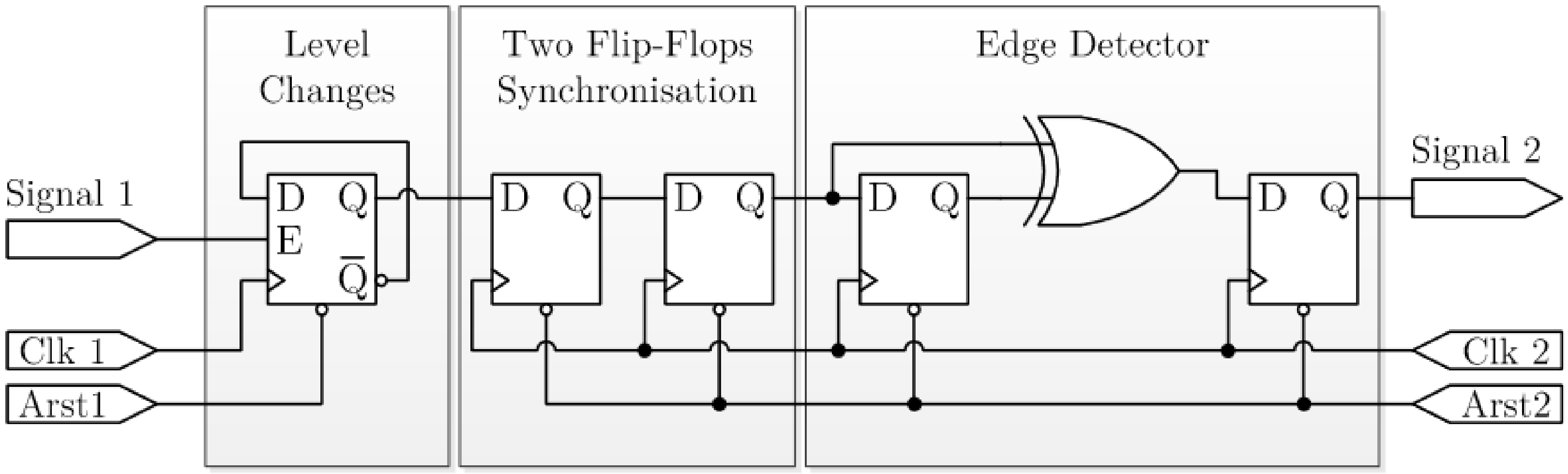
clk_dst

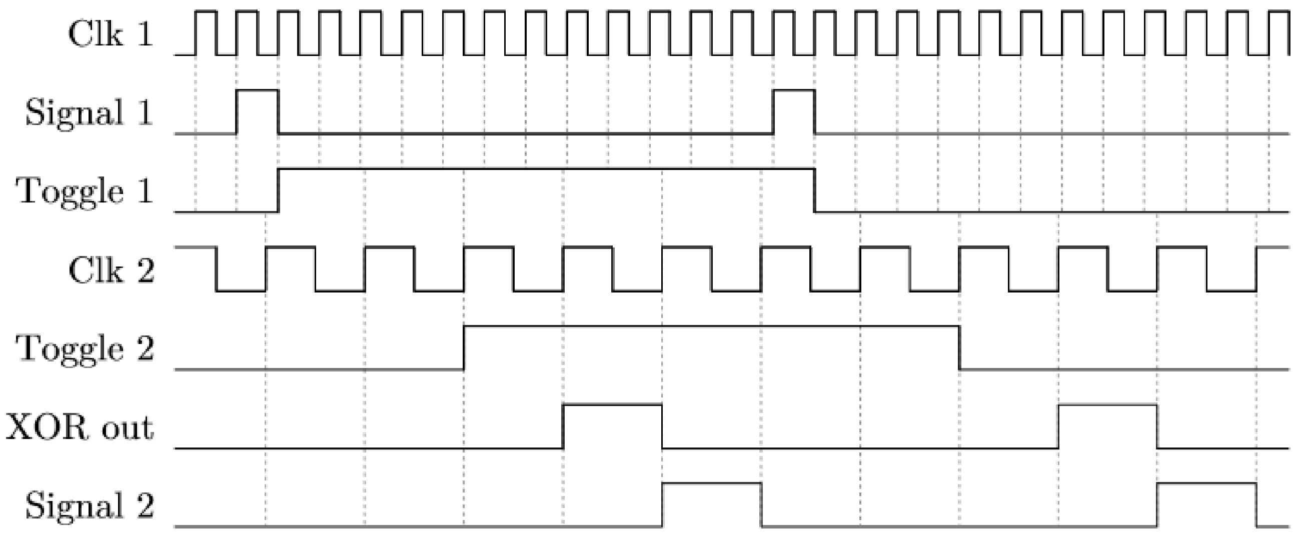
arst_src (optional)

arst_dst (optional)

flag_src

flag_dst





cdc_flag

- ❑ You should add timing constraints → timing analyzer should ignore only the path with cdc_flag.
- ❑ Other paths will be unaffected and still analyzed for timing correctness.