Timing Analysis

Reference:

Static Timing Analysis for Nanometer Designs by J. Bhasker and Rakesh Chadha

Contents

- Foundry Library; Liberty format;
- Gates: Propagation Delays;
- Analyze Flops: Propagation Delay; Setup time; hold Time
- Describe Contamination delay; Recovery time; Removal time
- Describe Clock frequency; Jitter; Skew(source & network latency);
- Analyze Timing Paths; Multi-input path; Clock Budget;
- Describe Multi-Clock; Multi-Cycle Path; False Path; Retiming

Library

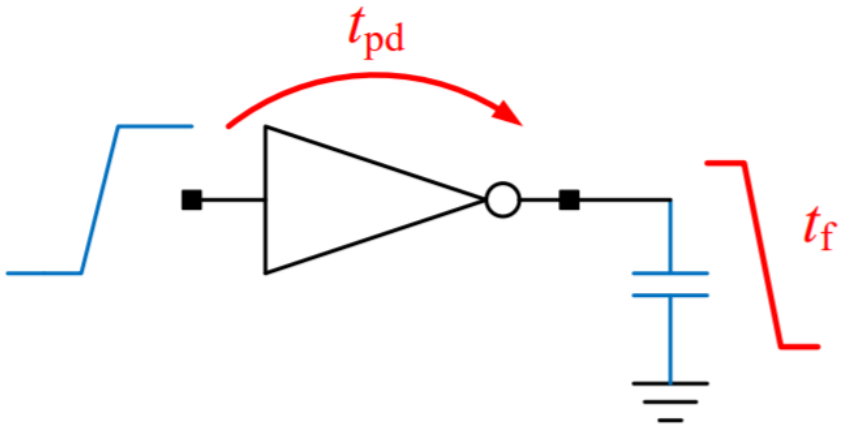
- Standard Cell Library:
- Standard cell library is a collection of well defined and pre-characterized logic cells with multi-drive strength and multi-threshold voltage cells in the form of a predefined standard cell layout.
- Cell Collections:
- In general, a standard cell library contains the following types of cell:
  - All basic and universal gates (like AND, OR, NOT, NAND, NOR, XOR etc)
  - Complex gates (like MUX, HA, FA, Comparators, AOI, OAI etc)
  - Clock tree cells (like Clock buffers, clock inverters etc)
  - Flip flops and latches
  - Delay cells
  - Physical only cells
  - Scannable Flip flops

Library

- File Collections:
- Apart from the standard cells, Standard cell library is delivered with a collection of files which contains all the information required to auto place and route. These files are mainly:
  - LIB files (.lib) - Timing library (LIB or DB) files are generated during the characterization of cells. Library files contain cell delay, power and area information
  - LEF files (.lef) - Physical library (LEF) file is an abstract view of the layout of the cells. LEF file contains the information of cell boundary, Pins inside the cell, location, direction, and metal layer of each pin.
  - Netlist file (.v) - Netlist file is a Verilog file of the standard cell which defines the functionality of a cell
  - GDS file (.gds) - GDS file is the layout of the standard cell
  - SPICE Netlist (.sp) - SPICE netlist is the netlist of cell in SPICE format is used for simulation
  - Model file (.m) - Model file contains the various fabrication parameters of the cell required for SPICE simulation

Liberty

- Liberty files with characterization of timing and power for STA
- Liberty Timing Models (.lib)
  - How do we know the delay through a gate in a logic path?
    - Running SPICE is way too complex.
  - Instead, create a timing model that will simplify the calculation.
- Goal:
  - For every timing arc, calculate:
    - Propagation Delay (tpd)
    - Output transition (trise, tfall)
  - Based on:
    - Input net transition.
    - Output Load Capacitance

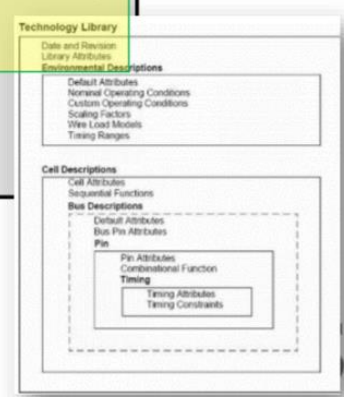


Note that every .lib will provide timing/power/noise information for a single corner, i.e., process, voltage, temperature, RCX, etc.

Liberty

- Timing data of standard cells is provided in the Liberty format.
- Library:
  - General information common to all cells in the library.
  - For example, operating conditions, wire load models, look-up tables
- Cell:
  - Specific information about each standard cell.
  - For example, function, area.
- Pin:
  - Timing
  - Power
  - Capacitance
  - Leakage
  - Functionality

```
library (nameoflibrary) {
... /* Library level simple and complex attributes */

/* Cell definitions */
cell (cell_name) {
   ... /* cell level simple attributes */

   /* pin groups within the cell */
   pin(pin_name) {
       ... /* pin level simple attributes */

       /* timing group within the pin level */
       timing(){
            ... /* timing level simple attributes */ }
       ... /* additional timing groups */

   } /* end of pin */
   ... /* more pin descriptions */
} /* end of cell */
... /* more cells */

} /* end of library */
```



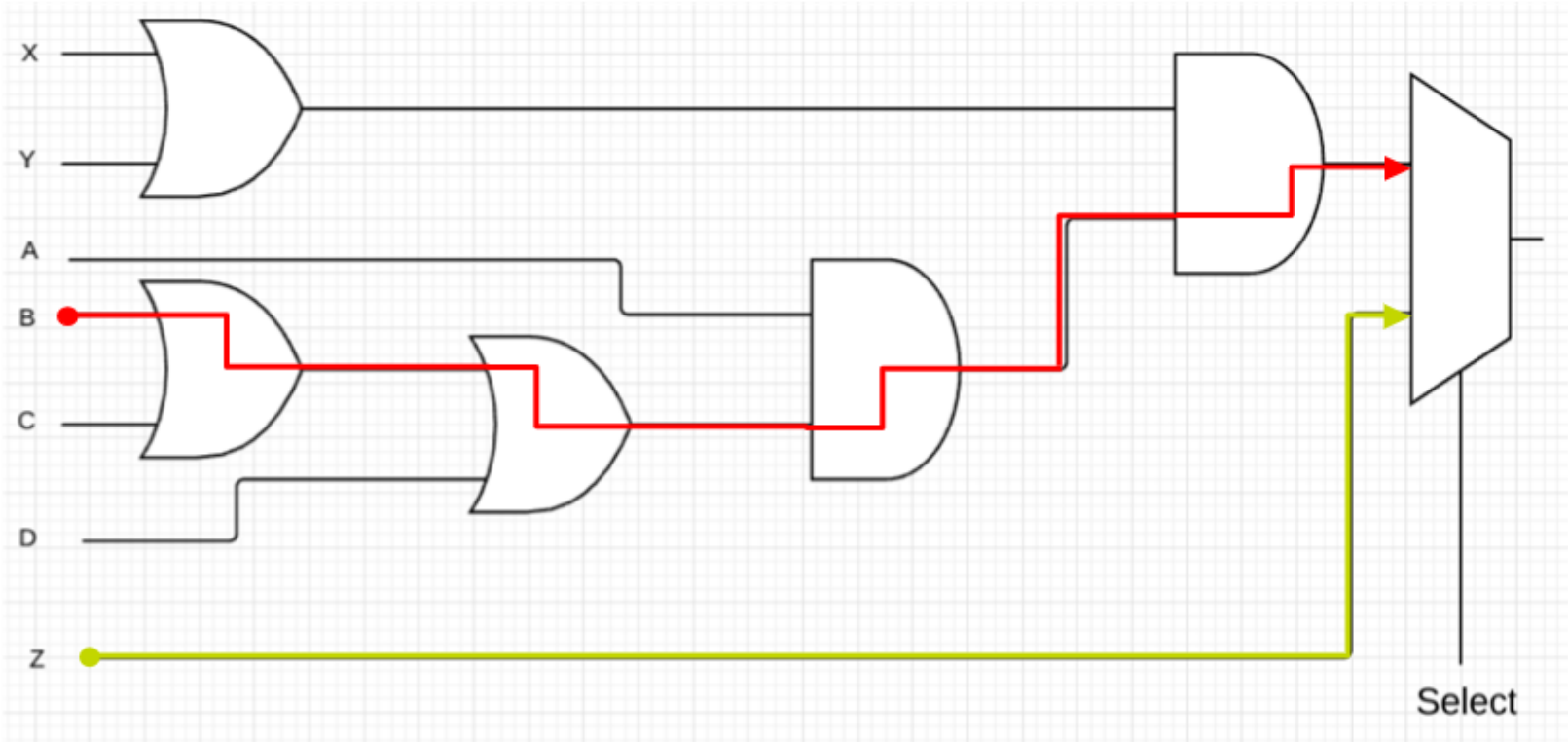
Why Semiconductors Fail

Many Opinions ... Here are some

- Functionally Incorrect
- **Bad Timing Constraints (* or no timing constraints)**
- Didn’t follow manufacturing guidelines (LVS, DRC)
- Test Escapes
- Too much power consumption
- Signal Integrity
- Crosstalk

What affects circuit timing?

- Length of wire
- R and C of wire
- **Logic depth of the path**
- Size of the transistors
- **P**rocess - deposition
- **V**oltage
- **T**emperature

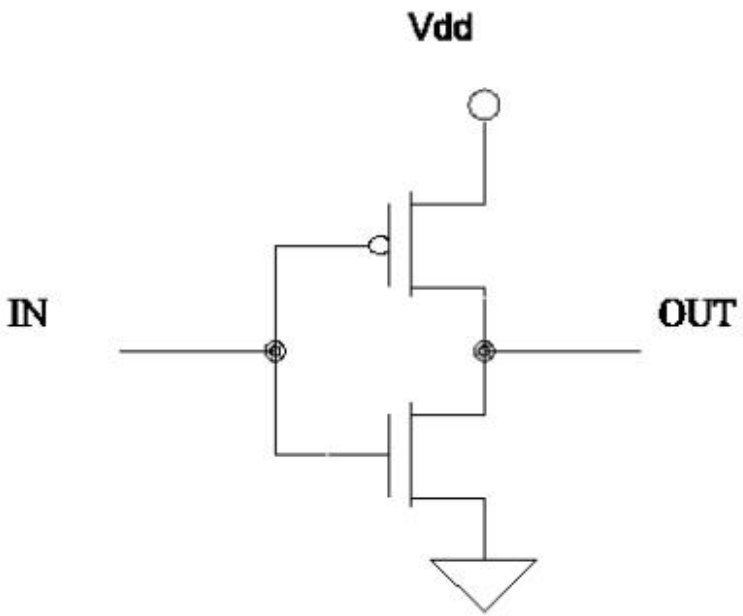


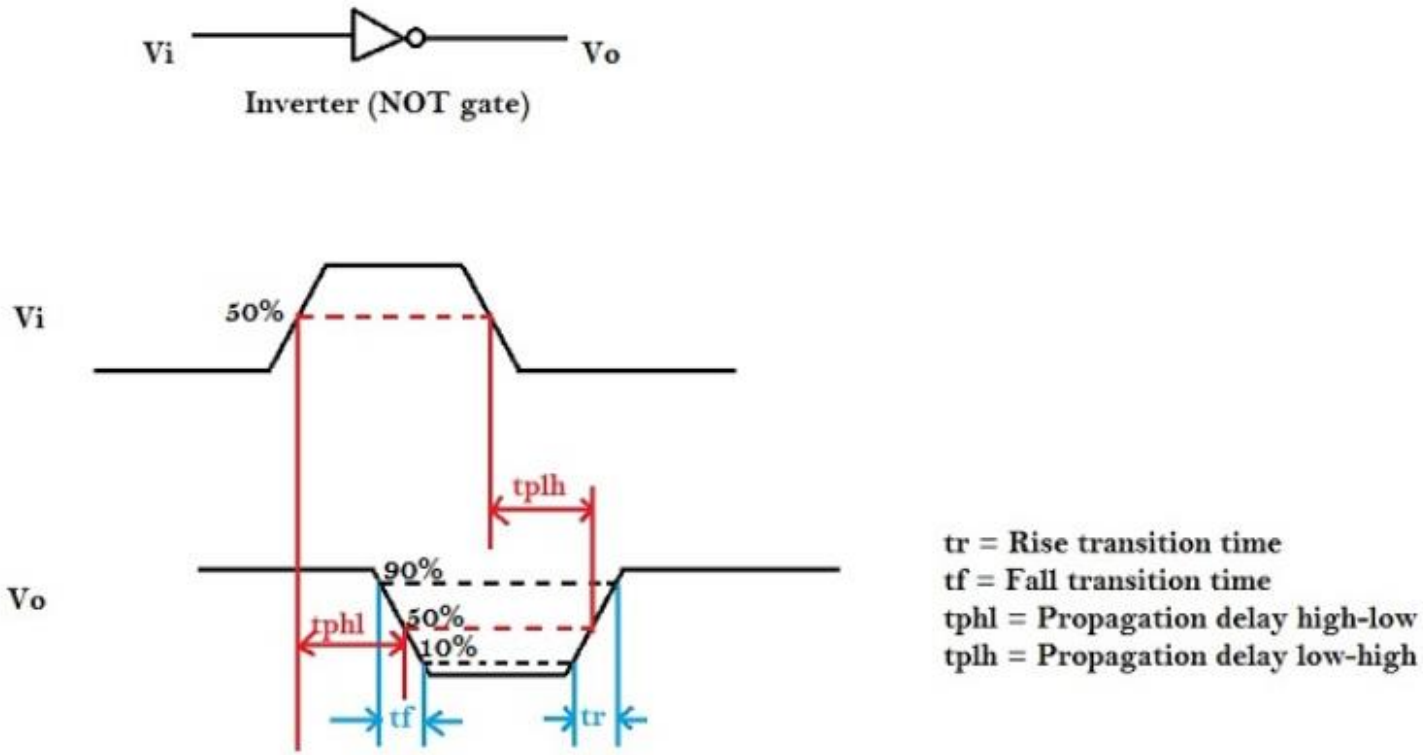
Some logic depicting different depths of paths

“Timing Arc”

It takes a finite time for an input to effect an output change

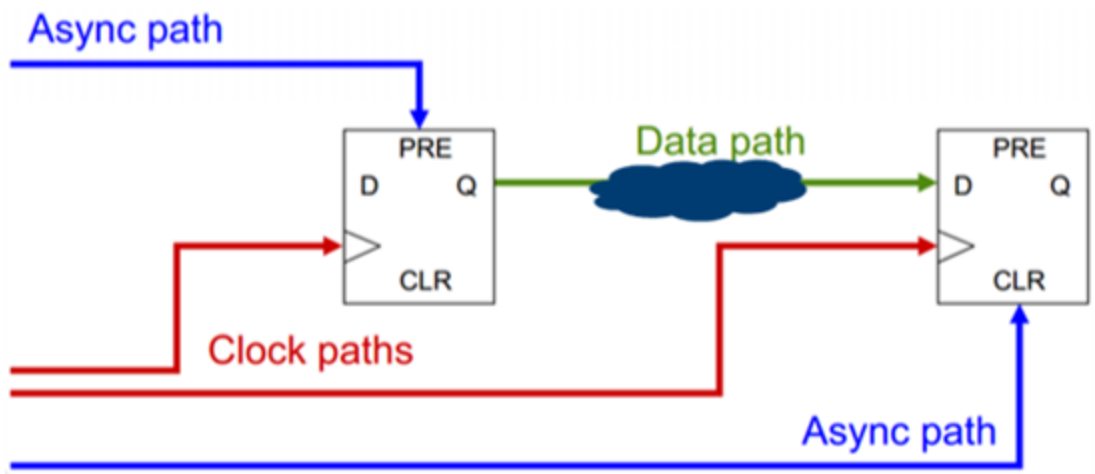
Very often tphl and tplh can be different amounts of time





Terminology

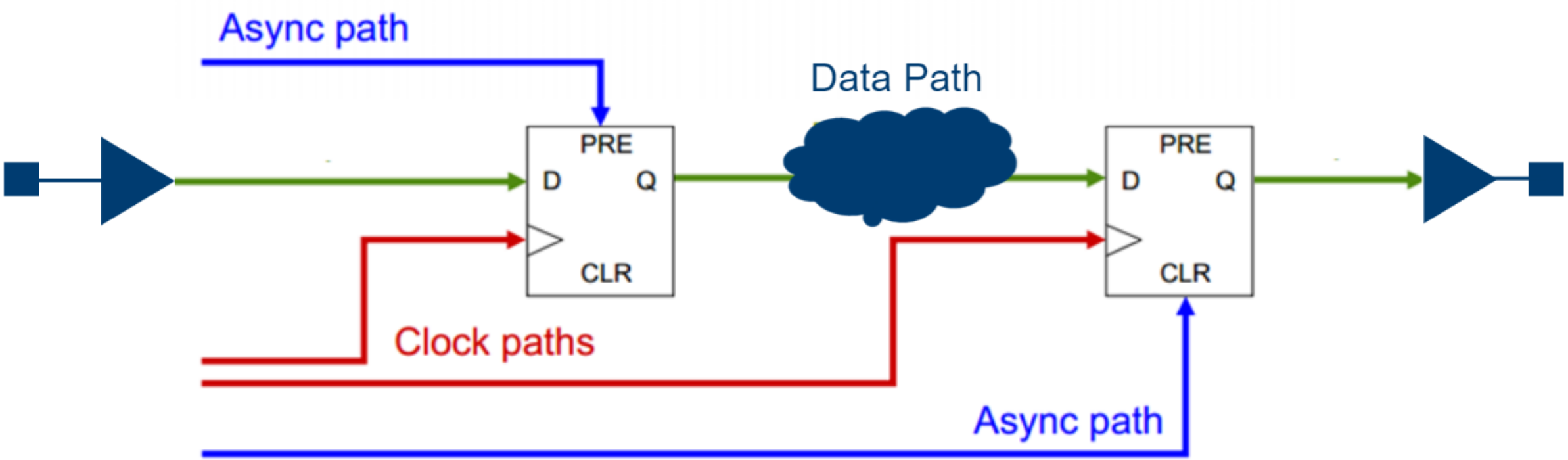
- Data Paths: Paths a signal travels between inputs, sequential elements, and outputs
- Clock Paths: Paths from device ports or internally generated clocks to the clock pins of sequential elements
- Asynchronous Paths: Paths between input port and asynchronous set or clear pin of a sequential element.



Timing Paths

Three Data Path Types

1. From input to a sequential element
2. From sequential element to sequential element
3. From sequential element to output



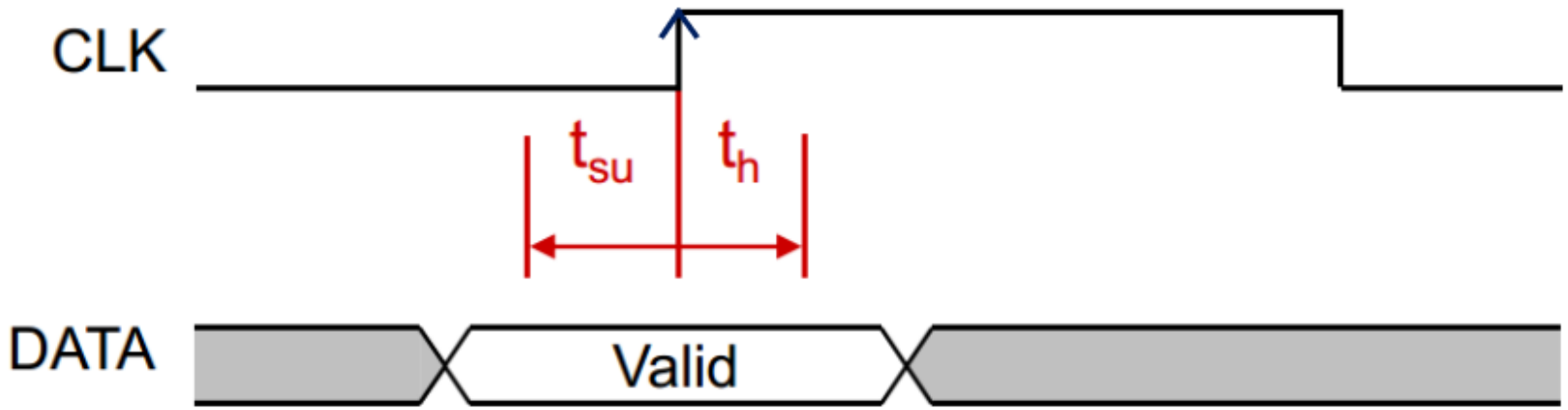
Setup and Hold Time

Setup:

The minimum time the data signal must be stable BEFORE the clock edge

Hold:

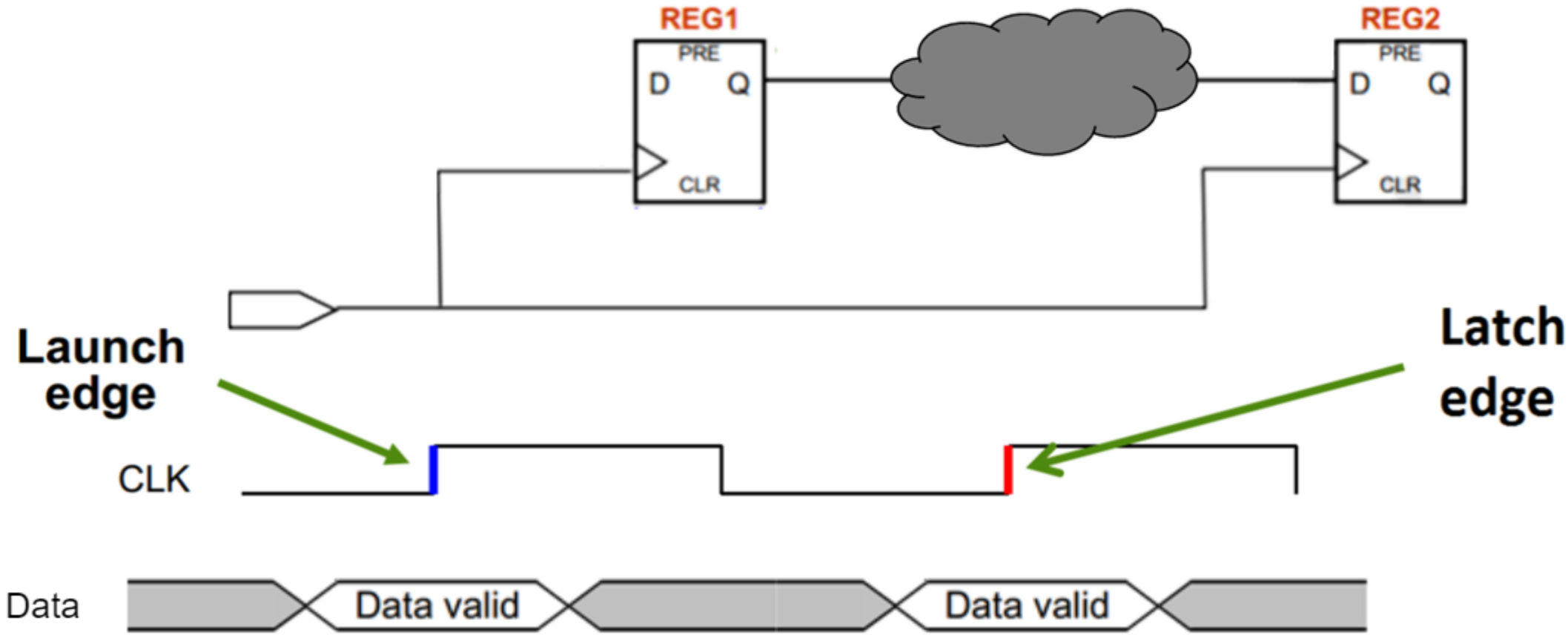
The minimum time the data signal must be stable AFTER the clock edge



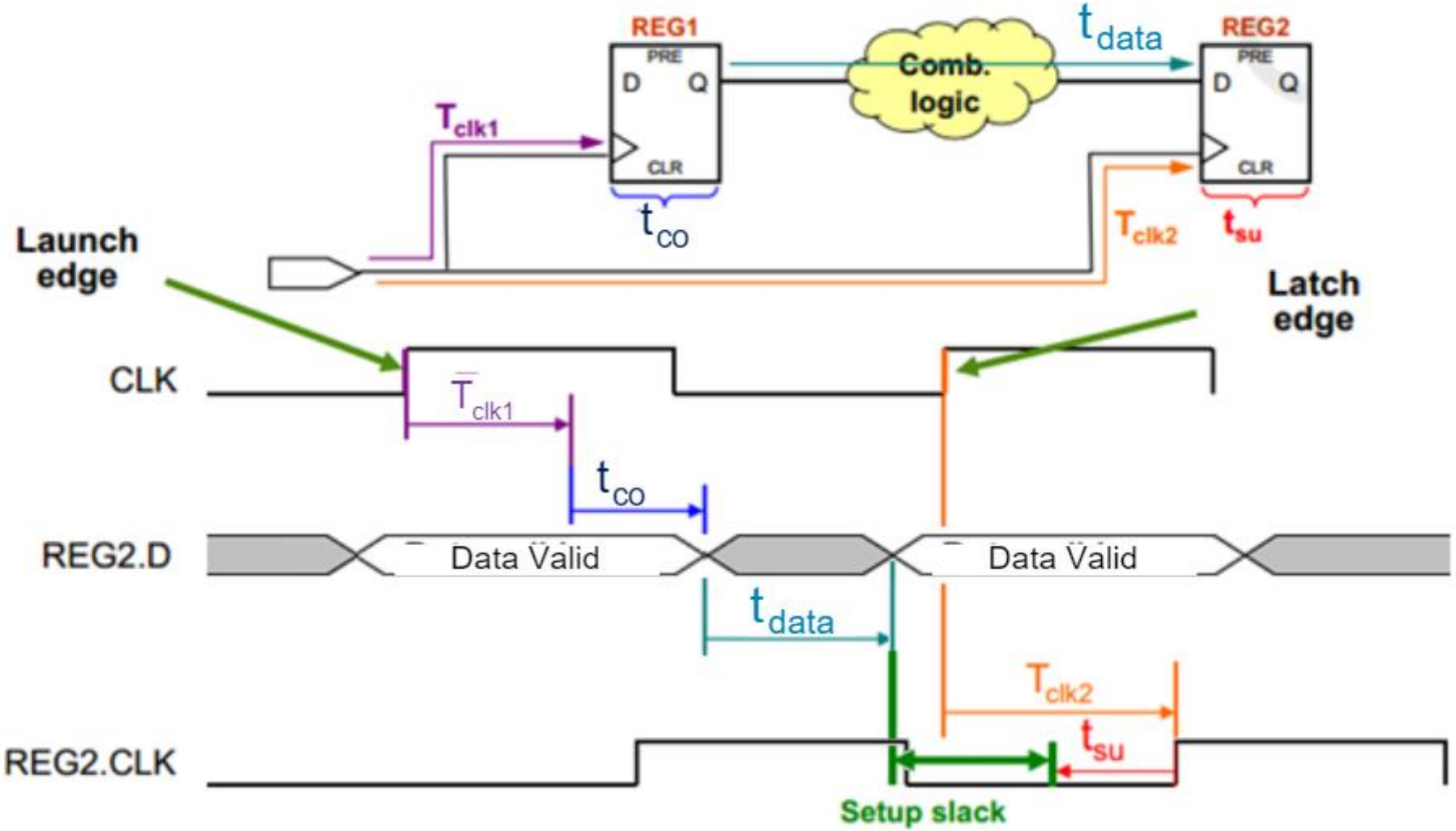
Data Valid Window : the range of time around the clock edge in which data must remain stable to be properly captured

Launch and Latch Edge

- Launch Edge: the clock edge that activates or *launches* the source register
- Latch Edge: the clock edge that latches the data into the destination register



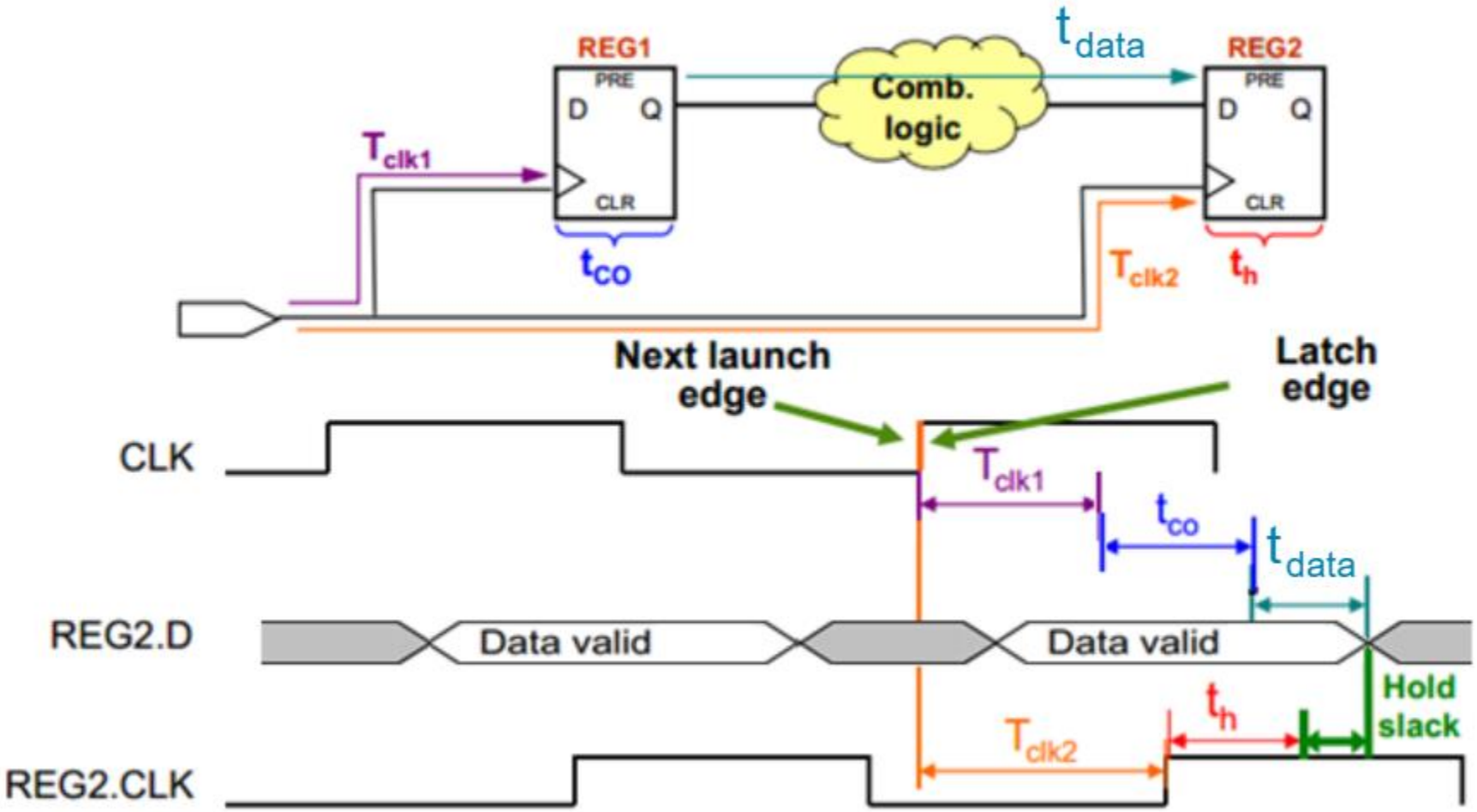
Setup Slack



Dependent on frequency!

Setup slack = minimum data required time – max data arrival time

Hold Slack



NOT dependent on frequency!

Hold slack = minimum data arrival time – max data required time

Test Yourself!

- Setup slack = **(a) min data req time (setup) – max data arrival time**

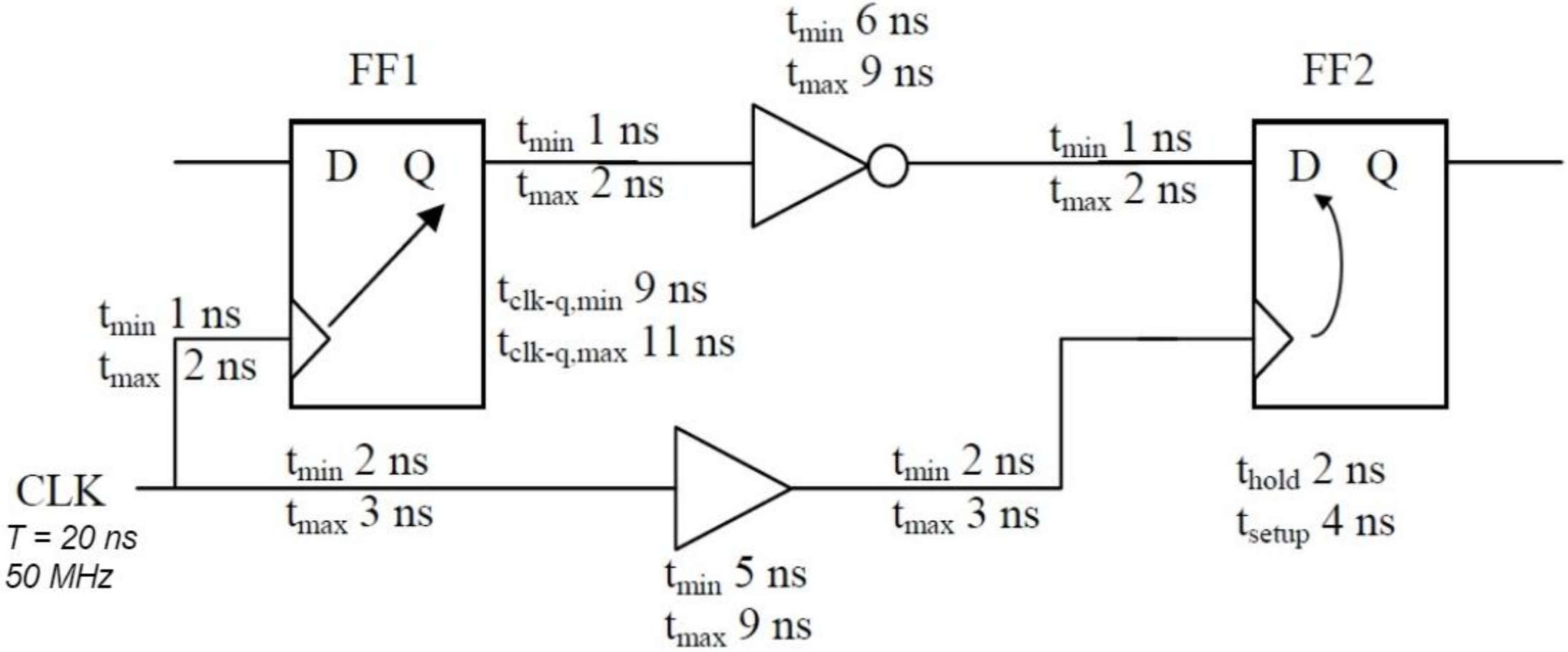
  (b) max data req time (setup) – min data arrival time

  (c) min data req time (hold) – max data req time (setup)

  (d) max data arrival time – min data req time (hold)

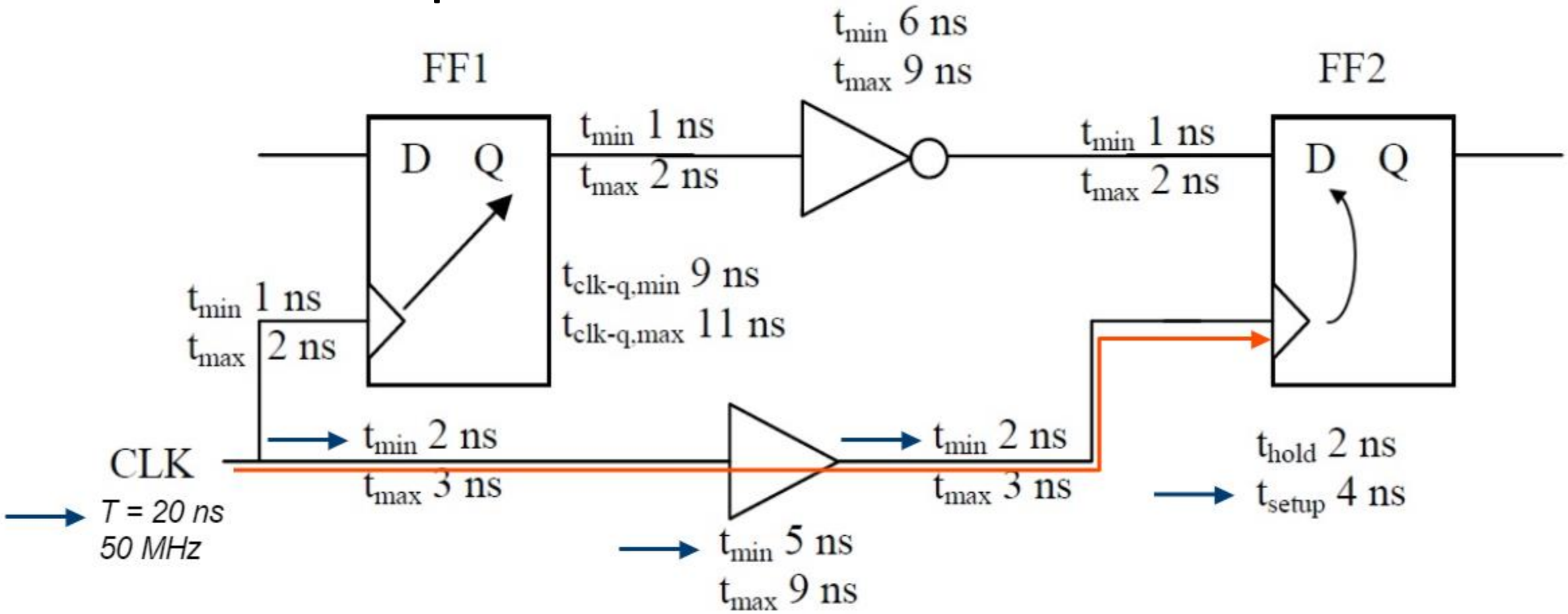
- Slack = *min* <something> - *max* <something>
- Setup = *min* DRT –*max* DAT
- Hold = *min* DAT –*max* DRT

Find the setup slack



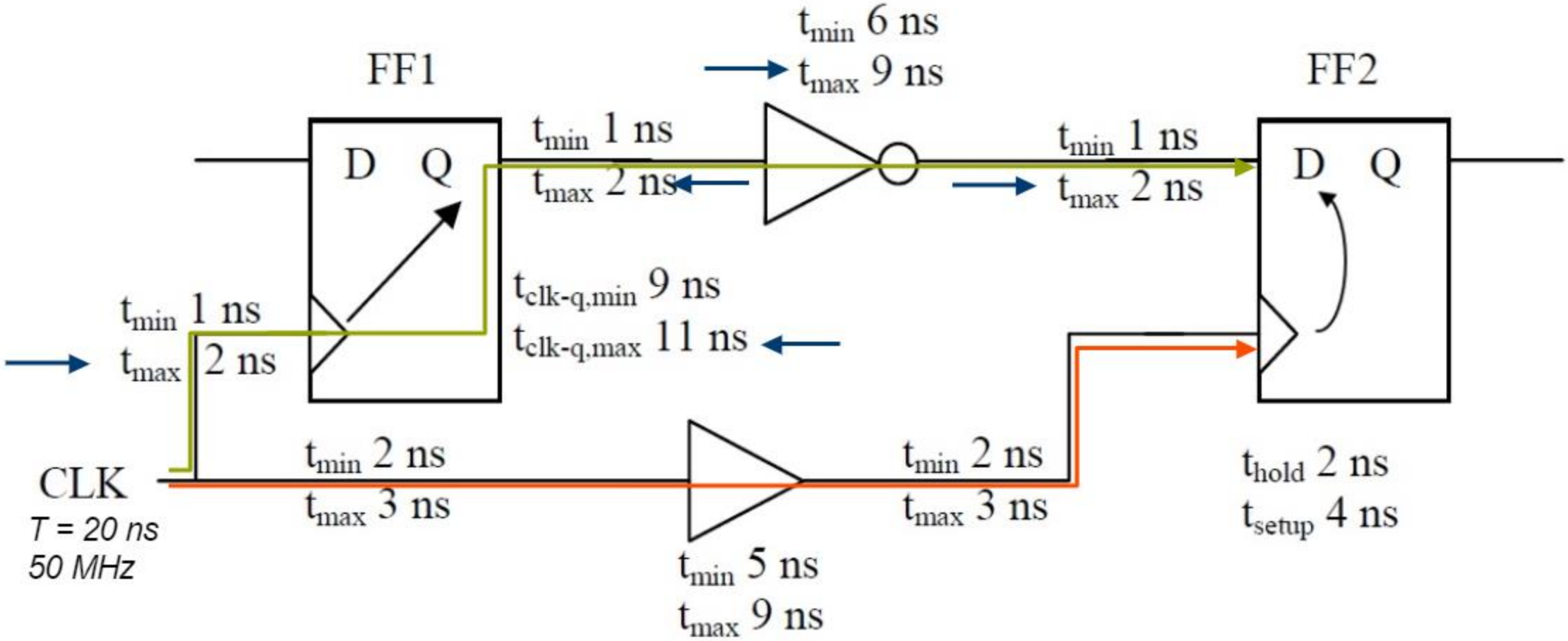
Setup slack: min delay along clock path – max delay along data path =
(min data required time) (max data arrival time)

Find the setup slack



Setup slack: min delay along clock path – max delay along data path = (20+2+5+2-4) -
(min data required time) (max data arrival time)

Find the setup slack



Setup slack: min delay along clock path (min data required time) – max delay along data path (max data arrival time) = (20+2+5+2-4) – (2+11+2+9+2) = 25-26 = -1ns