Static Timing Analysis (STA) – Basic Course

Presenter : Amit Kumar

21/07/2012

Agenda

- Static Timing Analysis – Overview
- STA Checks - Terminology and Definitions
- Timing constraints for STA
- Understanding Timing Exceptions
- Parasitic Formats
- STA Corners
- Fixing the Timing Problems
- Q & A

Static Timing Analysis (STA)- Overview

- Static timing analysis (STA) is an exhaustive method of analyzing, debugging, and validating design performance

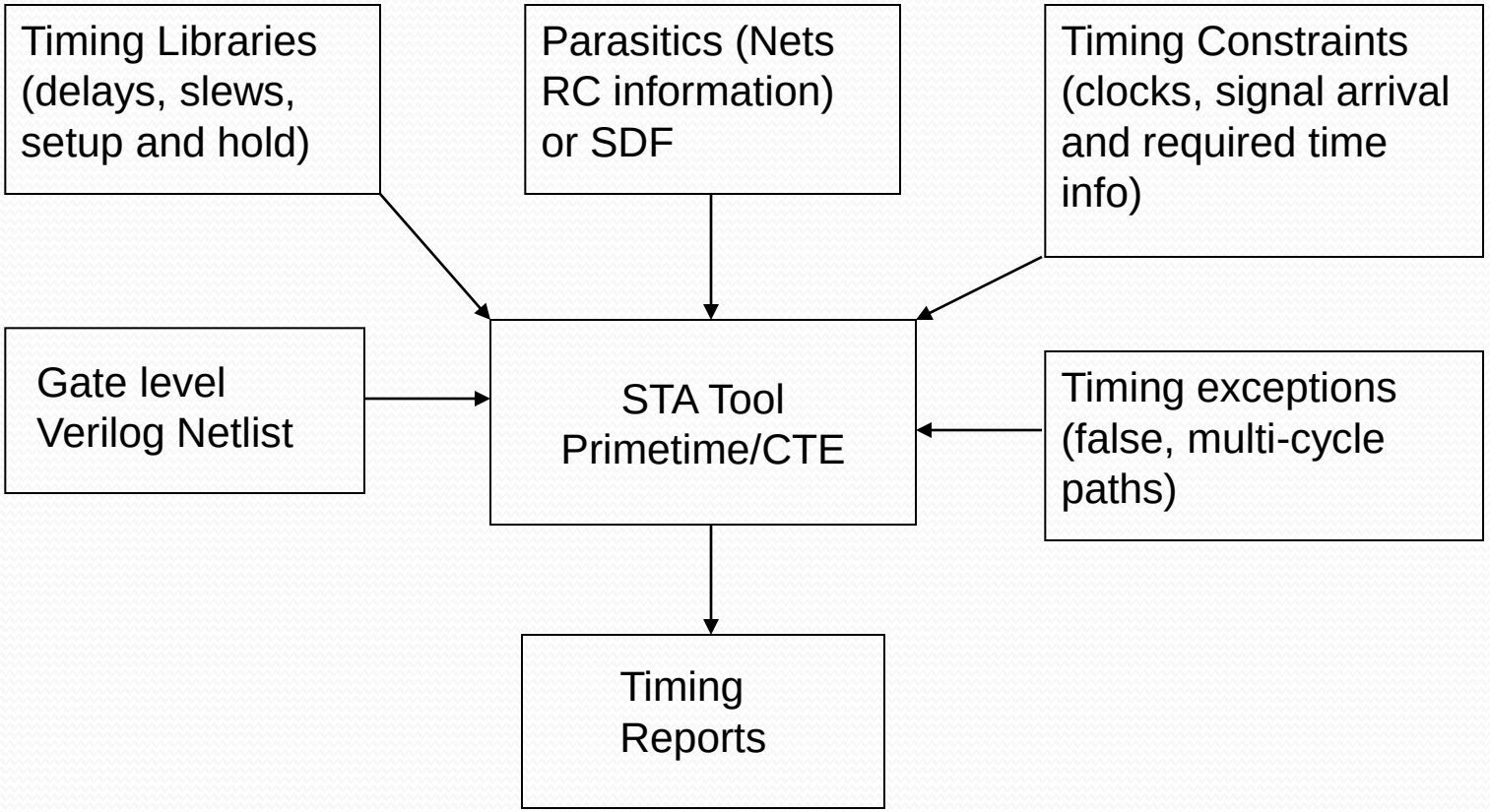
- All the possible paths in the design are checked against timing requirements

- No input vectors required

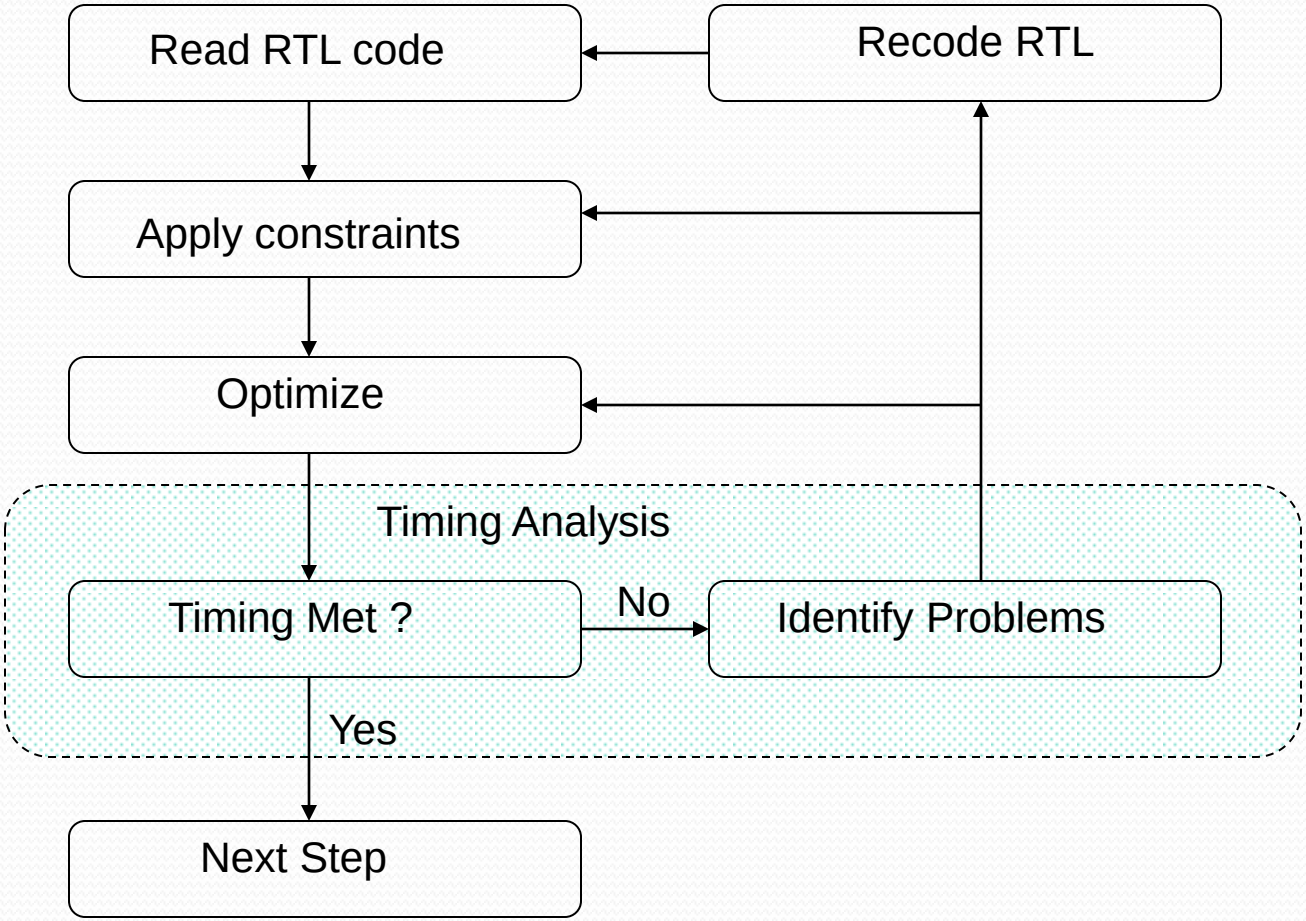
- Very fast as compared to the dynamic timing verification

- Also known as Static timing verification

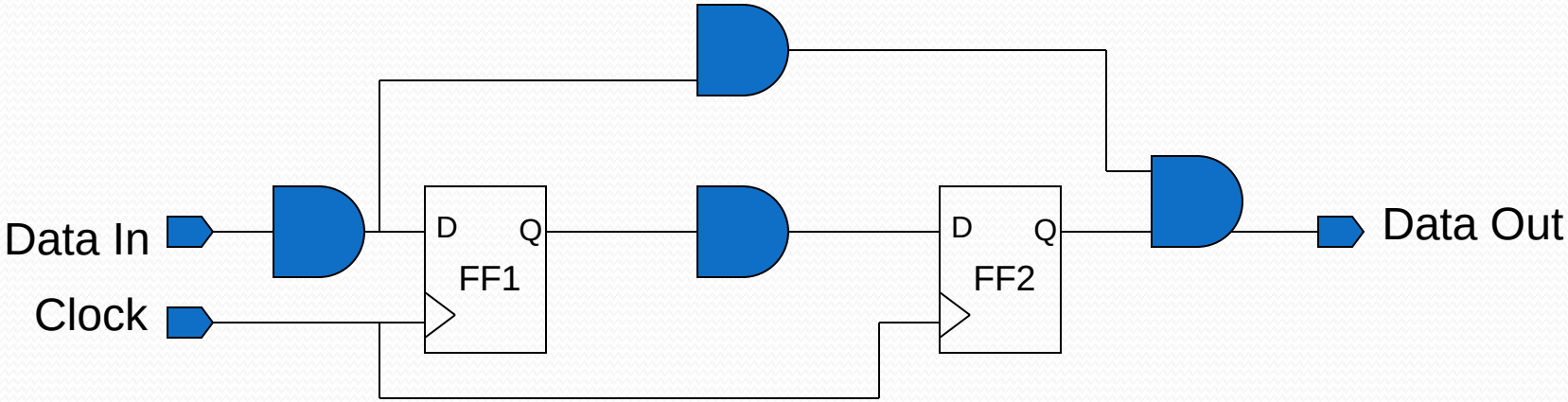
STA Tool inputs and outputs



STA Flow

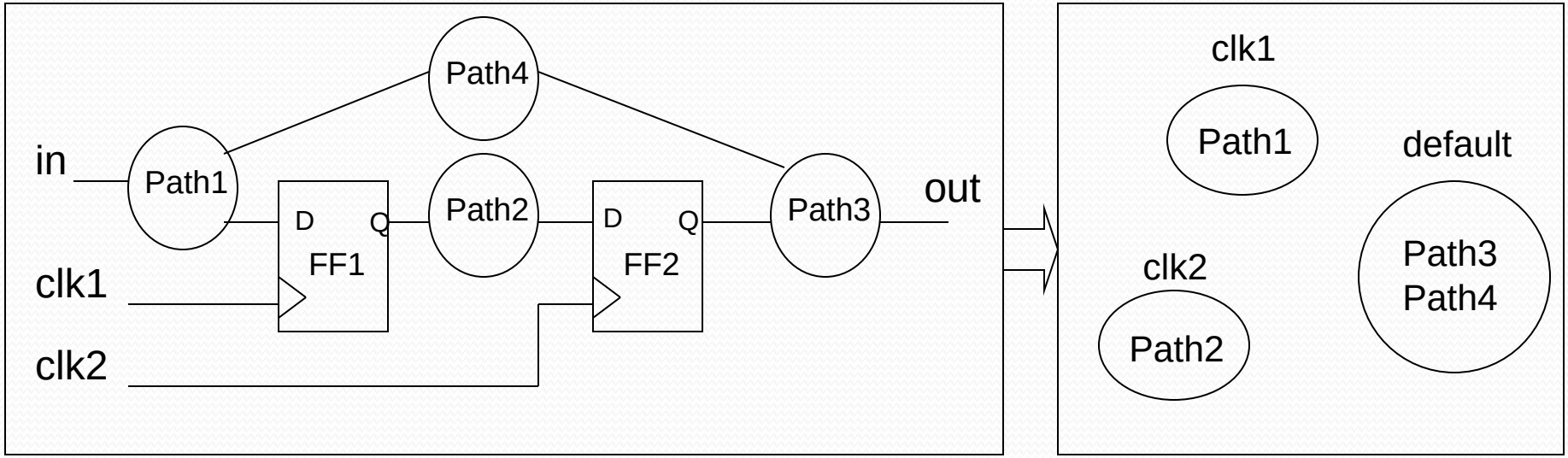

Read RTL code
Recode RTL
Apply constraints
Optimize
Timing Analysis
Timing Met ?
No
Identify Problems
Yes
Next Step

Timing Paths



- Timing tool breaks designs into sets of signal paths
- Each data path has a **start point** and an **end point**
  - Start points : Input ports and Clock pins of sequential devices
  - End points : Output Ports and Data input pins of sequential devices

Organizing Timing Paths into Groups

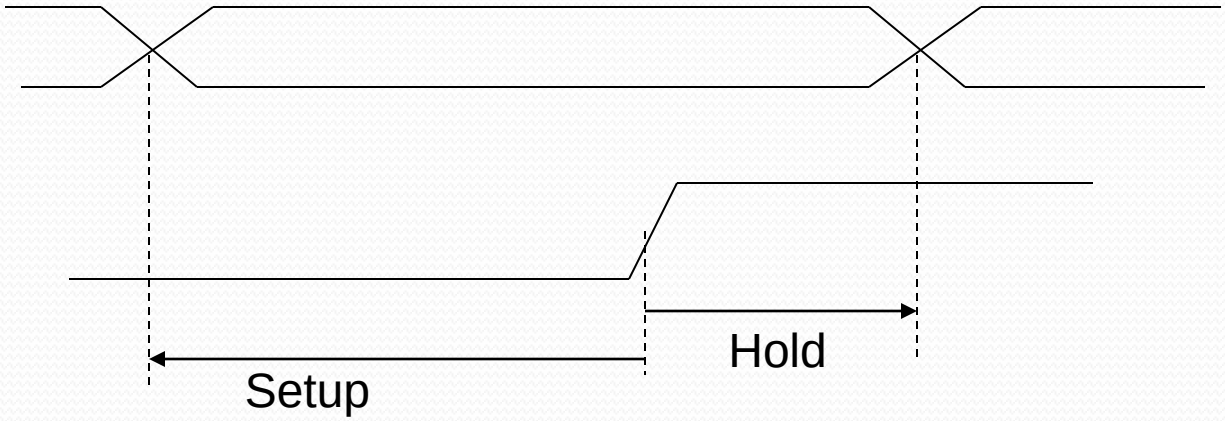


- Paths are grouped according to the clocks controlling their endpoints
- Each clock will be associated with a set of paths called a path group
- The default path group comprises all the paths not associated with any clock.

STA checks: Terminology and Definitions

- Setup and hold
- Recovery and Removal
- Minimum pulse width
- Glitch detection (clock gating)
- Transition violations

Setup and Hold

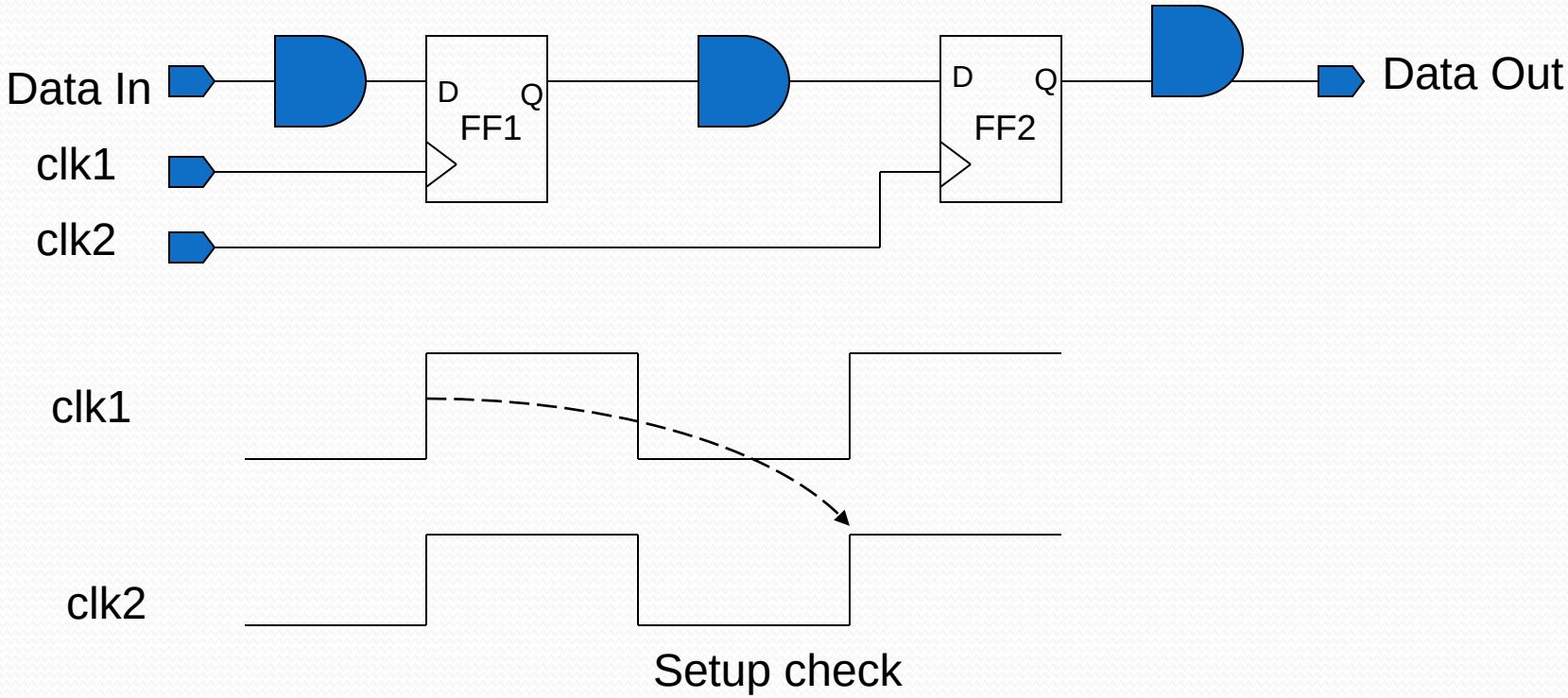


Setup: Minimum time a data input pin of a sequential device must be stable before the clock transition

Hold: Minimum time that a data input pin of a sequential device must be stable after the clock transition

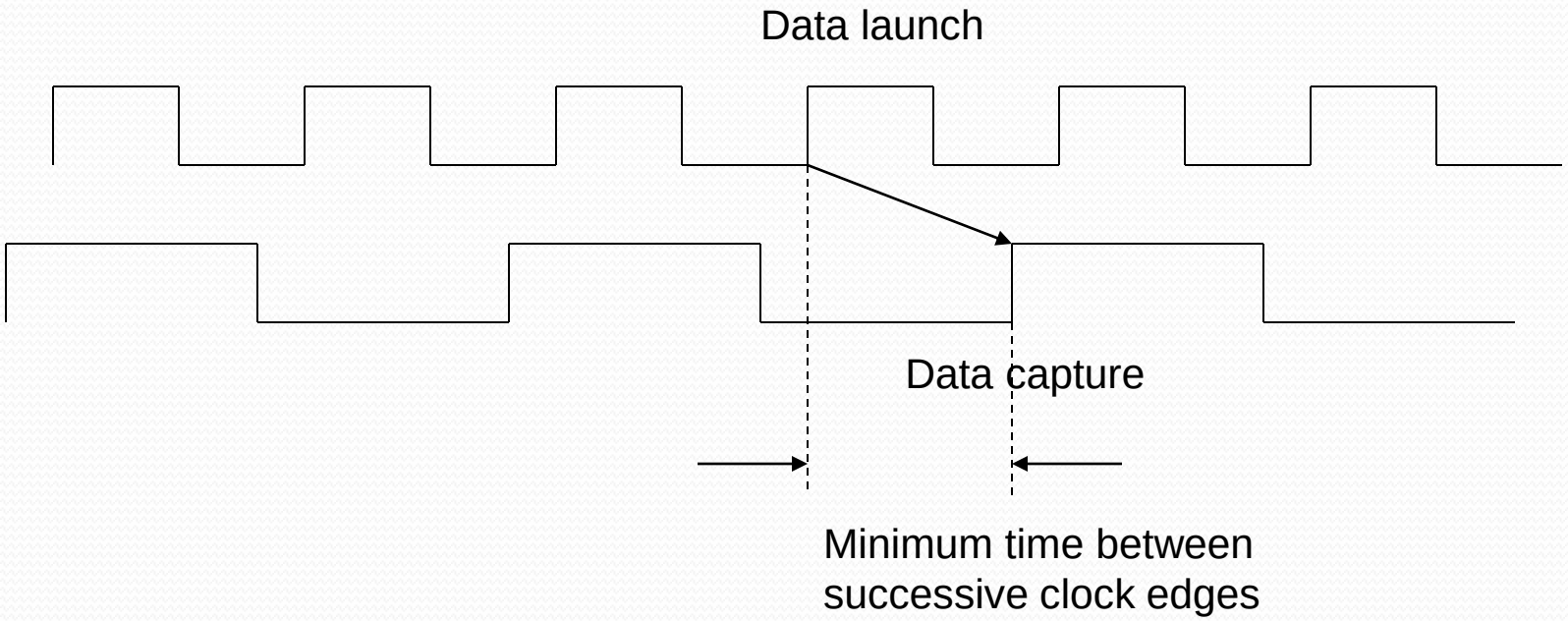
Determining Setup Relation Between Flip-Flops

The default setup check is one clock cycle setup check. Setup relationship is from the first edge of the clock to FF1 to the next (second) edge of the clock to FF2.



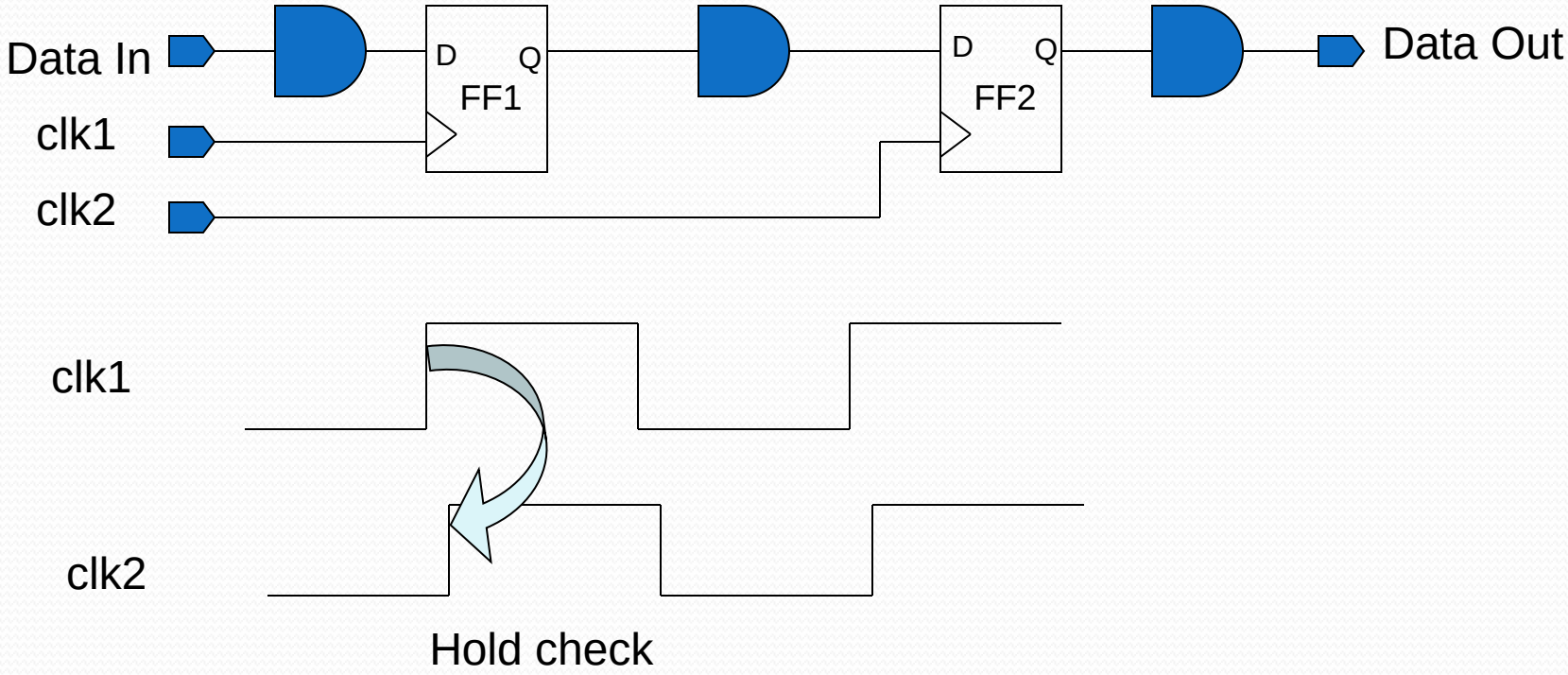
Determining the Setup Time Requirement

Setup check occurs over the **minimum** time between the launching edge and the receiving edge of the clocks



Determining Hold Relation Between Flip-Flops

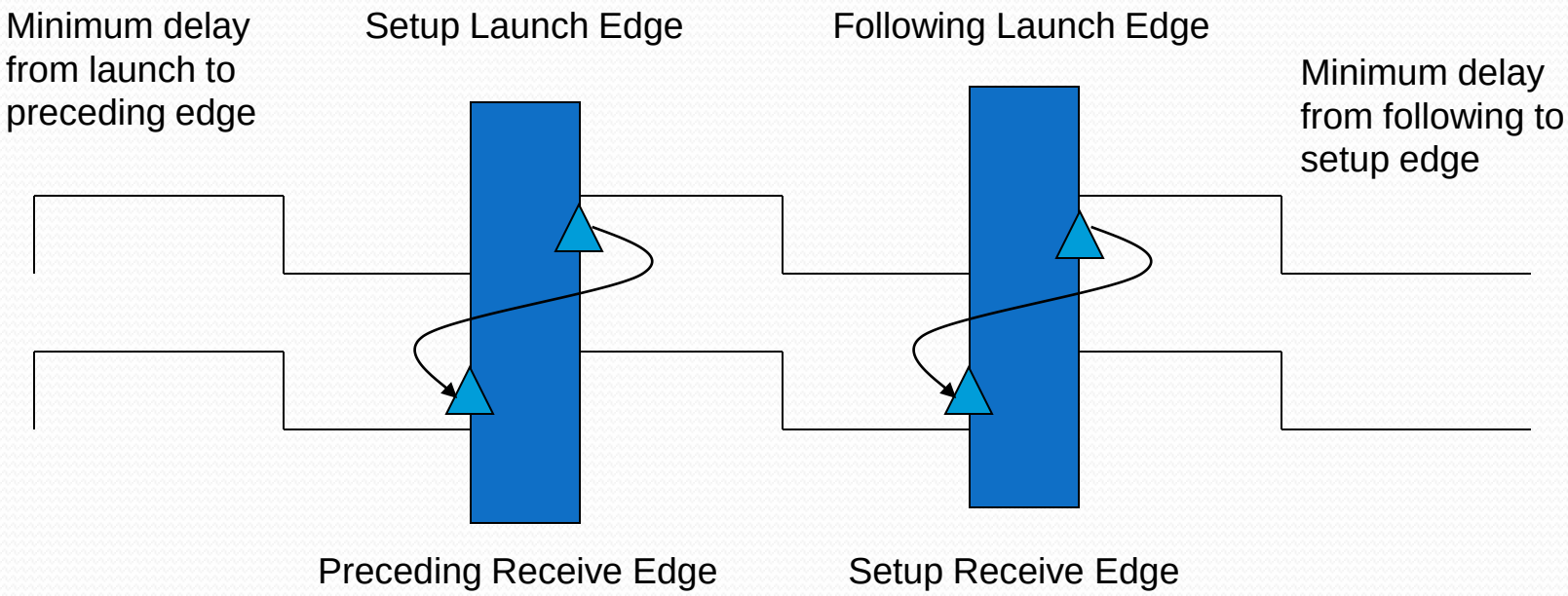
The default hold check is zero clock cycle hold check. Hold relationship is from the first edge of the clock to FF1 to the first edge of the clock to FF2



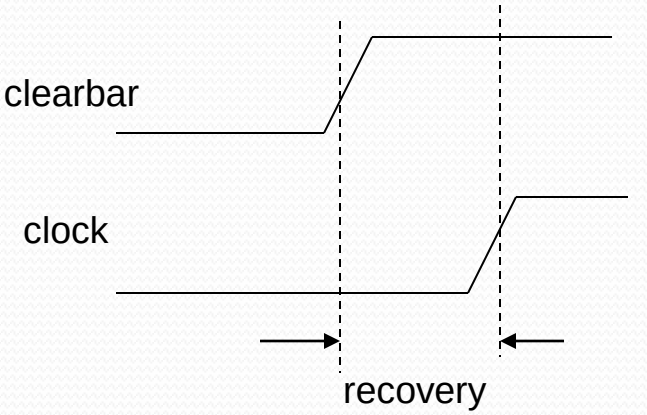
Determining The Hold Time Requirement

Hold time is the largest minimum delay requirement satisfying:

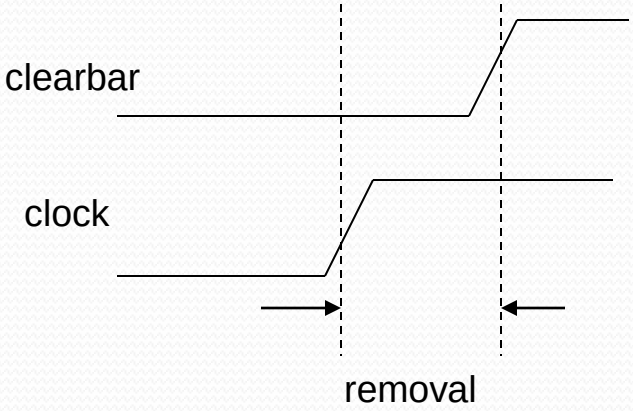
- Data from the following launch edge must not be captured by the setup receiving edge
- Data from the setup launch edge must not be captured by the preceding receiving edge



Recovery And Removal



Recovery:
Minimum time that an asynchronous control input pin must be stable after being de-asserted and before the next clock transition (active-edge).

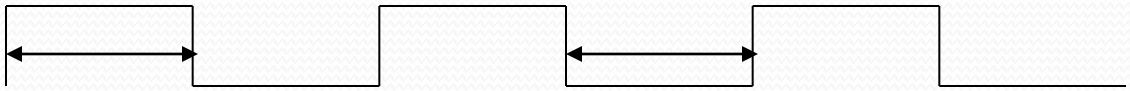


Removal:
Minimum time that an asynchronous control input pin must be stable before being de-asserted and after the previous clock transition (active-edge).

Minimum Clock Pulse Width

• Minimum High pulse width: The amount of time, after the rising edge of a clock, that the clock signal of a clocked device must remain stable.

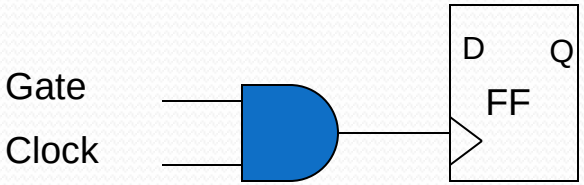
• Minimum Low pulse width: The amount of time, after the falling edge of a clock, that the clock signal of a clocked device must remain stable.

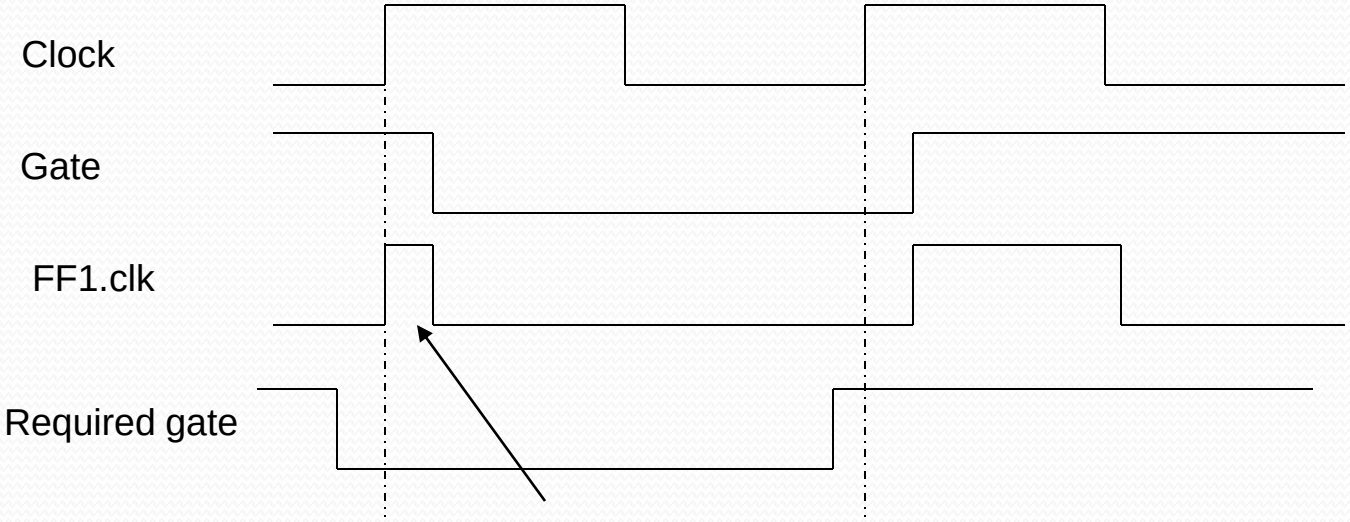


minimum
High pulse width

minimum
Low pulse width

Glitch Detection

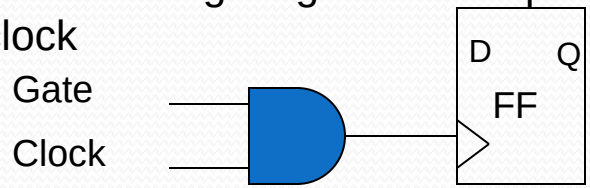


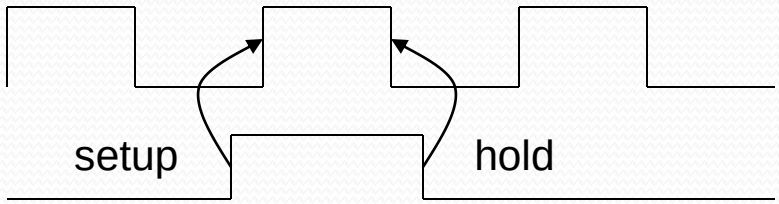


Glitch due to late arrival time of Gate

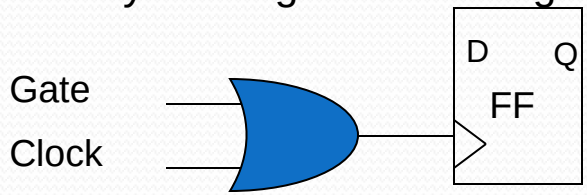
Clock Gating Checks

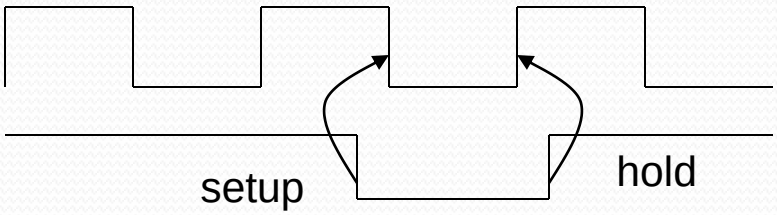
- Setup and hold checks are performed for the gating signal to ensure glitch-free clock
- The clock-gating relationship depends on the functionality of the gate which is gating the clock





Gating signal should only change when the clock is in low state





Gating signal should only change when the clock is in high state

Timing Constraints for STA

Timing constraints required for synthesis/STA to make sure the intended frequency performance is achieved.

- Chip-level STA team provides the initial constraints for synthesis/STA

- Module constraints include the following:-

  - Clock definitions (create_clock, create_generated_clock)

  - Input arrival times for the input ports (set_input_delay)

  - Output departure times for the output ports (set_output_delay)

  - Global constraints (set_max_transition, set_max_capacitance, set_input_transition, set_load)

Clock Definitions

A typical clock definition:

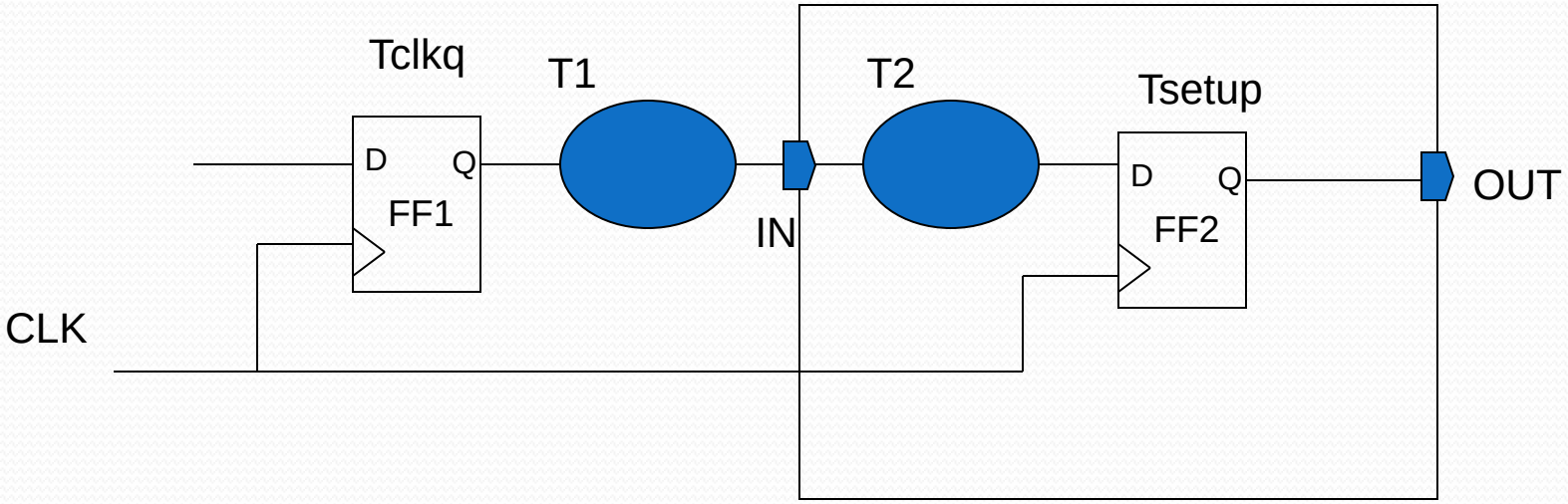
```
create_clock [get_ports core_clk] -period 10.0 -waveform [list 0 5.0]
set_clock_uncertainty 0.5 [get_clocks core_clk]
set_clock_latency $latency [get_clocks core_clk]
set_clock_transition 0.2 [get_clocks core_clk]
```

A virtual clock can also be defined like:

```
create_clock -name v_core_clk -period 10.0 -waveform [list 0 5.0]
```

- The virtual clock is used just as reference clock, defined with same or multiple of the real clock frequency and is not actually propagated in the design.
- This clock is defined for ease of constraints management, used to constrain the arrival for inputs and departure times for outputs.

Modeling Input Arrival Time (set_input_delay)

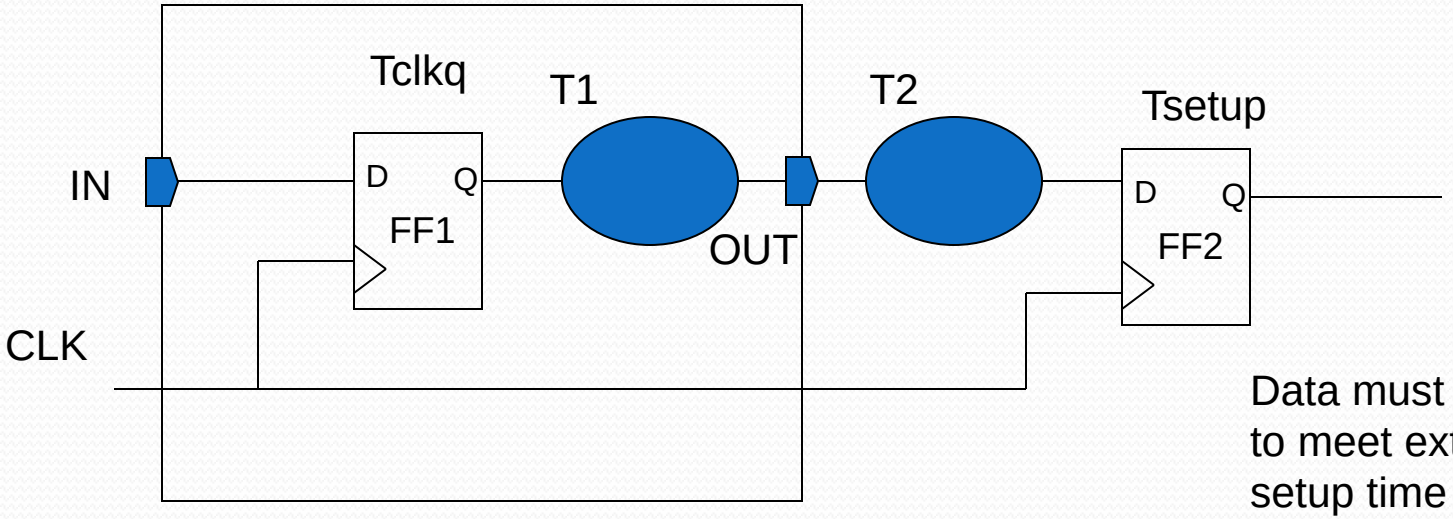


The input data (IN) arrives after Tclkq+ T1

This is the amount of time to specify as the input delay, specified as:

set_input_delay [expr $Tclkq + $T1] –clock CLK [get_ports IN]

Modeling Output Arrival Time (set_output_delay)



The external logic setup time with respect to CLK is T2 + Tsetup

This is the amount of time to specify as the output delay, specified as:

set_output_delay [expr $T2 + $Tsetup] –clock CLK [get_ports OUT]

Global Constraints

Global module constraints include the following:

- set_max_capacitance is mostly specified on input ports to limit the loading on the preceding driver

Syntax: set_max_capacitance <value> [all_inputs]

- set_input_transition is the transition time specified on the input ports.

Syntax: set_input_transition <value> [all_inputs]

- set_driving_cell models the driver on the input ports.

Syntax: set_driving_cell -lib_cell $driving_cell –library $driving_cell_lib -input_transition_rise 0.5 -input_transition_fall 0.5 $input

- set_max_transition is mostly applied on the output ports to control output transition times, leaving some margin for degradation along the interconnect

Syntax: set_max_transition <value> [all_outputs]

- set_load is an estimate of the interconnect loading on the driver

Syntax: set_load <value> [all_outputs]

Understanding Timing Exceptions

- Default single cycle timing for all paths in the design
- False Path, Multi cycle path, delay limits and case analysis
- As an exception, need to define false (invalid) paths or multi-cycle paths
- Specific/generalized timing exception commands, following an order of precedence as follows:-

1. *command* -from *pin* -to *pin*
2. *command* -from *clock* -to *pin*
3. *command* -from *pin* -to *clock*
4. *command* -from *pin*
5. *command* -to *pin*
6. *command* -from *clock* -to *clock*
7. *command* -from *clock*
8. *command* -to *clock*

- Following rules summarize interaction of timing exception commands
  1. General *set_false_path* commands override specific *set_multicycle_path* commands

False Paths

The paths which are not valid.

Syntax include the following generic options:-

-from list: Specifies the startpoint, usually a clock, a primary input or inout port, a register or clock pin of a register

-through list: Specifies intermediate points.

-to list: Specifies the endpoint, usually a clock, a primary output or inout port, a register or data pin of a register

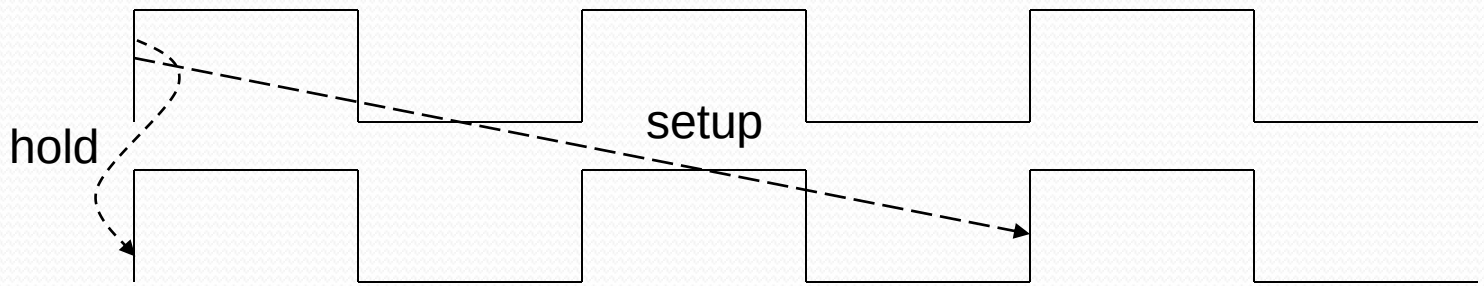
Multi-cycle Paths

- Paths requiring more than once cycle to propagate.
- Multiplier for setup and/or hold checks

- **Multi-cycle path clocked by identical clocks**

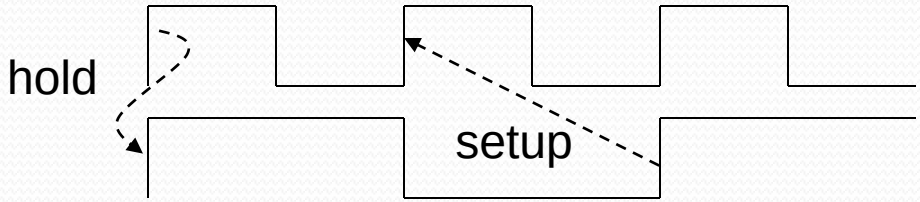
Example:

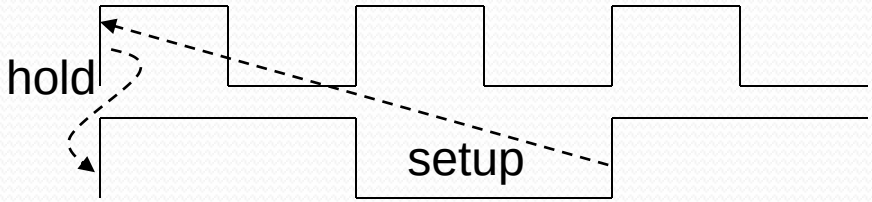
- set_multicycle_path 2 –**setup** -from <clock_name> -to <clock_name>
- set_multicycle_path 1 –**hold** -from <clock_name> -to <clock_name>



Multi-cycle Paths Contd.

- **Multi-cycle path between multi-frequency domain clocks**
  - **Paths from fast frequency clock to slow frequency clock**

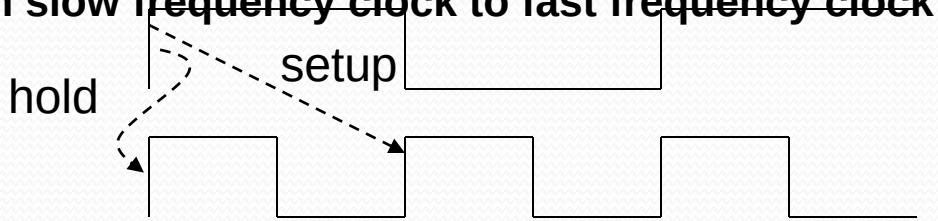


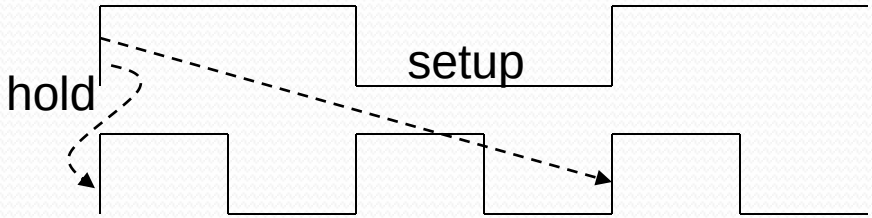


Example:
set_multicycle_path 2 -setup –**start** –from \
<start_clock>
set_multicycle_path 1 -hold –**start** –from \
<start_clock>

Multi-cycle Paths Contd.

- **Multi-cycle path between multi-frequency domain clocks**
  - **Paths from slow frequency clock to fast frequency clock**





Example 2:
set_multicycle_path 2 -setup –**end** –from \
<start_clock>
set_multicycle_path 1 -hold –**end** –from \
<start_clock>

Setting delay limits

set_max_delay:

Specifies a maximum delay for timing paths
Can be used to set a target delay for output ports in combinational logic
Can be used as a substitute for the set_multicycle_path command

set_min_delay:

Specifies a minimum delay for timing paths
Can be used to override the default hold relation for paths

Reporting Exceptions

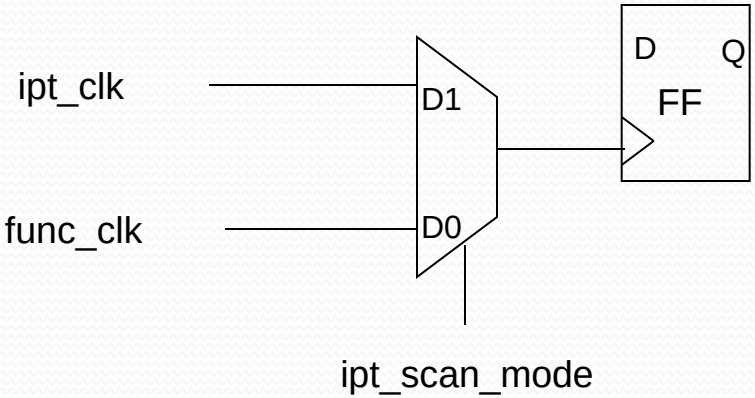
report_exceptions command

Case Analysis

Set constants on pins/ports to exercise only valid timing-paths

- The block can have different set of constraints for different configurations - Use Case Analysis to exercise only those paths which are relevant for that configuration. eg. functional mode and scan mode

- Select a particular clock if two or more clocks are mux-ed. eg. set the select pin appropriately to select the test-clock for scan mode and functional clock for functional mode



Timing Loops

Sequential Loop:

simple or gated q-to-d feedback loops, tool has no problem in timing paths to and from same flop.

Combinational Loop:

Combinational feedback loop is handled differently. Tool breaks combo loops at a timing arc.

Timing Reports

- Analysis coverage - Can be generated using report_analysis_coverage command. Look if the number of untested paths is large. Look for the reason of the paths being untested. Ideally, analysis coverage should be 100%.

- Unconstrained endpoints - Reported using the command check_timing –type endpoints. Unconstrained endpoints can also be due to defined false-paths and case-analysis

- Unconstrained inputs/outputs - Reported using check_timing –type input/output

- Registers with no clock - Reported using check_timing –type clocks. The clocks to the registers may be blocked by the case-analysis and disable-timing statements.

- Registers clocked by multiple clock sources - Reported using check_timing –type multiple_clocks.

- Timing loops in the design - Reported using check_timing –loops (primetime). CTE generates report during building timing graphs.

- Timing Violations – Setup, Hold, Recovery, Removal, Load, slew.

Clock Gating Checks

- Automatically performed for combinational gates other than inverters or buffers
- Not performed between two clock signals
- Setup and Hold checks on gates
- Setup check ensures that the controlling data signals are stable before clock becomes active
- Hold check ensures that the controlling data signals are stable when the clock is active

Pre-layout STA

- Before APR is done, during logical and physical synthesis.
- Wire-loads are picked up from the wireload-model/parasitics from Physical Synthesis tool
- Delays for the cell are calculated from the library based upon the load on the net
- Delays for the net is calculated from the R & C values on the net
- Clocks are treated as ideal clocks since the clock-tree is not inserted

Post-layout STA

- After clock tree balancing and routing is done.

- The cell and net delay information is either calculated by tool using interconnect parasitics (RC information of all the nets in design) information or supplied by back-end tool in the form of SDF.

- The cell and net delays can be calculated by the back-end tools and this information is supplied in the form of SDF to Primetime for timing-analysis.

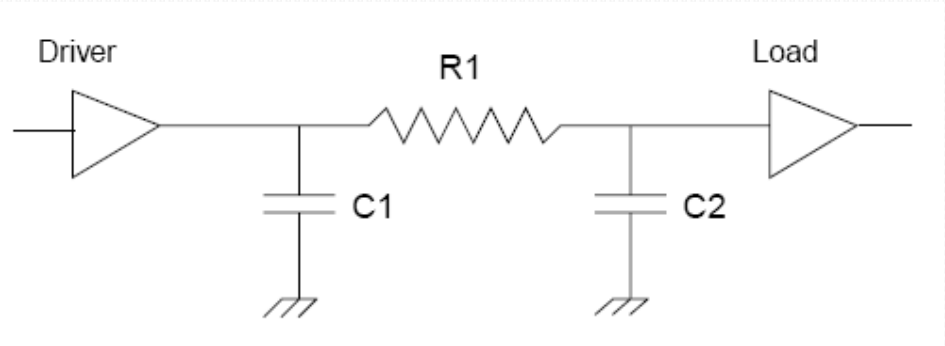
- Alternately, the extraction tools can extract the parasitics for the nets in the design - These parasitics can be back-annotated in Primetime and Primetime uses this information to calculate the cell and net delays.

- Clocks are propagated using set_propagated_clock command and the actual clock delay to each flip-flop is calculated

Parasitics Formats

RSPF - Reduced Standard Parasitic Format

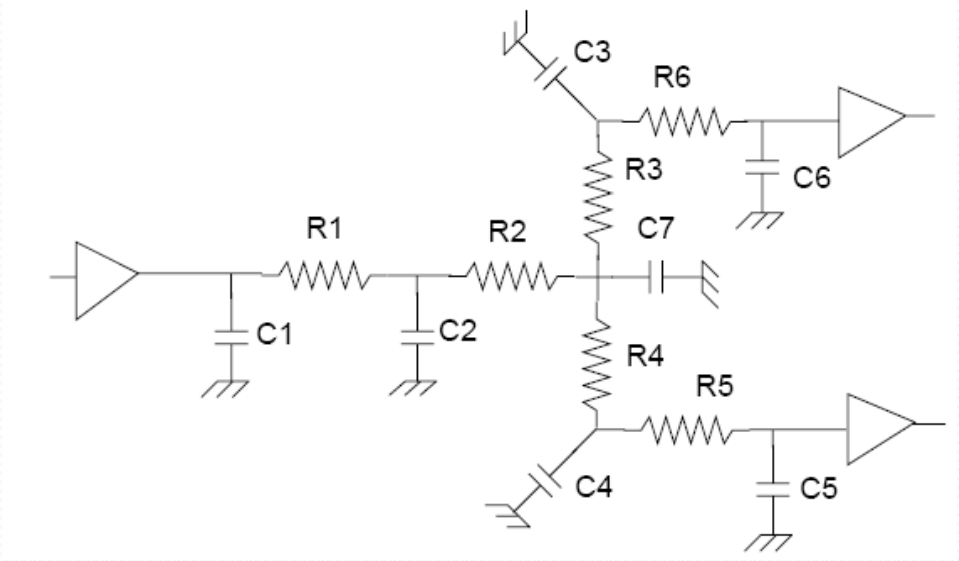
- This format is the proprietary of Cadence
- Each net is represented by a RC-Pi network as shown



Parasitics Formats Contd.

DSPF - Detailed Standard Parasitic Format

- Represents the more detailed parasitics for each net
- Each physical segment of the route is represented by a resistance and capacitance
- This format is proprietary of Cadence



SPEF - Standard Parasitics Exchange Format

- IEEE Format
- Same representation as that provided by DSPF
- Smaller database (files)

STA Corners

- Transistors are the basic building block of any SoC. Interconnects connect them properly to make them work in a desired way. The properties of both these can vary depending upon the condition.
- Static Timing Analysis of any SoC is done in all possible combination of variations seen in Transistors and Interconnects. Each combination is called, STA CORNER.
- Each STA CORNER constitutes Four **parameters** which affect transistors and interconnects behavior:

  - Process Variation
  - Temperature Variation
  - Voltage Variations
  - Interconnect Extraction Variation

STA Corners (continued)

- **Process Variation:**
  - Pin Capacitance Variations of Transistors/Logic Cells
  - While silicon chip is working, Pin caps can vary from lower limit to upper limit put on them for various logic cells (in modeled libraries).
  - Two Process Variations used for STA: bcs & wcs
  - **Bcs:** pin caps are minimum, best case situation, transistors/logic cells are said to be faster as they need to drive a smaller output load.
  - **Wcs:** pin caps are maximum, worst case situation, transistors/logic cells are said to be slower as they need to drive a larger output load.

- **Temperature Variation:**
  - Affects both Transistors and interconnects' behaviour.
  - Transistor's threshold voltage is a function of Temperature. So, is the Drain current. Hence, driving strength of a transistor/logic cell depends on Temp.
  - Interconnect's resistivity, length, area of cross section etc. depends on the Temp. So, is the Resistance.

STA Corners (continued)

- **Voltage Variation:**

  - Transistor/Logic cell parameter.
  - Interconnect R & C are independent of Supply Voltage Variations.
  - This variation is due to IR Drop. Generally, it is targeted around +/- 10%.
  - Drain current is a function of Supply Voltage.
  - Better driving strength with the higher Supply Voltage. And, faster is the Logic cell.

- **Interconnect Extraction Variation:**

  - Two parameters :
    - Resistance – Can be maximum, can be minimum
    - Capacitance – Can be maximum, can be minimum

When,

C is minimum - CMIN

C is maximum - CMAX

R is minimum, RC(delay) is minimum - XTALK

R is maximum, RC(delay) is maximum – DLY

Interconnect Dimensions have more control on Resistance than on Capacitance.

Fixing the Timing Problems

A typical checklist for fixing timing problems can be as follows:

- Ensure that the path facing violation is valid.
- If valid, then try and identify the critical time consuming segments from the verbose report.
- Check for high transition times and/or high loads.
- For setup violations on input/output paths, check constraints, consult STA team.
- For hold violations on input/output paths, not critical, can wait for top level STA.

Checklist For STA Sign-Off

- Check_timing report.
- Parasitics annotations.
- All timing exceptions validated from designers.
- Setup and hold timing violations reports; including clock gating on functional gates.
- Recovery and removal violations.
- Transition violations.
- Minimum pulse width violations.
- Electrical specs for external interfaces.

Thanks!