Efficient Clock Monitoring System for SoC Clock Verification

Nam Pham Van

NXP Semiconductor

Outline

- **Introduction**
- **Clock Monitor Module**
- **Test Bench Integration**
- **Debugging and Pass / Fail Criteria**
- **Summary**

Outline

- **Introduction**
- Clock Monitor Module
- Test Bench Integration
- Debugging and Pass / Fail Criteria
- Summary

Introduction

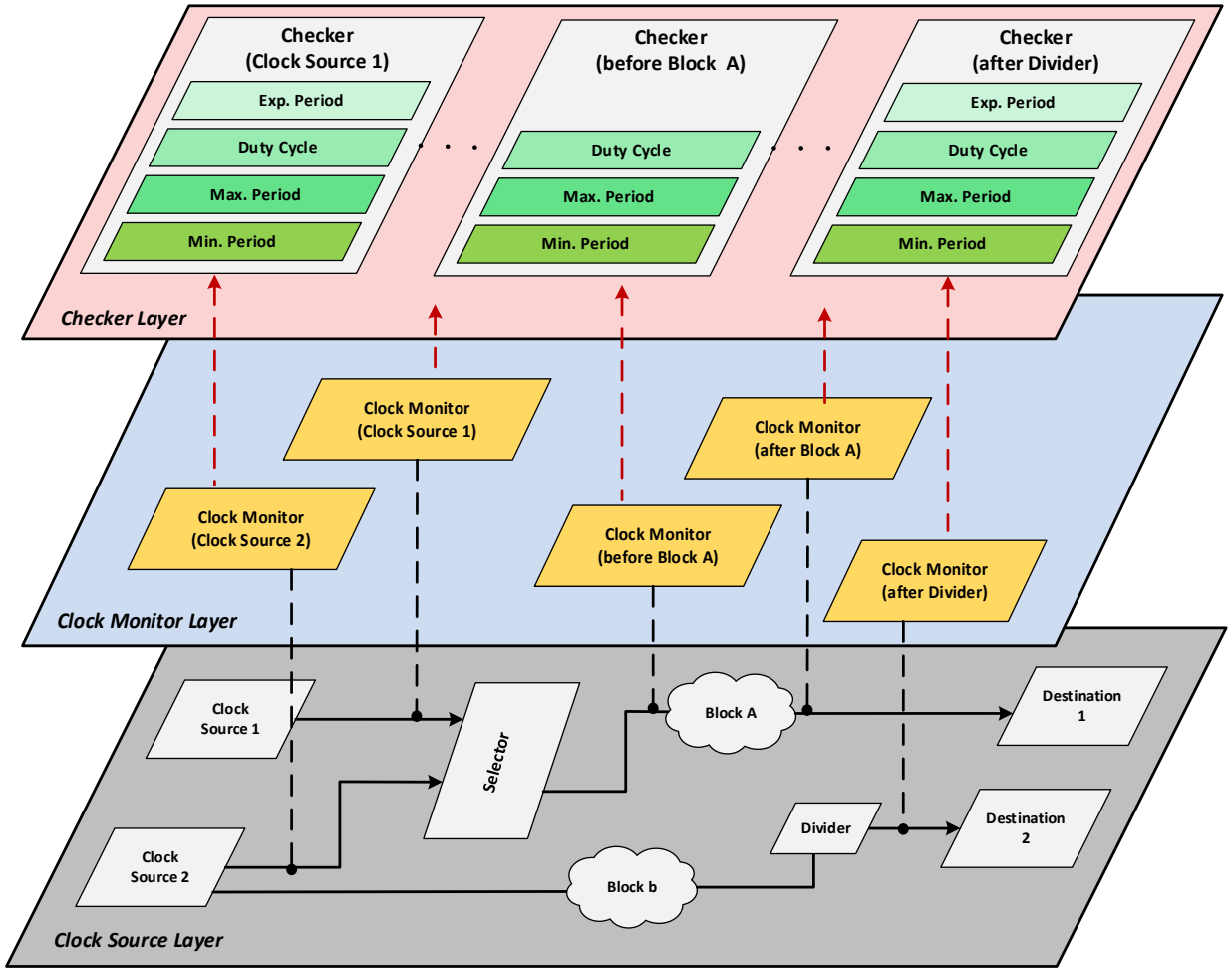
Situation:

- Mixed of Clock Monitors & Checks
- Functionalities are difficult to understand
- Difficult to use and to debug
- Reusability questionable

New approach:

- Separation of Clock Monitors & Checks
- Simplify the debugging
- Modular modules (reusability)
- Easy to expand

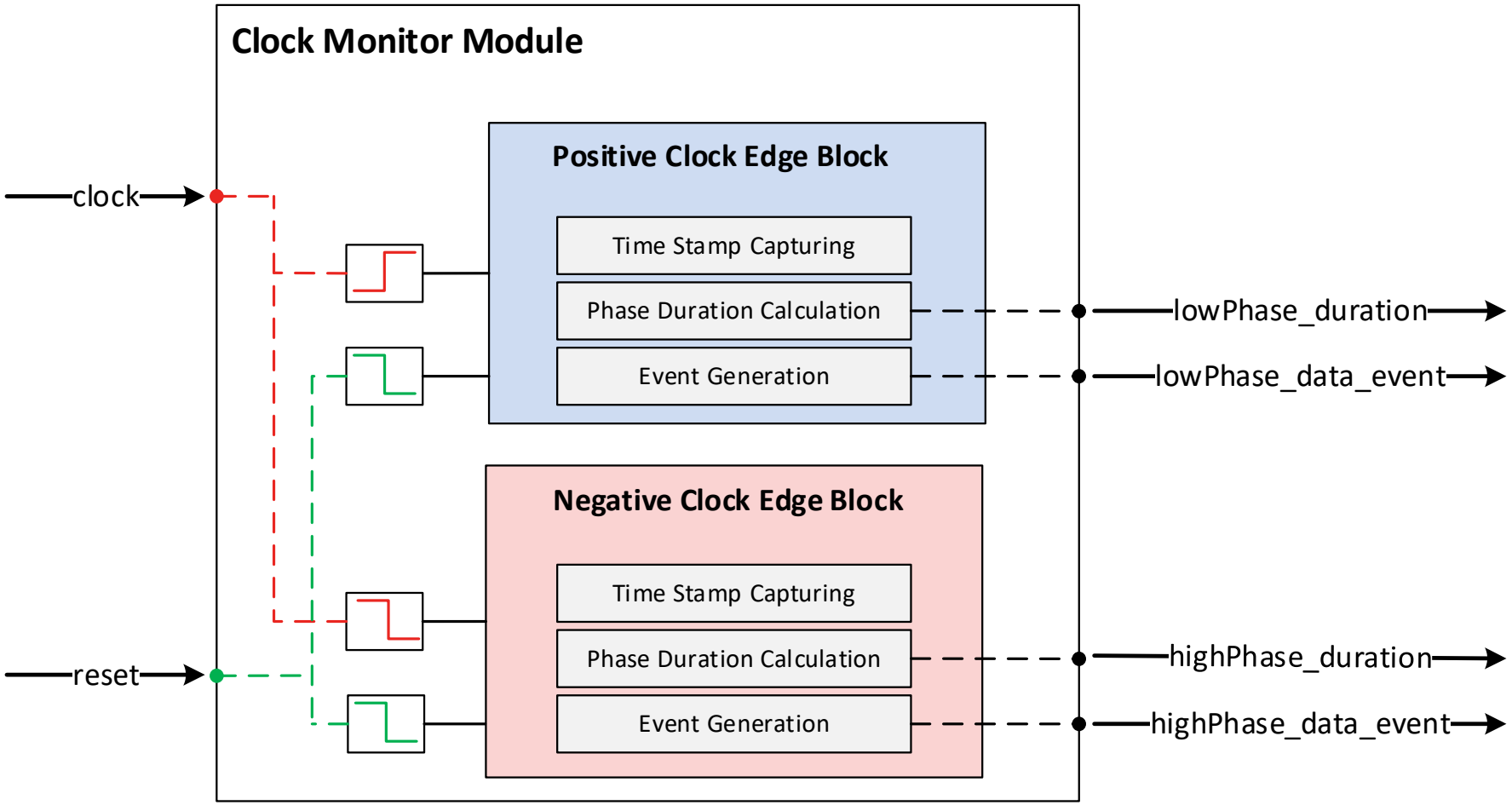
Introduction – New Concept



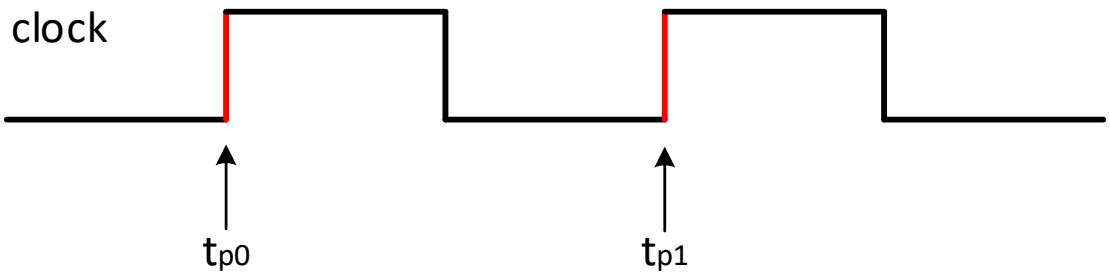
Outline

- Introduction
- **Clock Monitor Module**
- Test Bench Integration
- Debugging and Pass / Fail Criteria
- Summary

Clock Monitor Module – Architecture



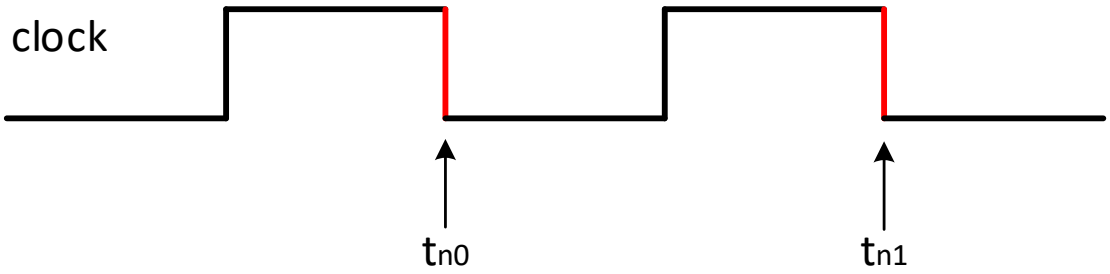
Clock Monitor Module – Time Stamp Capturing



Positive clock edge time stamp:

t_{p0} : last time stamp

t_{p1} : current time stamp

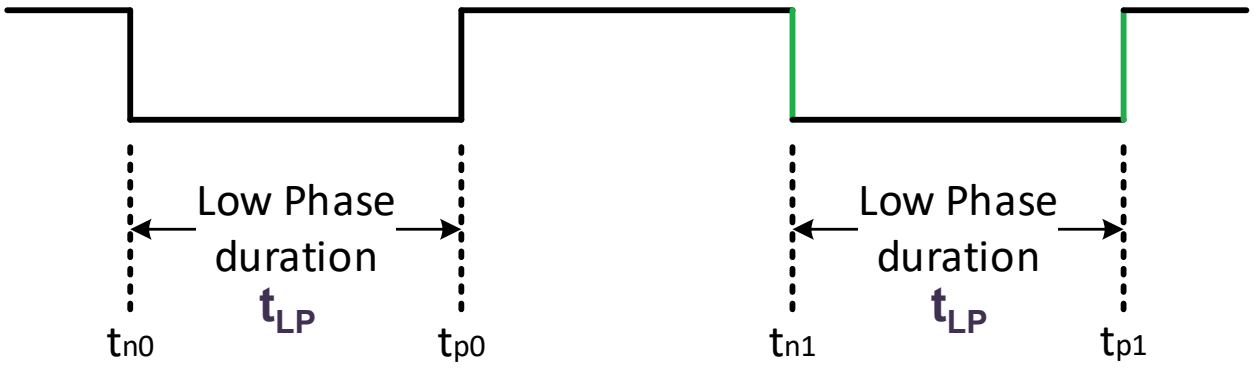


Negative clock edge time stamp:

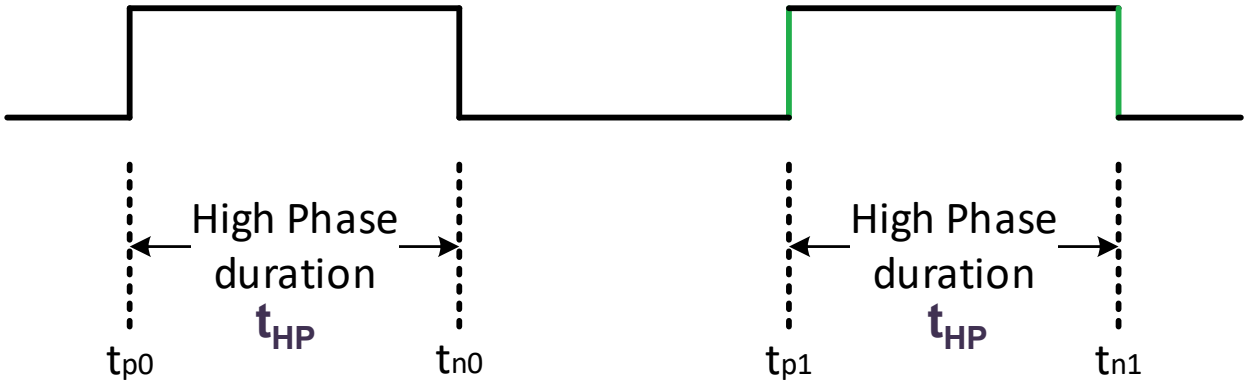
t_{n0} : last time stamp

t_{n1} : current time stamp

Clock Monitor Module – Phase Duration Calculation



$$t_{LP} = t_{p1} - t_{n1}$$

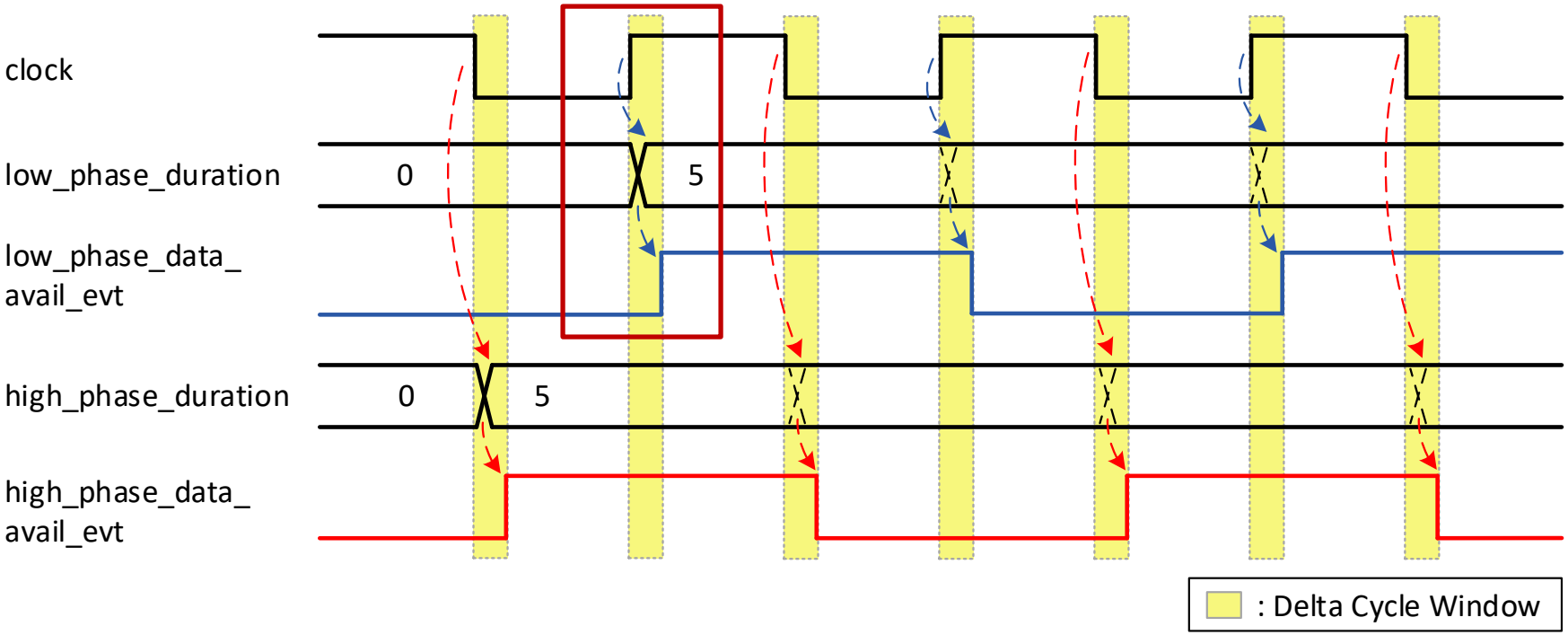


$$t_{HP} = t_{n1} - t_{p1}$$

t_{p0} : last pos. clock edge time stamp, t_{p1} : current pos. clock edge time stamp

t_{n0} : last neg. clock edge time stamp, t_{n1} : current neg. clock edge time stamp

Clock Monitor Module – Event Generation



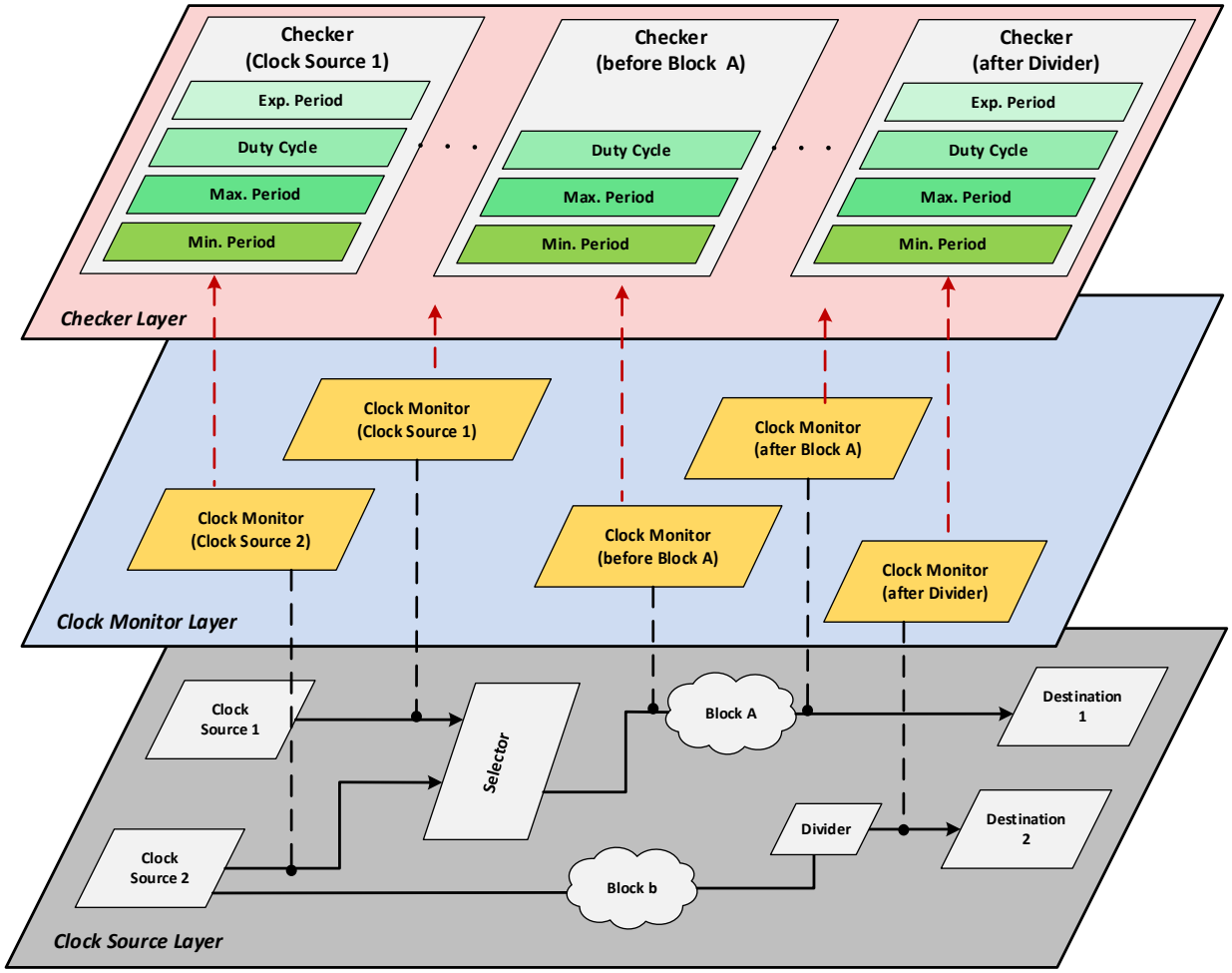
Order:

1. Time Stamp Capturing
2. Phase Duration Calculation
3. Event Generation

Outline

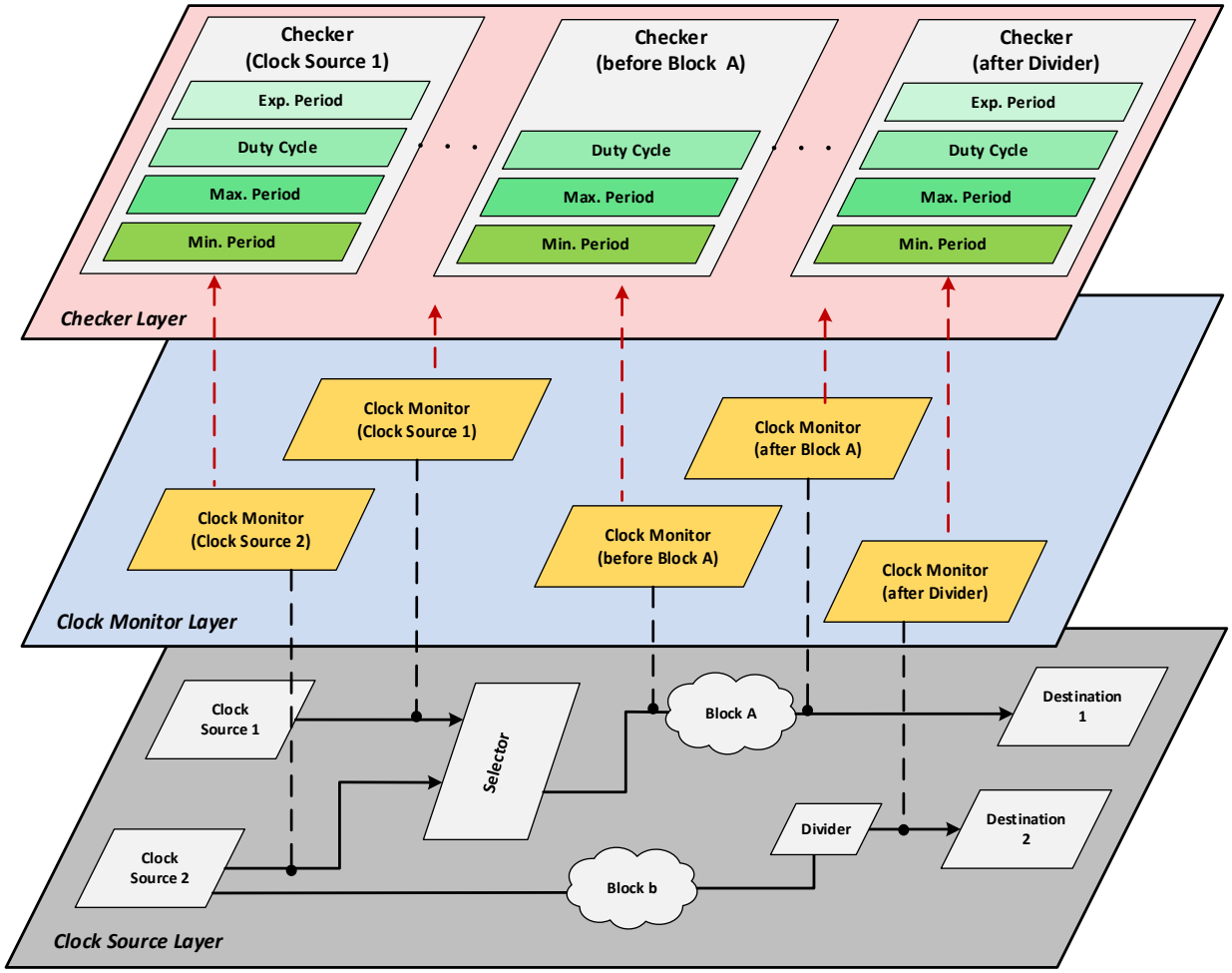
- Introduction
- Clock Monitor Module
- **Test Bench Integration**
- Debugging and Pass / Fail Criteria
- Summary

Test Bench Integration

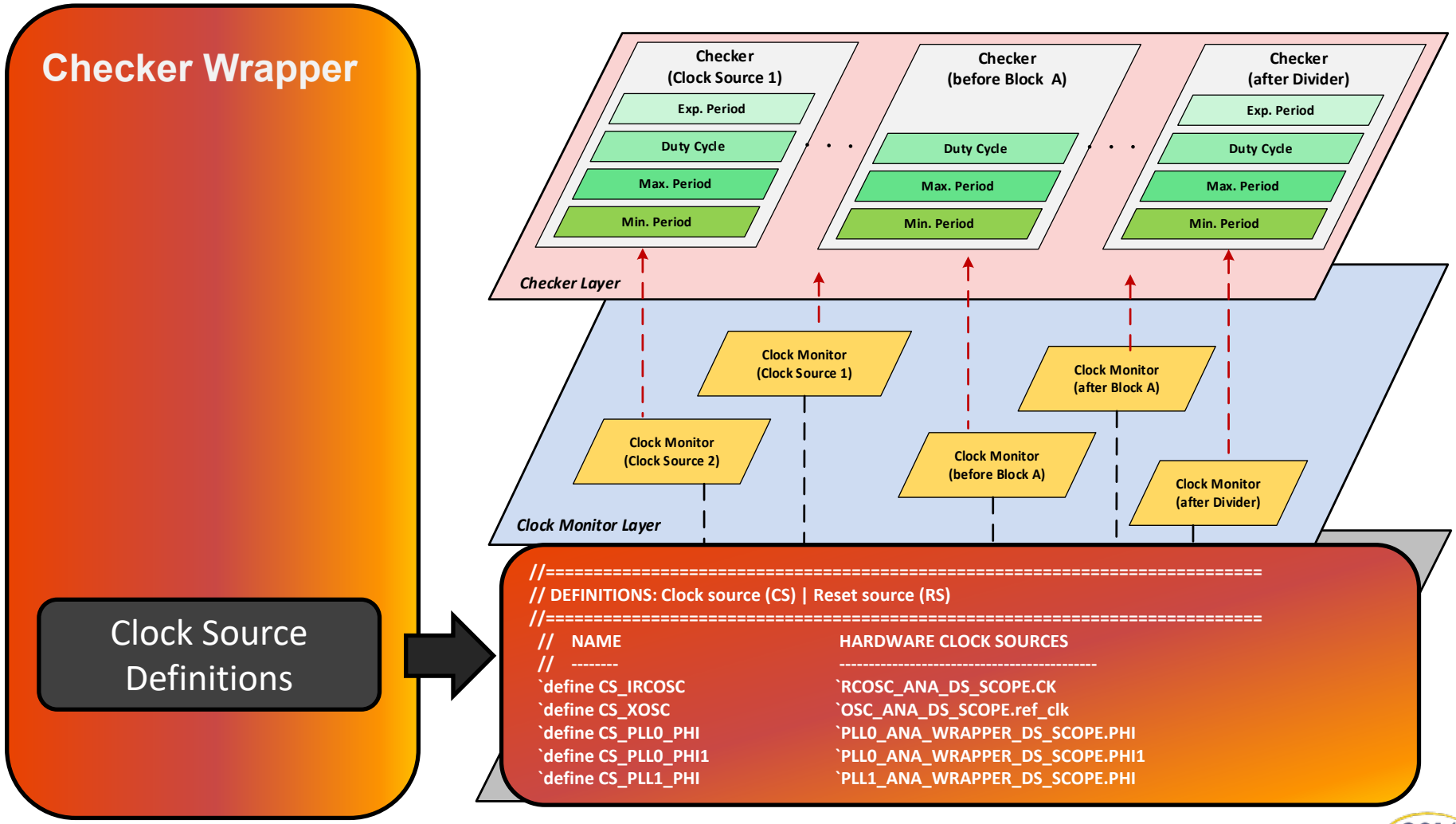


Test Bench Integration

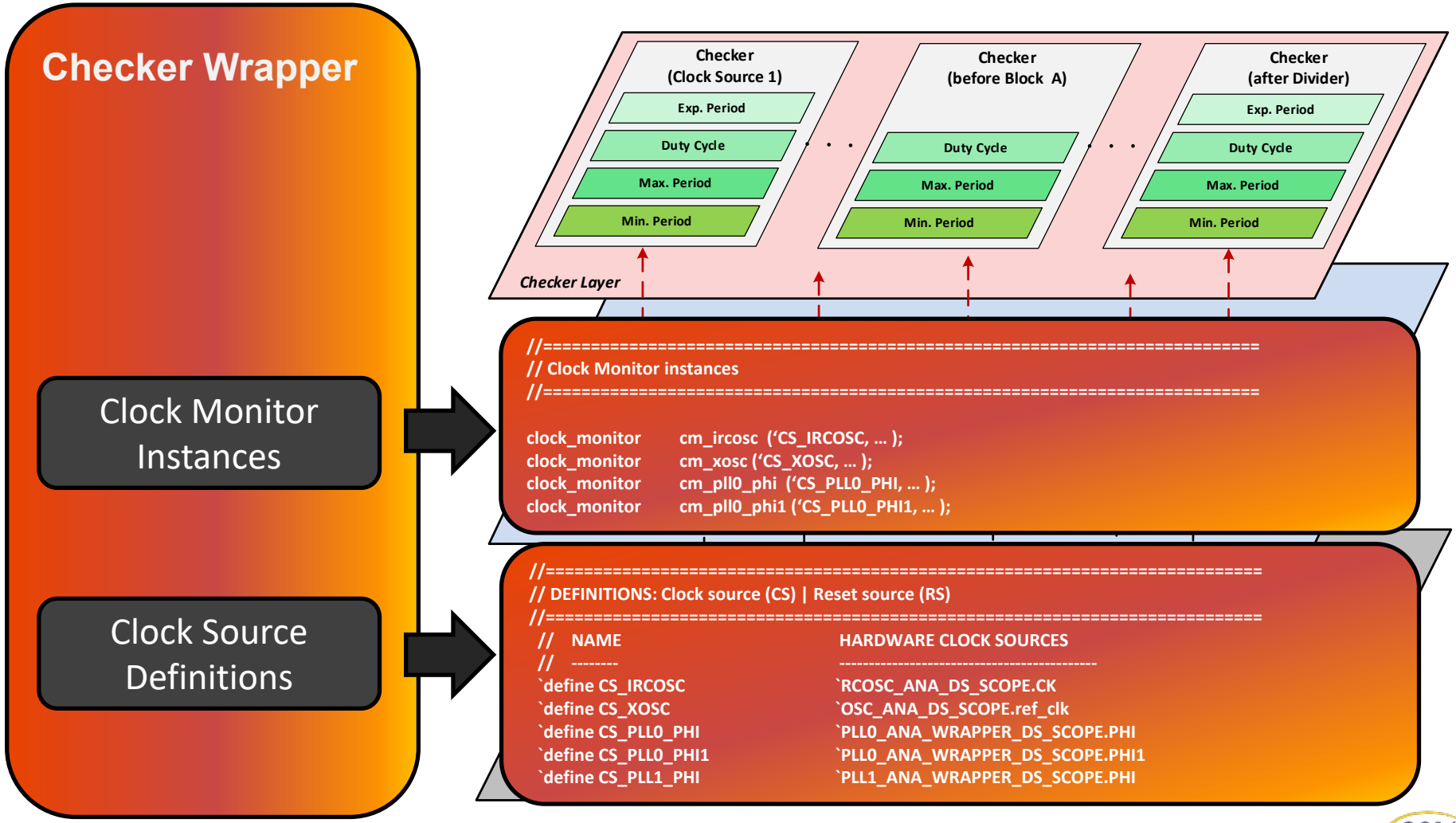
Checker Wrapper



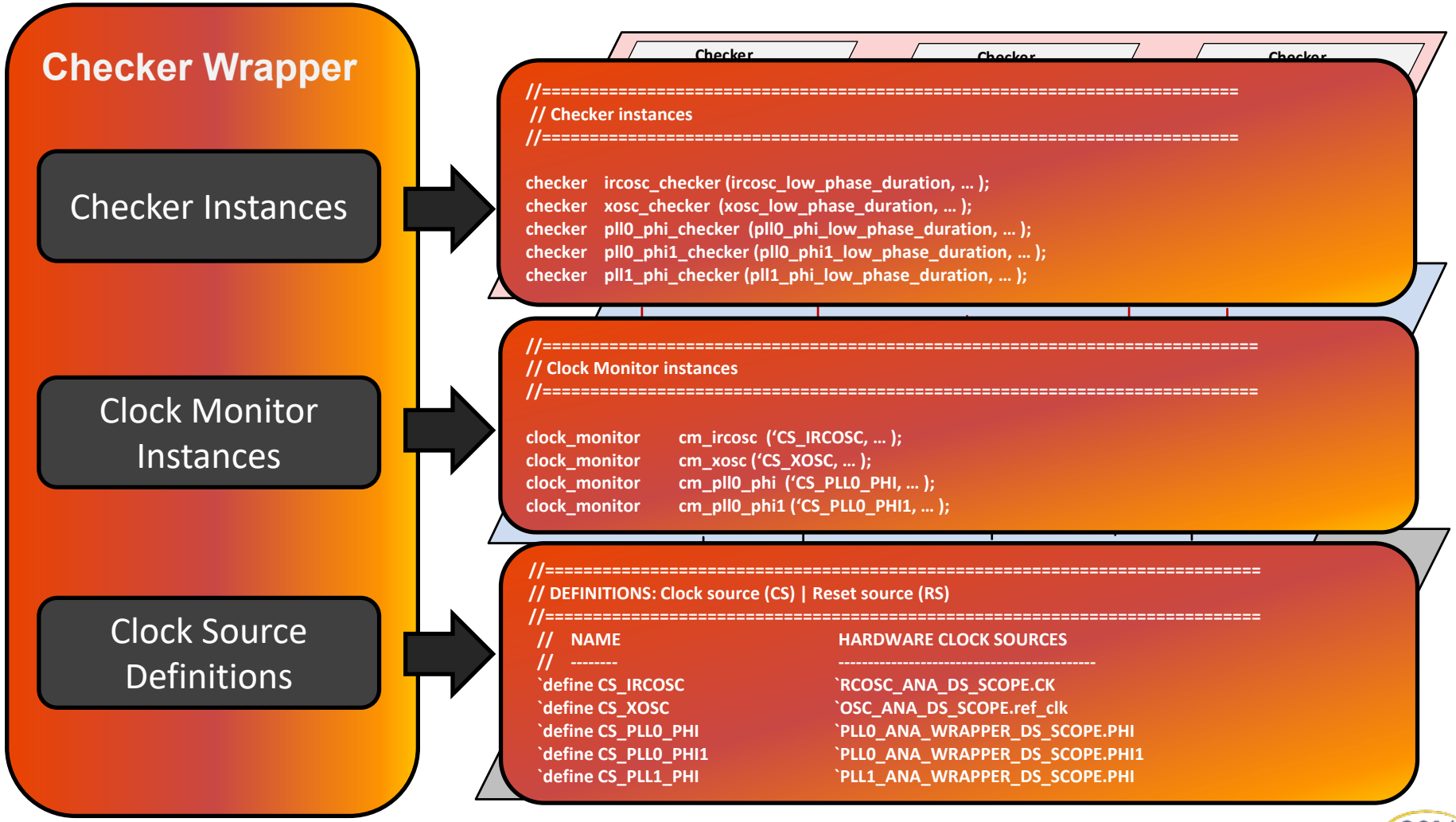
Test Bench Integration



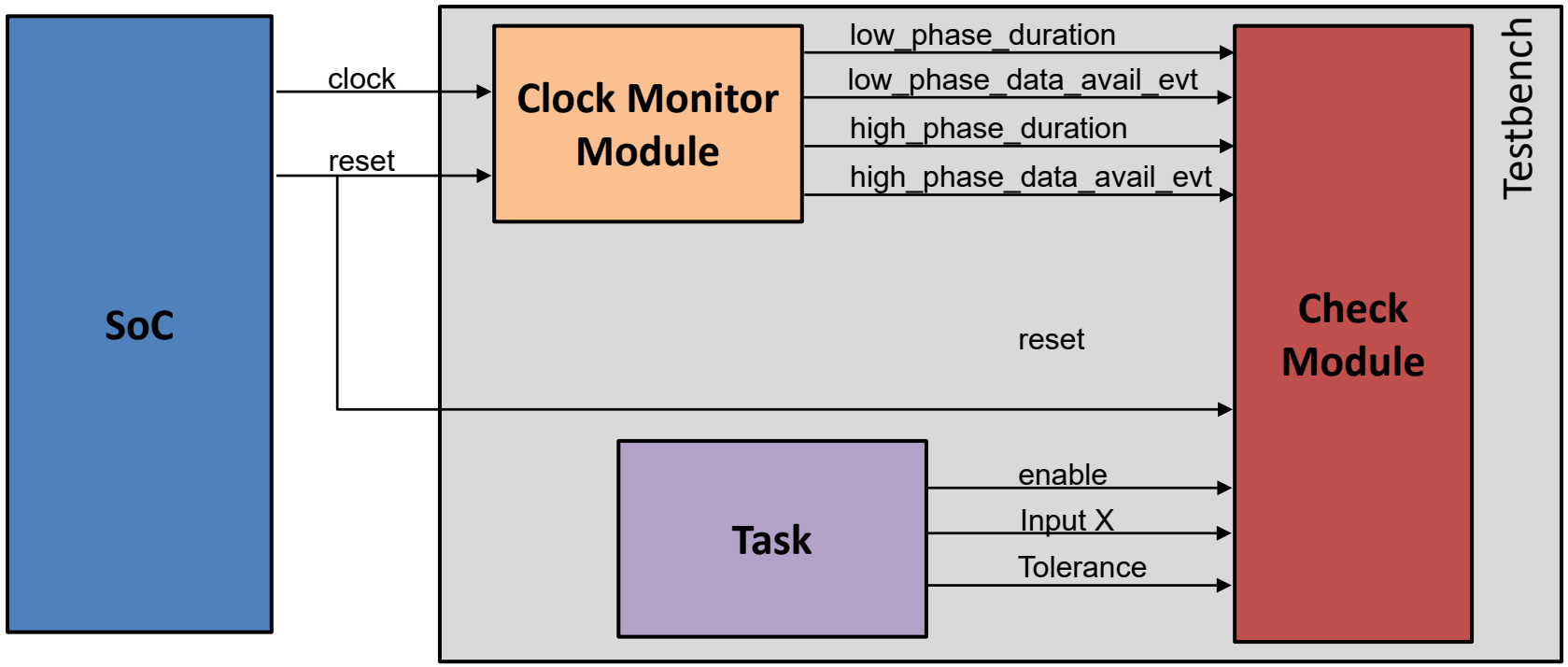
Test Bench Integration



Test Bench Integration



Test Bench Integration



Check Module	Input X	Assertion Condition
Exp. Period	Exp. Period	Min. Exp. Period < Clock Period < Max. Exp. Period
Max. Period	Max. Period	Clock Period < (Max. Period + (Max. Period * Tolerance))
Min. Period	Min. Period	Clock Period > (Min. Period - (Min. Period * Tolerance))
Duty Cycle	Exp. Duty Cycle	Min. Exp. Duty Cycle < Duty Cycle < Max. Exp. Duty Cycle

Test Bench Integration – Tasks

Examples:

testbench.checker_wrapper.<clock source checker>.<check>(.enable(),.min_period(),.tolerance());

`CHECKER...SCOPE.pll1_phi_checker.max_period(.enable(1),.max_period(6.000ns),.tolerance(0.001));

→ time in nanosecond

`CHECKER...SCOPE.pll1_phi_checker.exp_period(.enable(0),.exp_period(8.333us),.tolerance(0.001));

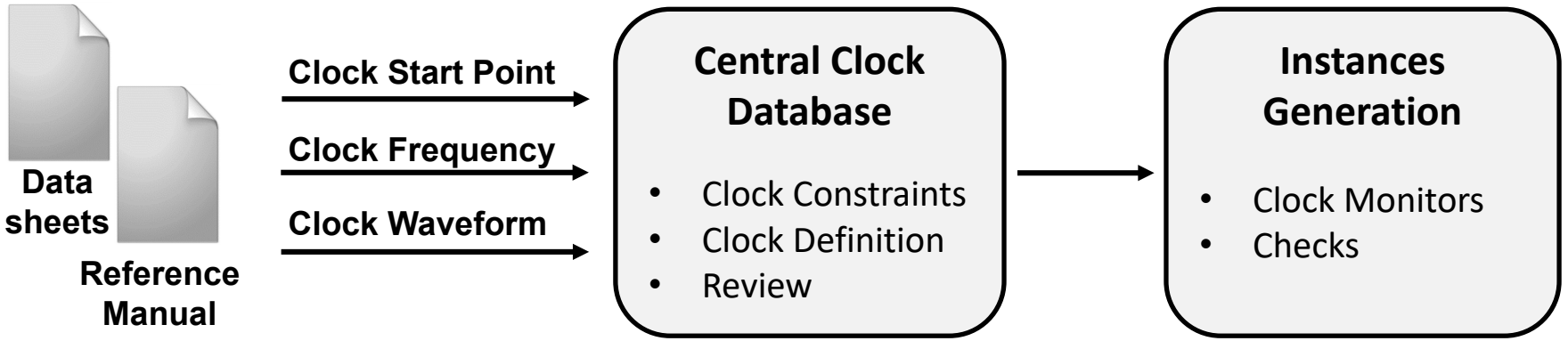
→ time in microsecond

`CHECKER...SCOPE.pll1_phi_checker.min_period(.enable(1),.min_period(25.000),.tolerance(0.001));

→ time depends on the default configuration defined in the test bench!

`CHECKER...SCOPE.xosc_checker.duty_cycle(.enable(1),.exp_duty_cycle(0.5),.tolerance(0.001));

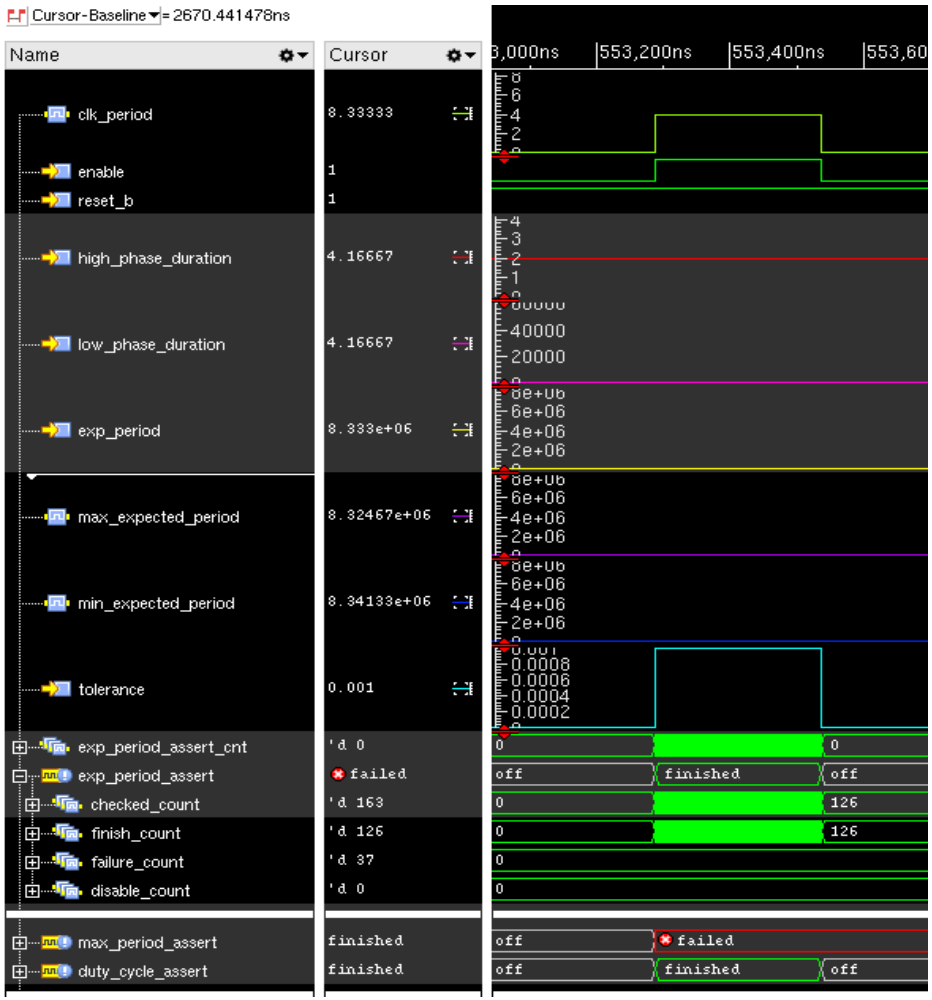
Test Bench Integration



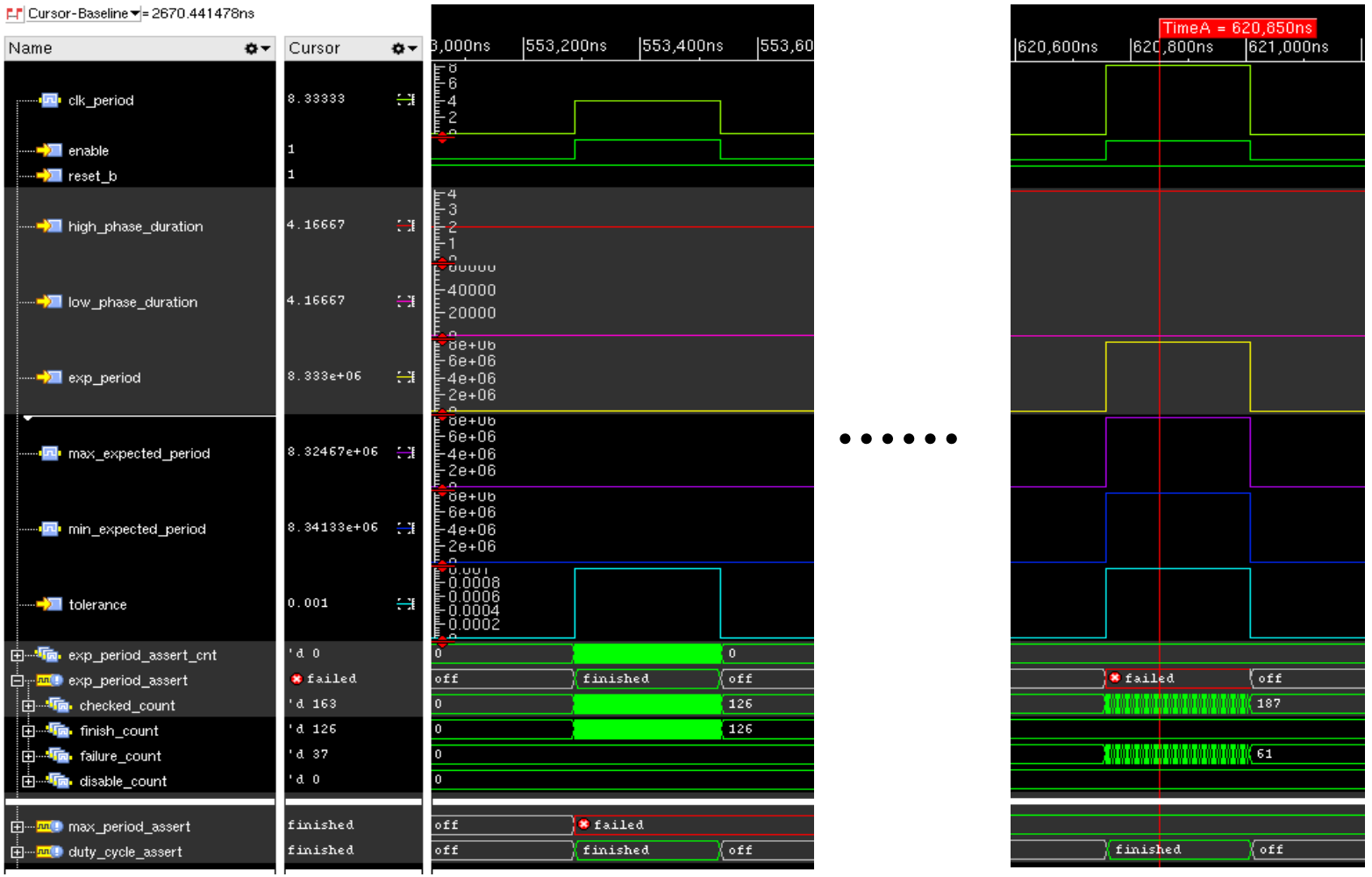
Outline

- Introduction
- Clock Monitor Module
- Test Bench Integration
- **Debugging and Pass / Fail Criteria**
- Summary

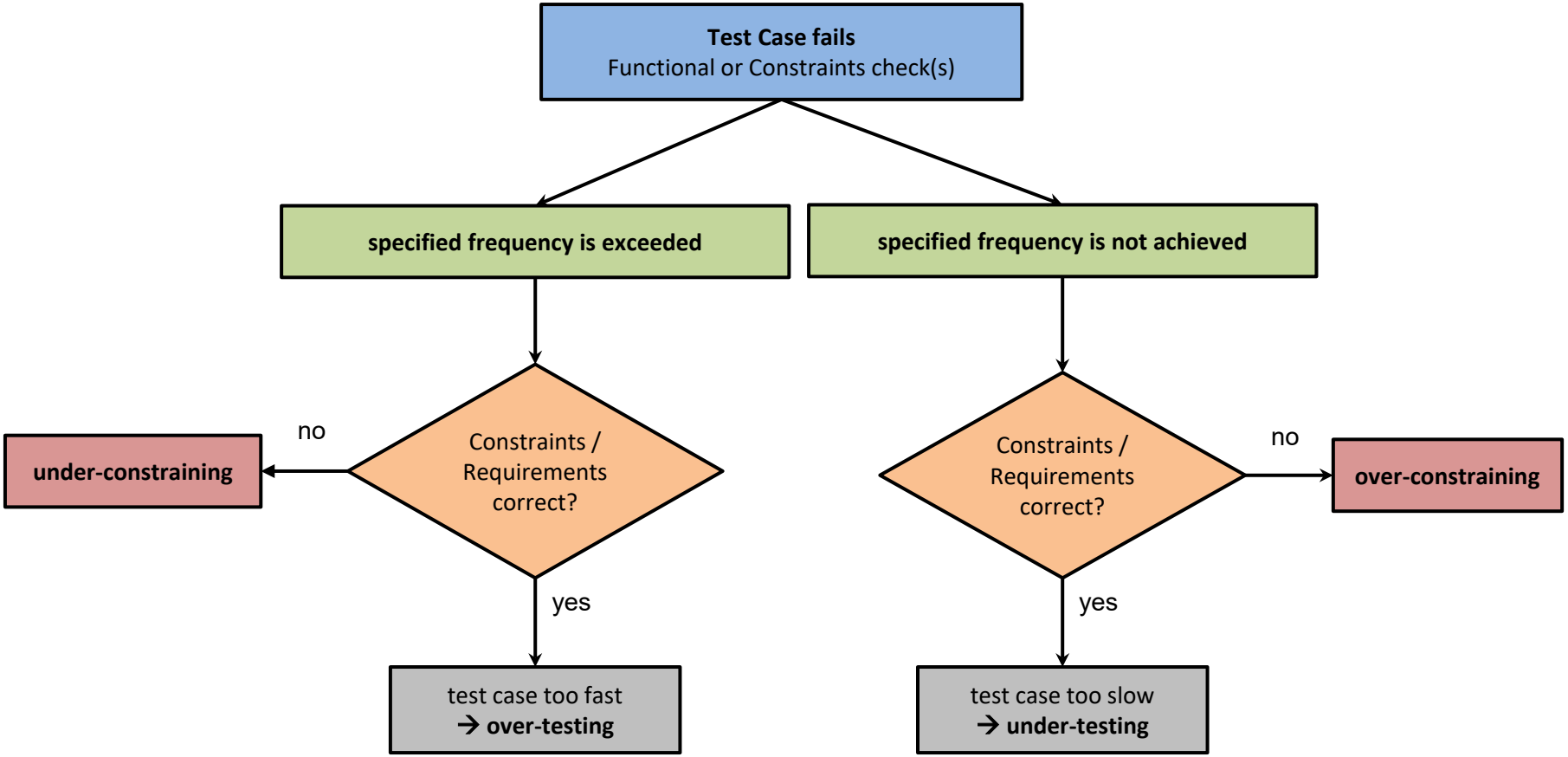
Debugging – Waveform (e. g. Cadence)



Debugging – Waveform (e. g. Cadence)



Debugging - Pass / Fail criteria



Outline

- Introduction
- Clock Monitor Module
- Test Bench Integration
- Debugging and Pass / Fail Criteria
- **Summary**

Summary

New Clock Monitor & Checker:

- Easy to use
- Reusability is given
- Checks (as module) are easy to integrate
- Simple to debug
- Implementation can be extended easily with new clock monitors and checker

Special Thanks

- Thomas Lüdeke, NXP Semiconductor
- Bernhard Braun, NXP Semiconductor
- Clemens Roettgermann, NXP Semiconductor
- Dirk Moeller, NXP Semiconductor

Thank you for your attention!

Questions