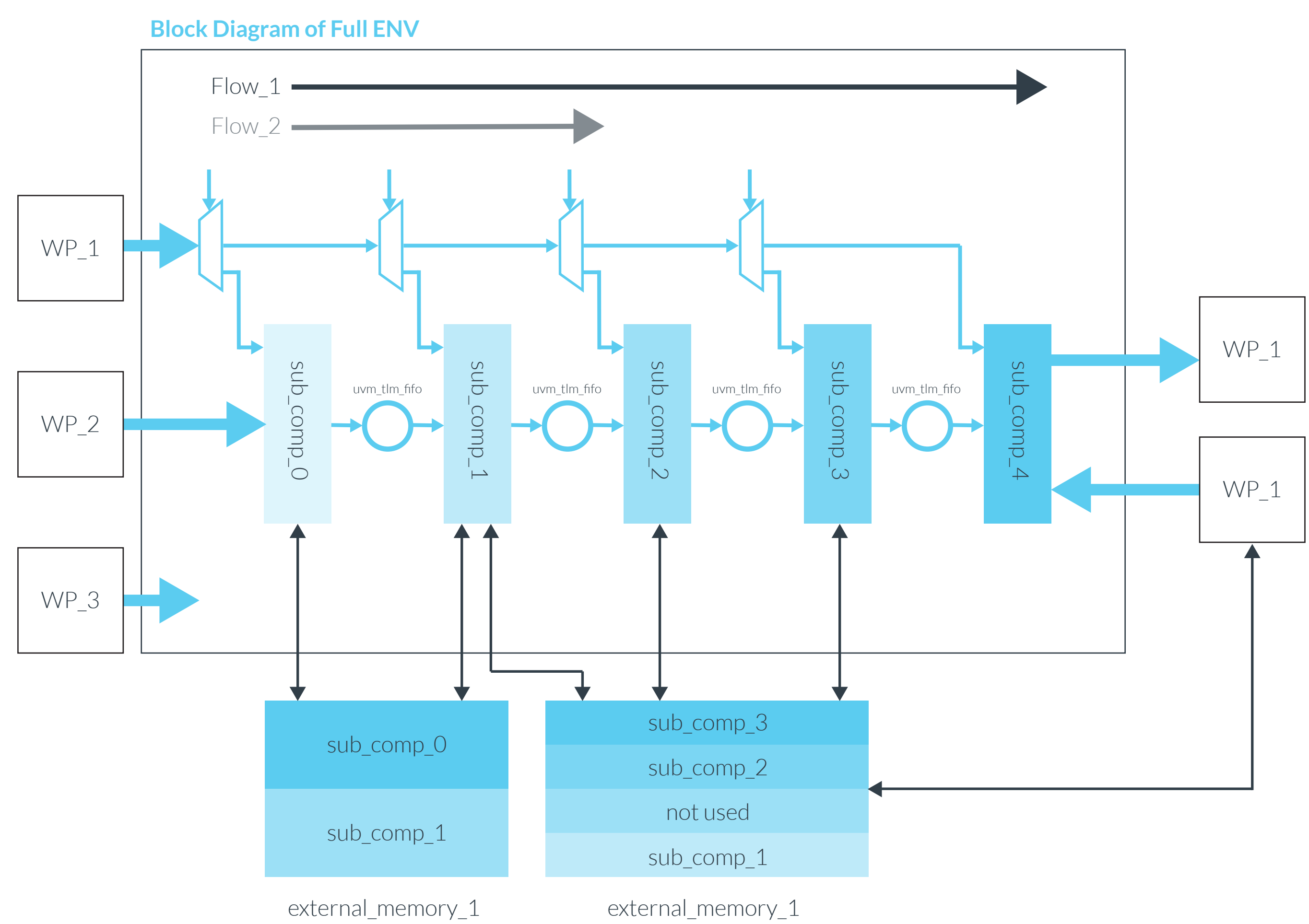
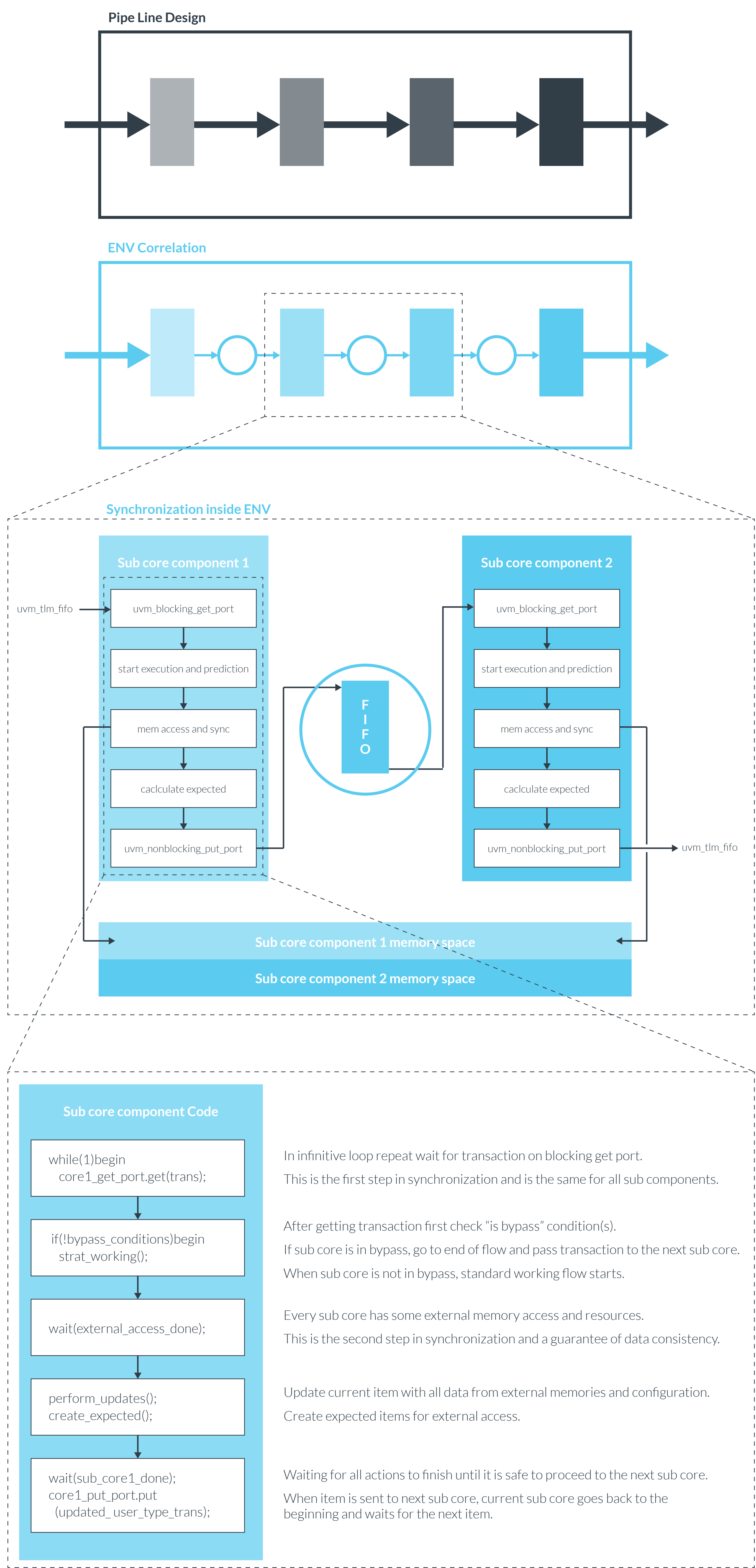


Verification Strategy for Pipeline Type of Design



Introduction

- Complexity of ASICs requires adjusting the methodologies.
- Modularity of the design should be followed by the verification methodology.
- Pipeline** type of design should be verified with **pipeline** testbench.
- Efficiency in verification process is a must!

Pipeline type of design

- Pipeline** type of design consist of several independent sub cores.
- Sub cores have different configurations for execution flow.
- Synchronization and arbitration between sub cores is a challenge.
- The main goal is to achieve good performance even with back pressure.
- Shared resources create race conditions.

Design and ENV Correlation

- Each sub core should be covered with one uvm_component.
- All sub_core_components should be connected with a tlm_fifo_port.
- Synchronization should be done with blocking/non_blocking put/get port.
- Synchronization in shared resources should be covered with semaphores.
- ENV/DUT synchronization is done only on transaction level.

Verification Scopes

- Creating a uvm_component for each sub core of DUT create reference model.
- Stimulus are unique and random for the entire system.
- Parallelism and back pressure is distributed between components.
- Bypass of each component is performed on the sequence level.
- Coverage is collected on component level in a real use case.

Reusability and maintainability

- Great reusability in both directions:
 - Reference model of each component can be used as standalone.
 - Entire pipeline can be used in top level verification environment.
- Any change request affects only one sub component.
- Any component can be easily integrated and adjusted for usage in other project.

Advantages and improvements

- Entire scope of work is divided to logical parts.
- More engineers can work without dependencies.
- Good synchronization between RTI and verification environment.
- Improvements could be done in generalization of this approach on each complex DUT.

Summary

- Biggest challenges with complex DUT is to choose a suitable verification strategy.
- Great reusability and maintainability of the verification environment.
- Entire scope of work is split logically and with no dependencies.
- Synchronization between DUT/TB is handled by the strategy itself.
- Shared resources and arbitration is covered and handled by the methodology.