Utilizing SystemC/TLM for adapting block level verification environment for reuse at System level

Wasiq Zia

Cadence Design Systems

accellera SYSTEMS INITIATIVE™

2014 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Topics

- Introduction
  - IP and System level verification
  - Testbench reuse
- Methodology
  - Register interface
  - Test Writing
  - Testbench Models
- Advantages
- Conclusion
- Questions

Introduction

- Enable adaption of the IP level testbench and environment at the system level verification
- Utilize SystemC/TLM interfaces for data exchange
- Enable configuration control through the register based interface
- Complete portability of IP level testcases to system level environments

Comparison

IP Level Verification

- Smaller environment pertaining to a single block only
- Random and extensively controlled verification scenarios
- Testing targeted more at functionality instead of connectivity
- No integration overheads

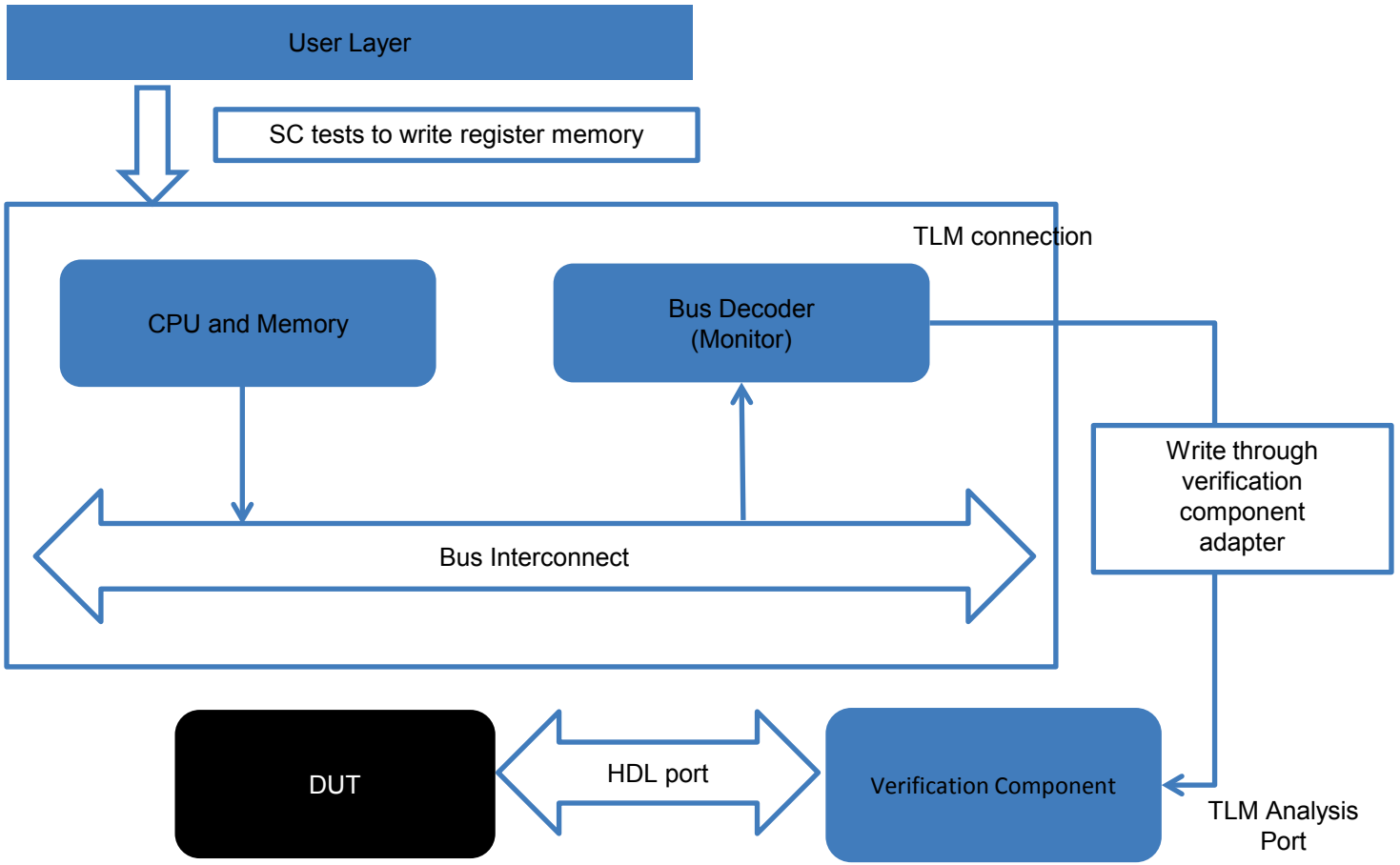
System Level Verification

- Large verification environment with several components
- Mostly directed scenarios with low level of control and test randomization
- Interface connection testing is an important goal
- Huge integration overheads due to multiple blocks having interconnectivity

Need for Testbench Reuse

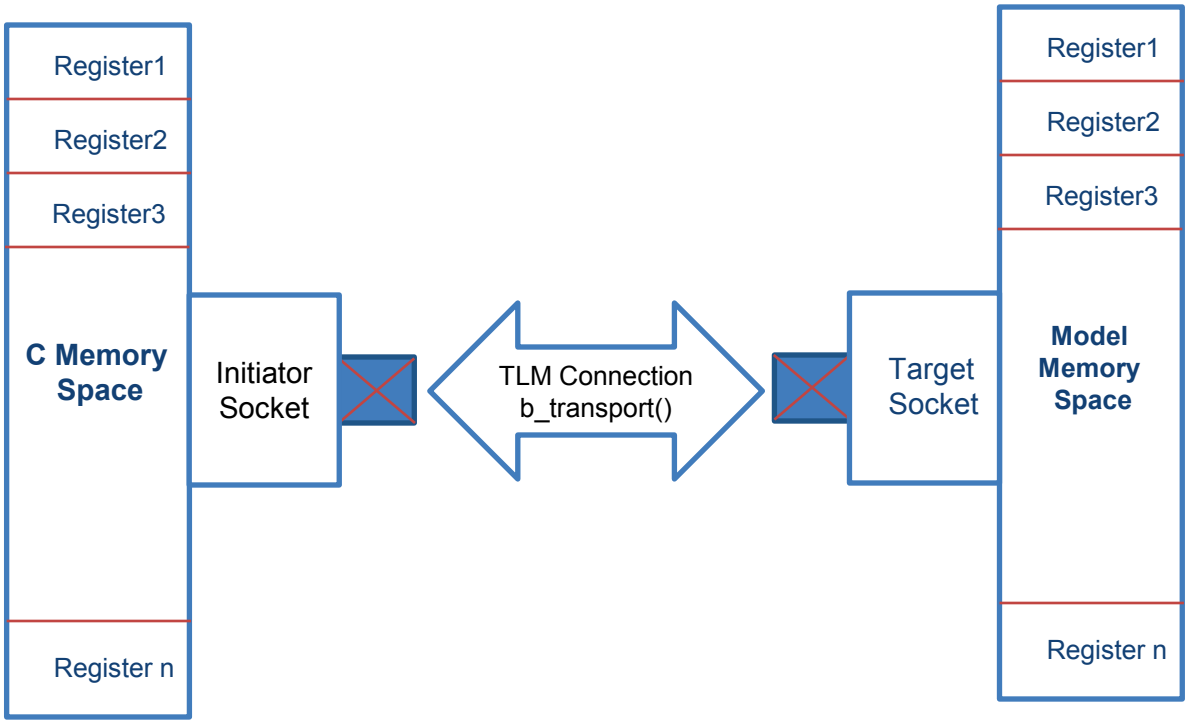
- Integration tasks take up a lot of verification engineers time at system level
- Creating new set of test vectors and validating them again requires knowledge of each block
- The integrator should also have knowledge of each verification technology used as separate verification components could come from different sources
- No synergy between the block level verification and system level verification

Methodology Model



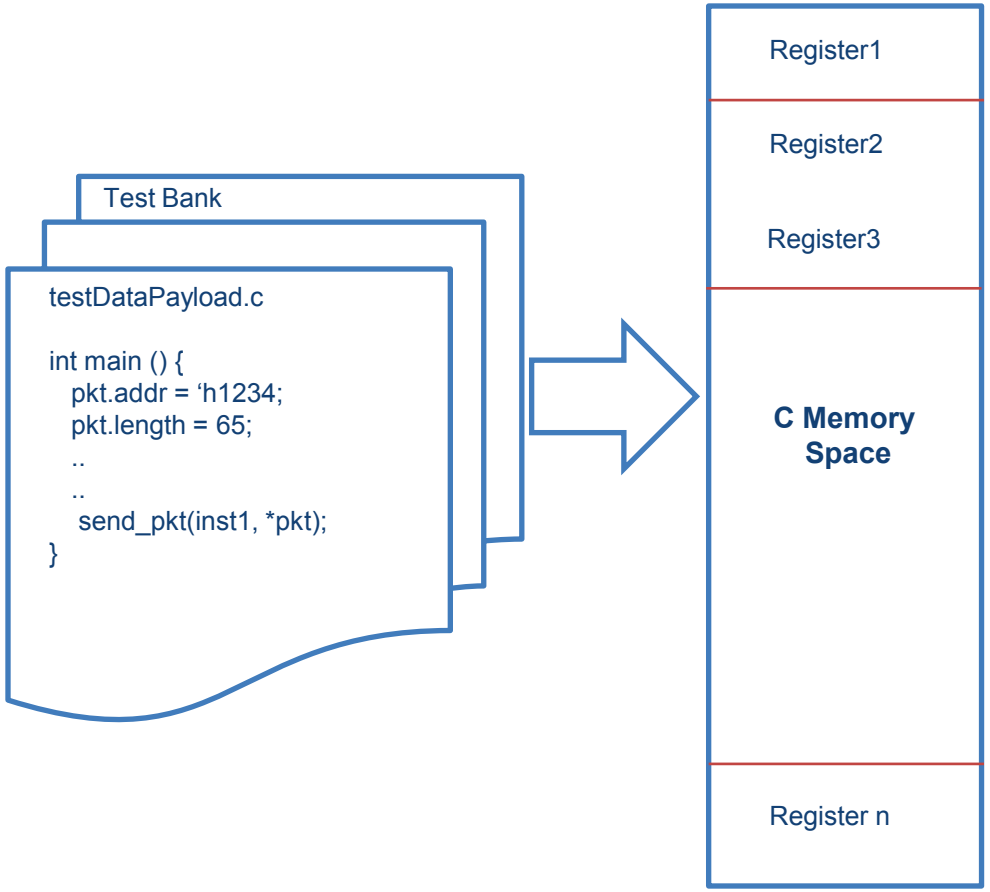
Register Interface and TLM Sockets

- Provide register interface for mapping the information to be transferred
- Use address offsets to support different instances
- Utilize TLM Sockets for communication
- Generic payload used for interoperability
- Use of blocking transport mechanism for ensuring that simulation timings are honoured
- OVM_ML bindings for connecting to analysis ports defined for HVL based interfaces

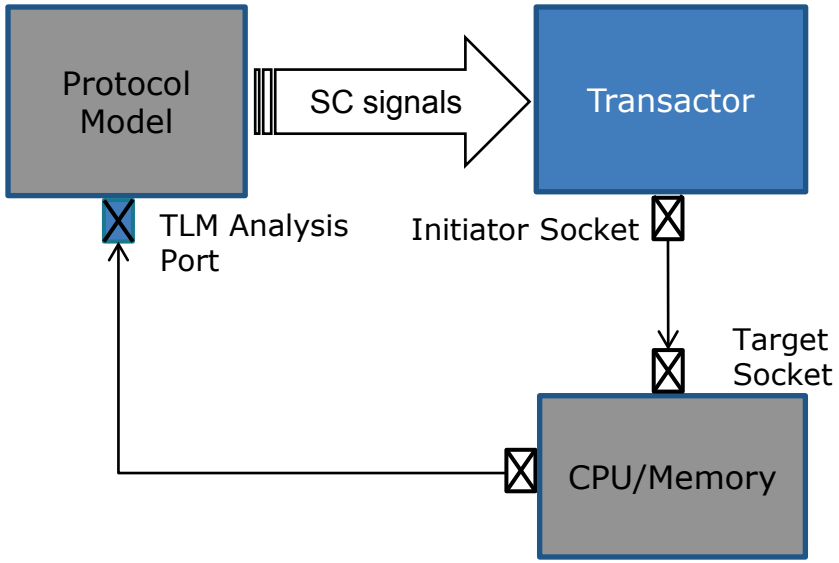


Test Writing Interface

- Testcases written in a high level language like C
- Test vectors are fixed and provided through the C API's to the register interface
- High level API's are easy to use with a consistent interface that reduces the learning for each model environment
- Provides a common mechanism to control the test infrastructure, low level software and the verification component

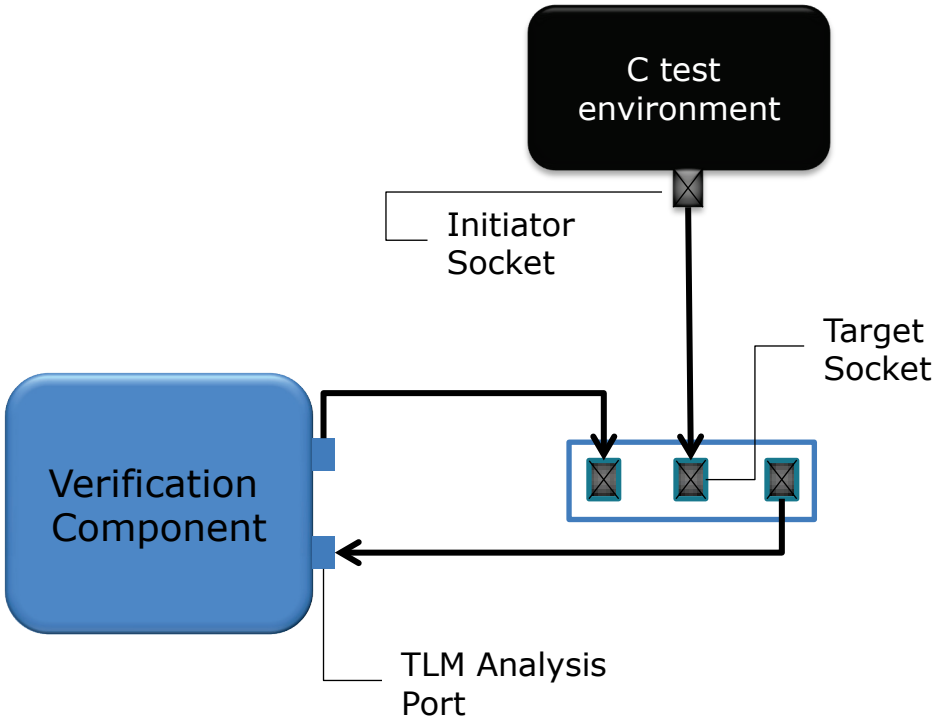


Use Models



SystemC port interface connection

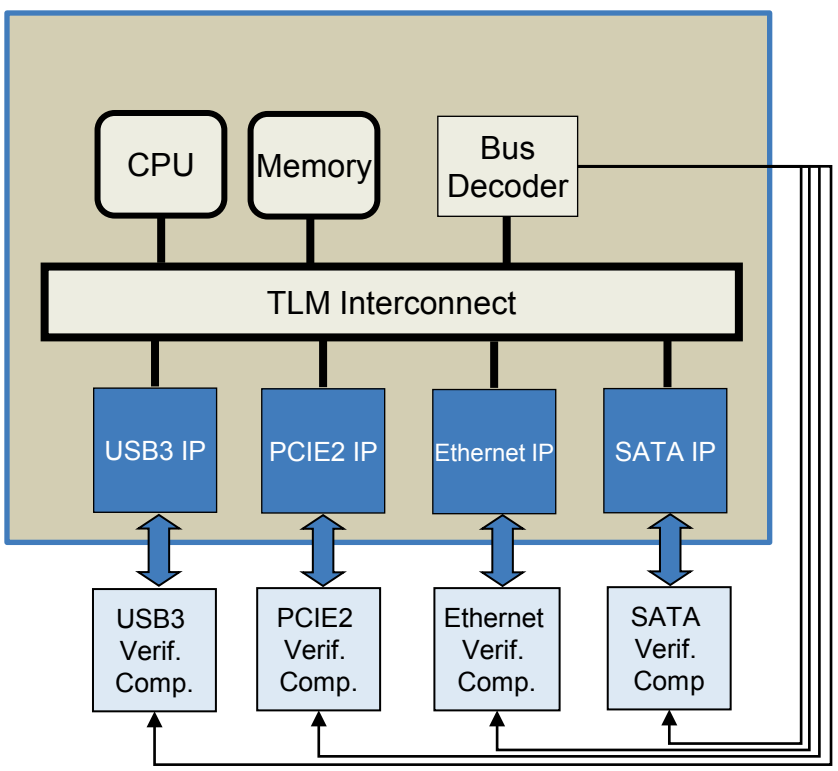
RTL connection using TLM sockets for transaction driving

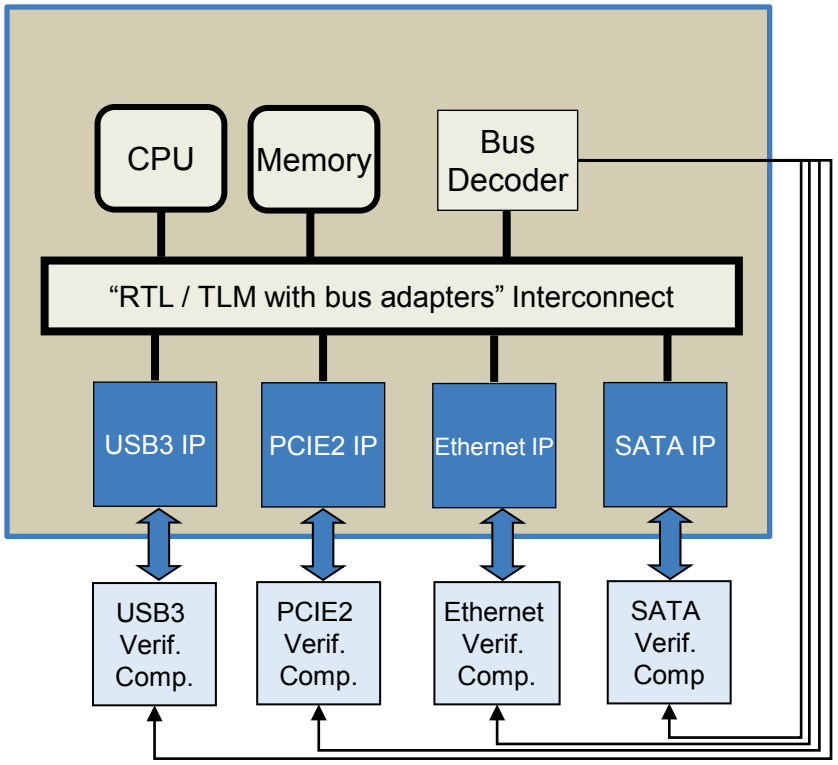


Testbench Model

The testbench connections can be done in two possible ways:

- A TLM based connection where the model and the verification interface interact through TLM sockets
- RTL-TLM connection where the connection interface will use RTL/TLM adaptors to connect to the verification compoenent





Re-use benefits

- Existing testcases and test vector
- Ease of adaptation at different levels of hierarchy
- Re-use MDV infrstructure defined at the IP level

Control and Configuration

- Set of consistent APIs written in a high level language
- Dynamic configurability using register writes
- Protocol and technology knowledge not a blocking factor

Performance

- Significantly faster than the normal HVL level testcases
- Could be used as a layer between the software and hardware path without any performance hits

Future Work

- Addition of adaptors for the high level verification languages in order that testcases in languages like SV, e etc. can also be run using this methodology
- Callback based monitoring mechanism
- Support for acceleration through the same interface

Conclusion

- Reduces time to getting first test working at system level
- Effort for creating system level verification environment reduced significantly
- Ease of use for running the testcases and creating verification environment
- Common API's for test creation allow for easier migration from one level to next

Questions