Draft A

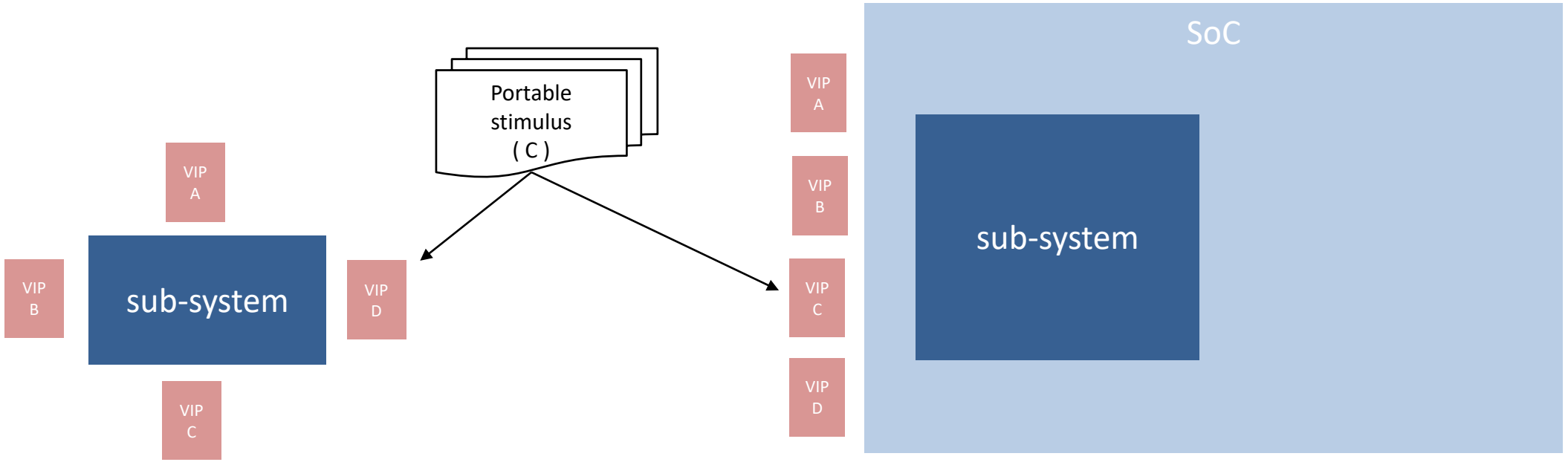
A pragmatic approach leveraging portable stimulus from subsystem to SoC level and SoC emulation

Karandeep Singh, (karandeep.singh@nxp.com) [1]
Aditya Chopra,(aditya.chopra@nxp.com) [1]
Joachim Geishauser, (joachim.geishauser@nxp.com) [2]
Nitin Verma (nitin.verma@nxp.com) [1]
NXP Semiconductors, [1] Noida, India , [2]Munich, Germany

Introduction

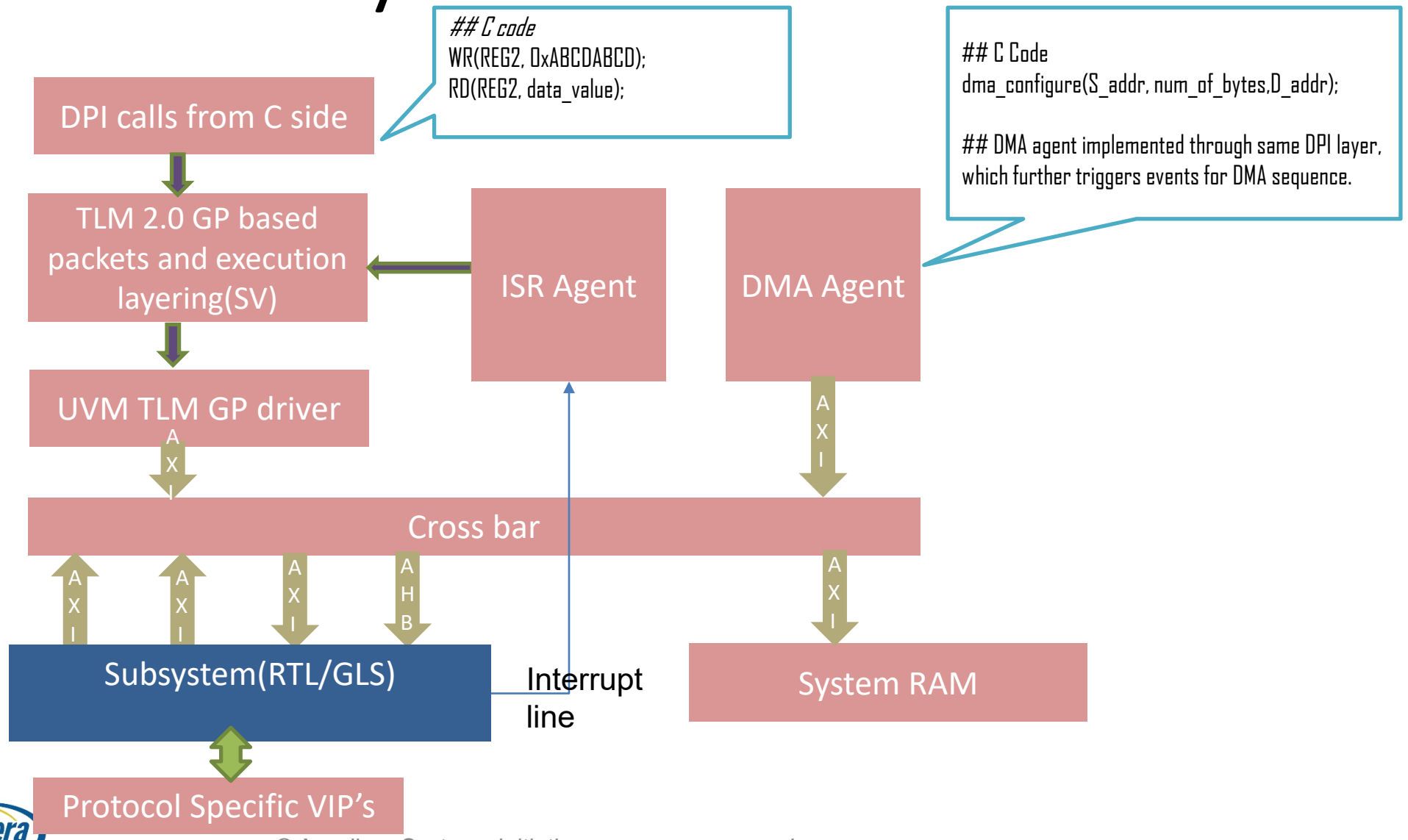
- SoC complexity
  - Verification Challenge
- Verification Stages
  - IP/Subsystem
  - SoC
- Key of testbench architecture
  - Seamless usability for native UVM Verification (sequences based approach)
  - C based test verification at subsystem
  - Stimulus portability to SoC simulation and SoC emulation

Execution at Subsystem Level

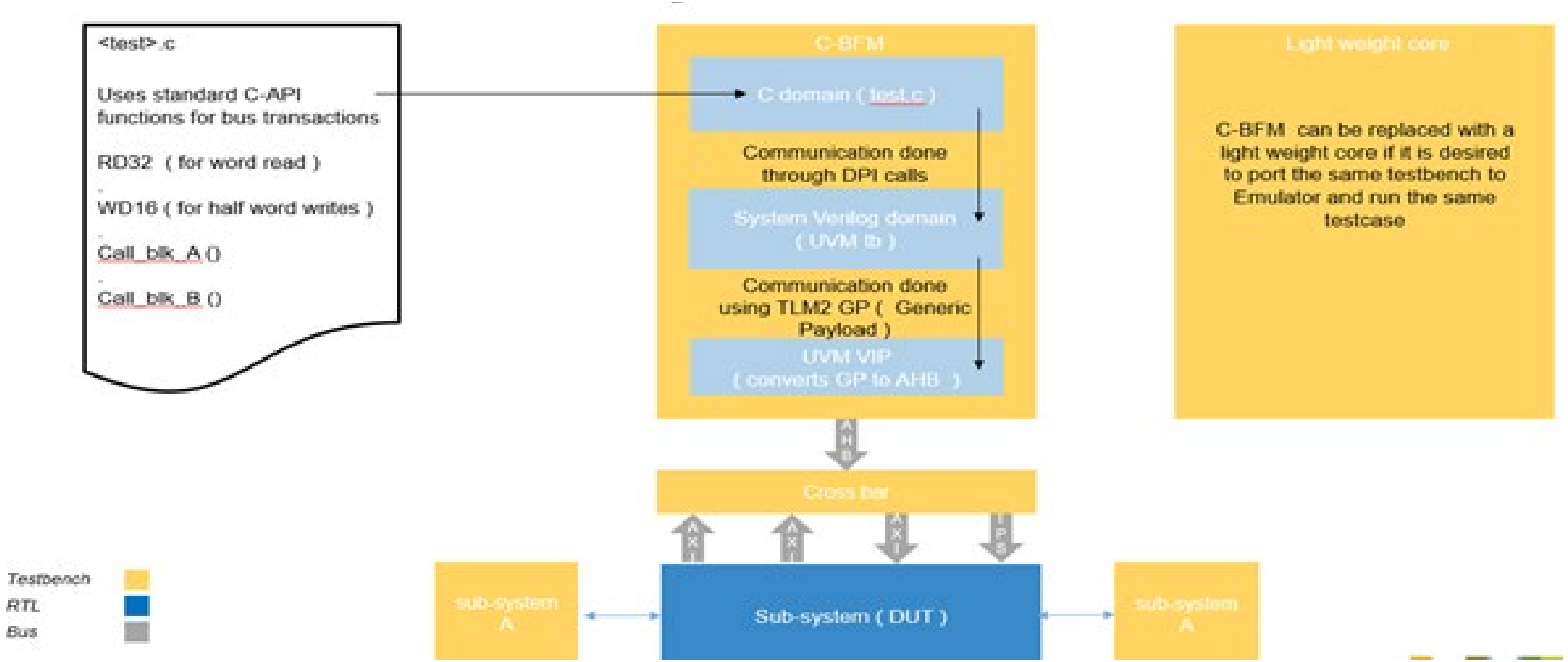


Testbench
RTL
Bus

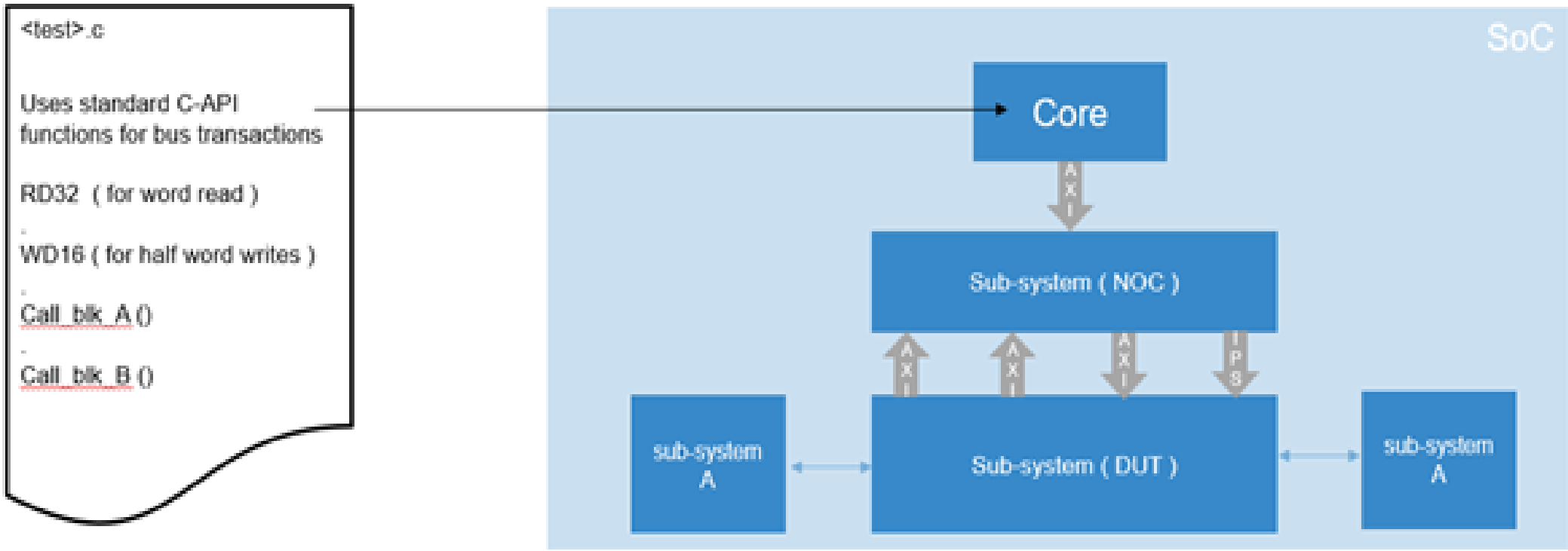
Subsystem Testbench architecture



Stimulus execution at Subsystem Level



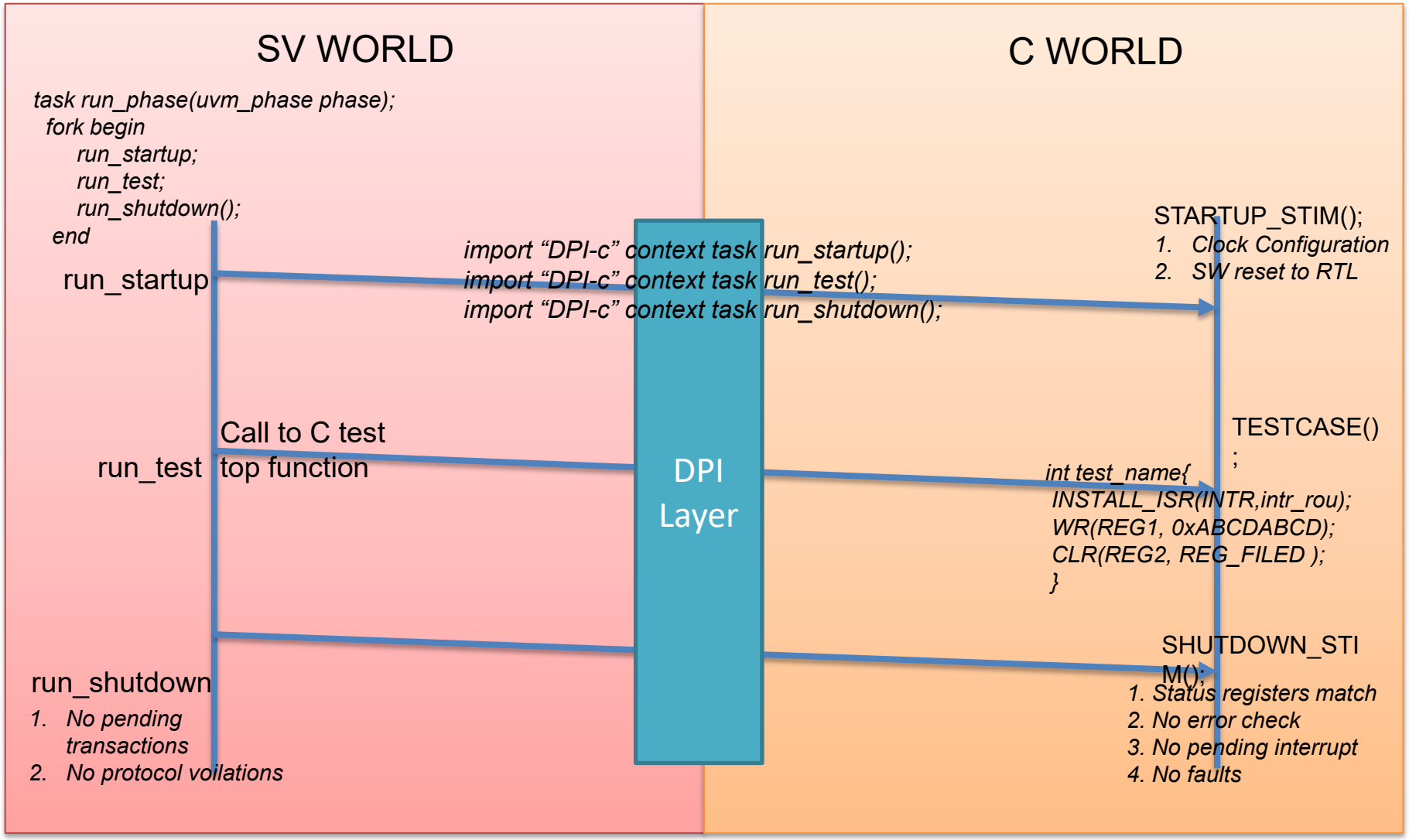
Stimulus Execution at SoC Level



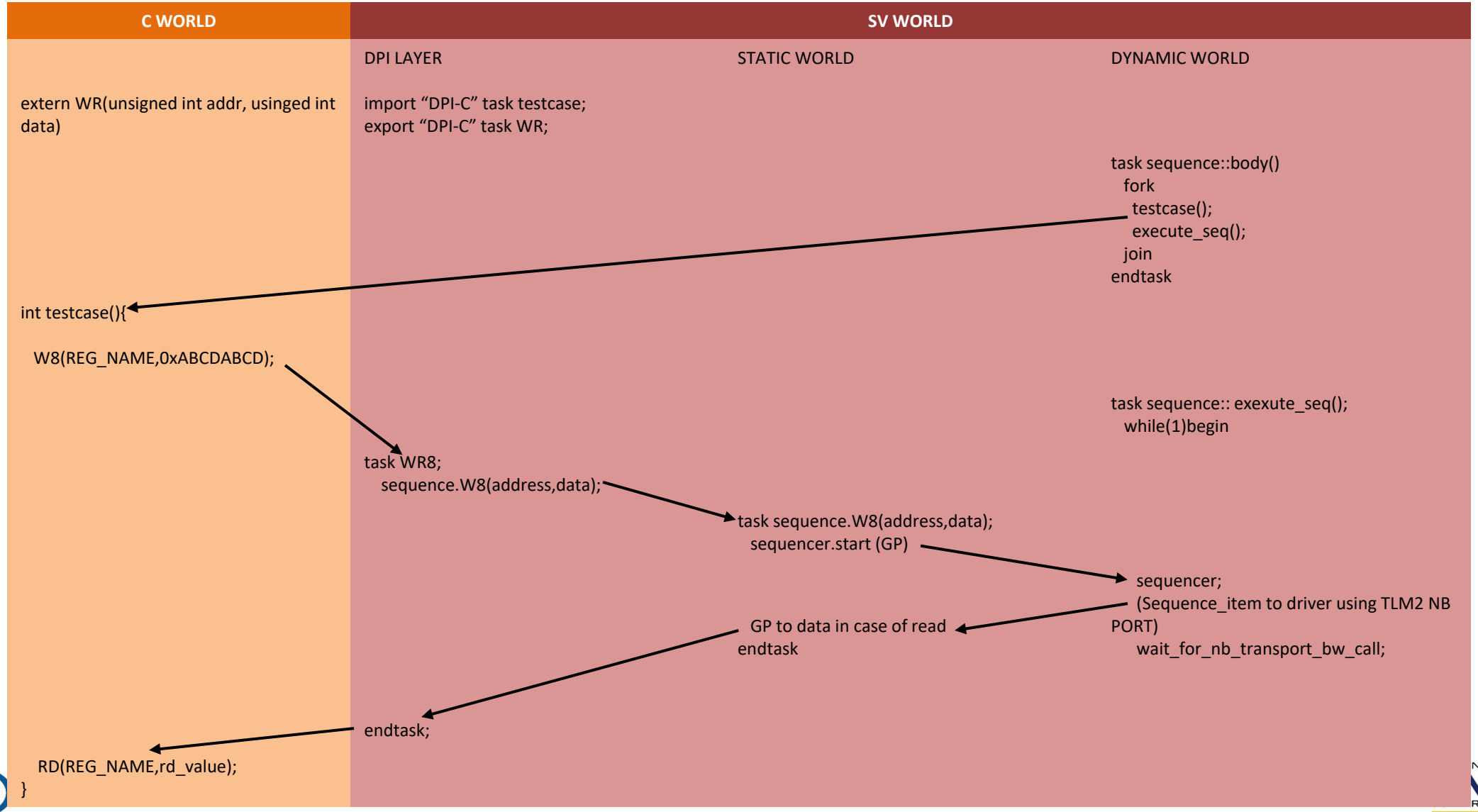
Major Requirements

- C and SV communication
- Test execution flow
- Reset Handling
- Interrupt execution
- Reusable-Interoperable-Protocol Independent test sequence

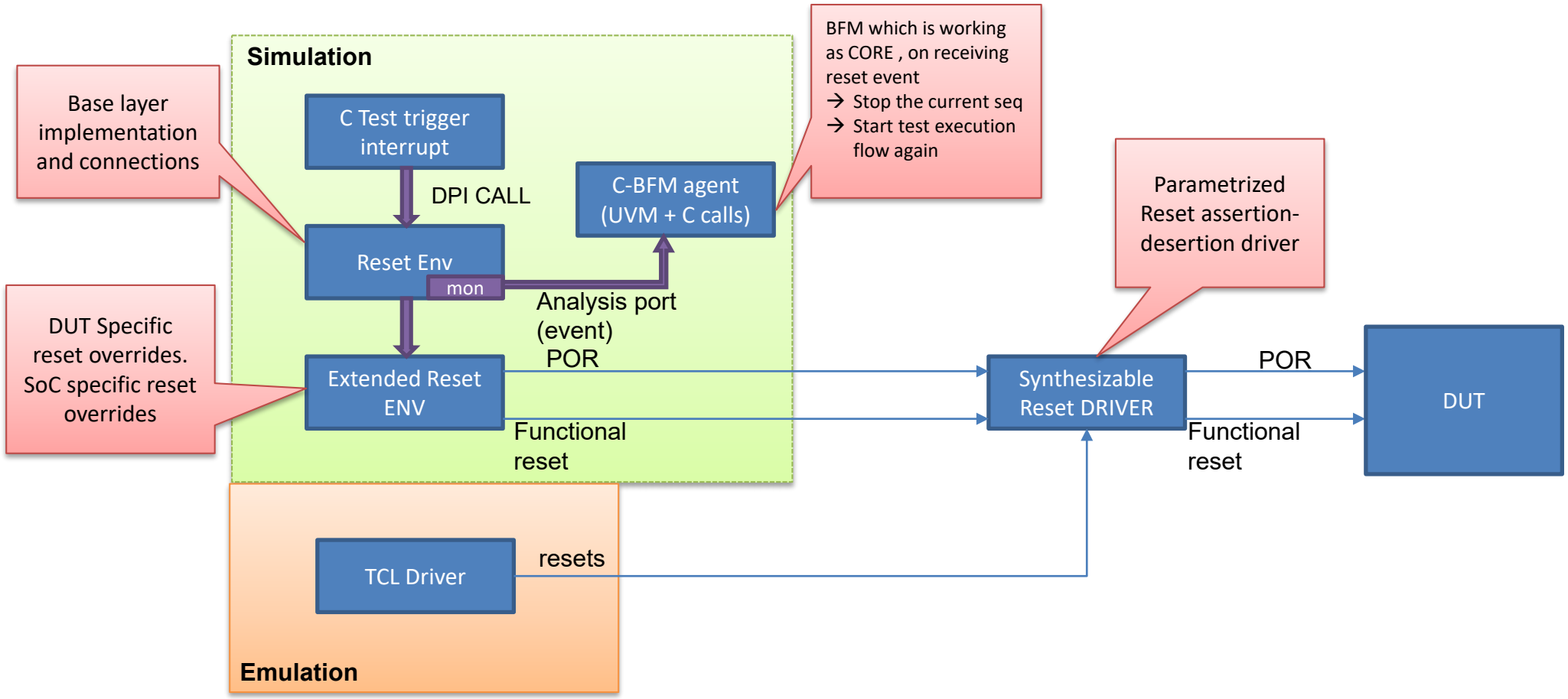
C and SV communication through DPI calls.



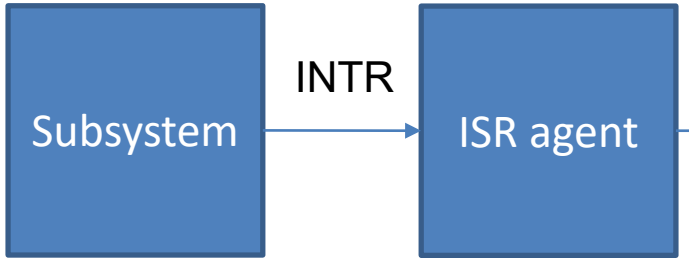
Test Execution Flow



Reset Handling



Interrupt execution diagram



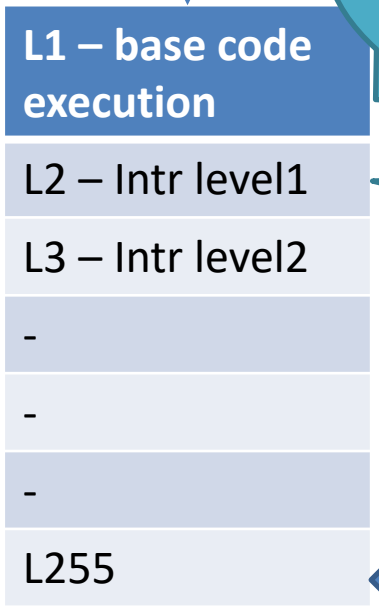
C world

```
INSTALL_ISR(INT_NUMBER, intr_routine)

void intr_routine{
    intr++
    CLR(INTR_REG,INTR_FILED);
}
```

SV world for interrupt execution

→ Lock the current execution layer
→ Add execution layer semaphore for +1
→ Execute the interrupt routine



Execution Semaphore: Application layers

1. Lock the execution current layer and add UVM objection for this layer
2. Move to next layer

Whenever there is interrupt

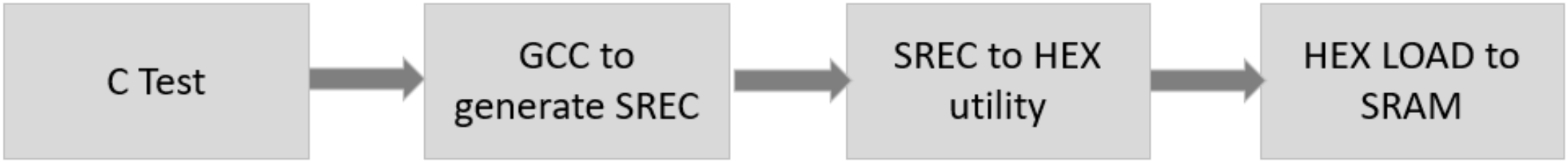
1. Grab the handle for sequencer
2. Execute service routine
3. Move to L-1 layer
4. Drop UVM objection raised for this interrupt.

TLM GP based C-agent(DPI calls based from C)

Grab the sequencer and fire intr_routine

Direct re-use of testbench and testcases in emulator

- Frontend/RTL Changes for emulation
- Testbench Changes for emulation
- Execution Flow



Application

- Stimulus creation that can be re-used in SoC level testbench.
- L-BIST and M-BIST verification.
- Gate level verification.
- Special purpose RTL simulations (Ex: To generate power analysis VCD).
- Direct re-use of testbench and testcases in emulator.
- DFT verification.

Summary

- Seamless reusability for native UVM verification.
- Faster RTL compilation/simulations
- Faster UVM-SV based driver leads early verification closure.
- Synthesizable Verilog based slave memories eased portability to SoC emulation.
- Interoperable and protocol independent implementation using TLM 2.0 GP compliance.
- SoC Independent tests development (i.e. even when the sub-system has not been integrated into the SoC).
- TLM 2.0 compliance testbench can easily plugged with SYSTEMC hosts for early low-level driver development.
- GLS faster run time and can be run independent of the SoC.

Questions

Finalize slide set with questions slide