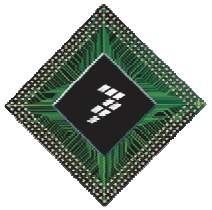
SW-HW co-Simulation



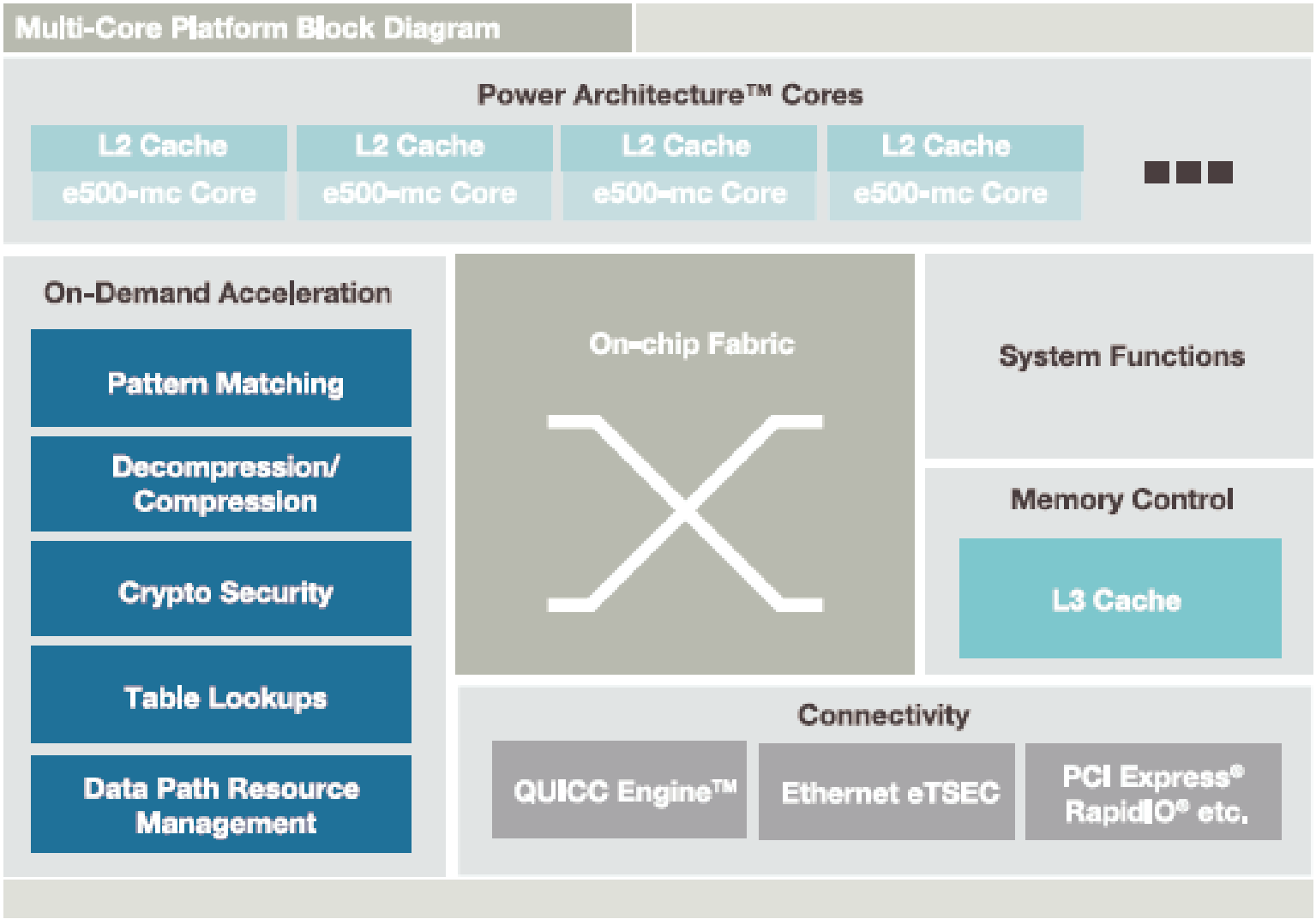
Amit Hermony
amith@freescale.com

freescale™ semiconductor

Preview

- Current design challenges – as inflicted multi core Soc's.
- Time to market is the design group goal.
- No one solution fits all.
- Finding a common ground of the different solutions.
- Implementation in different environments.

Current design challenges – as inflicted multi core Soc's



freescale™ semiconductor

Current design challenges – as inflicted multi core Soc's

- Design Size – slow simulation
  - Hard to debug
  - Long verification code development cycle
  - Not enough cycles to reach coverage goals .
- Design Complexity
  - Complex SW needed to utilize cores and data path.
  - Complex scenarios including coherency cases.
  - Hardware supported Data flows.
- Tight Schedule and resources.

Name of the game is Time to market
OR: What do we need to start selling?

- Provide Proof of concept
  - Modeling.
  - Power estimation
- High quality HW
  - Advanced verification tetchiness (and debug).
  - High coverage and cycle count.
  - Verify SW- HW interconnect.
  - Validate components (make sure standards are meet).
- Allow customer to develop SW (specially in multi core environment).
- Efficient silicon validation process.

Wide range of solutions
OR: No one solution fits all

- Provide Proof of concept
  - Modeling. → **SW model (Emulation)**
  - Power estimation. → **Emulation**
- High quality HW
  - Advanced verification tetchiness (and debug) → **Simulation**
  - High coverage and cycle count. → **Simulation and Emulation**
  - Verify SW- HW interconnect. → **Emulation (Simulation)**
  - Validate components. → **Emulation and simulation**
- Allow customer to develop SW. → **SW model (Emulation)**

- Efficient silicon validation process. → **Emulation / SW model**

freescale™ semiconductor

A Complex problem demanding different solutions
OR: A “Small” problem called resources

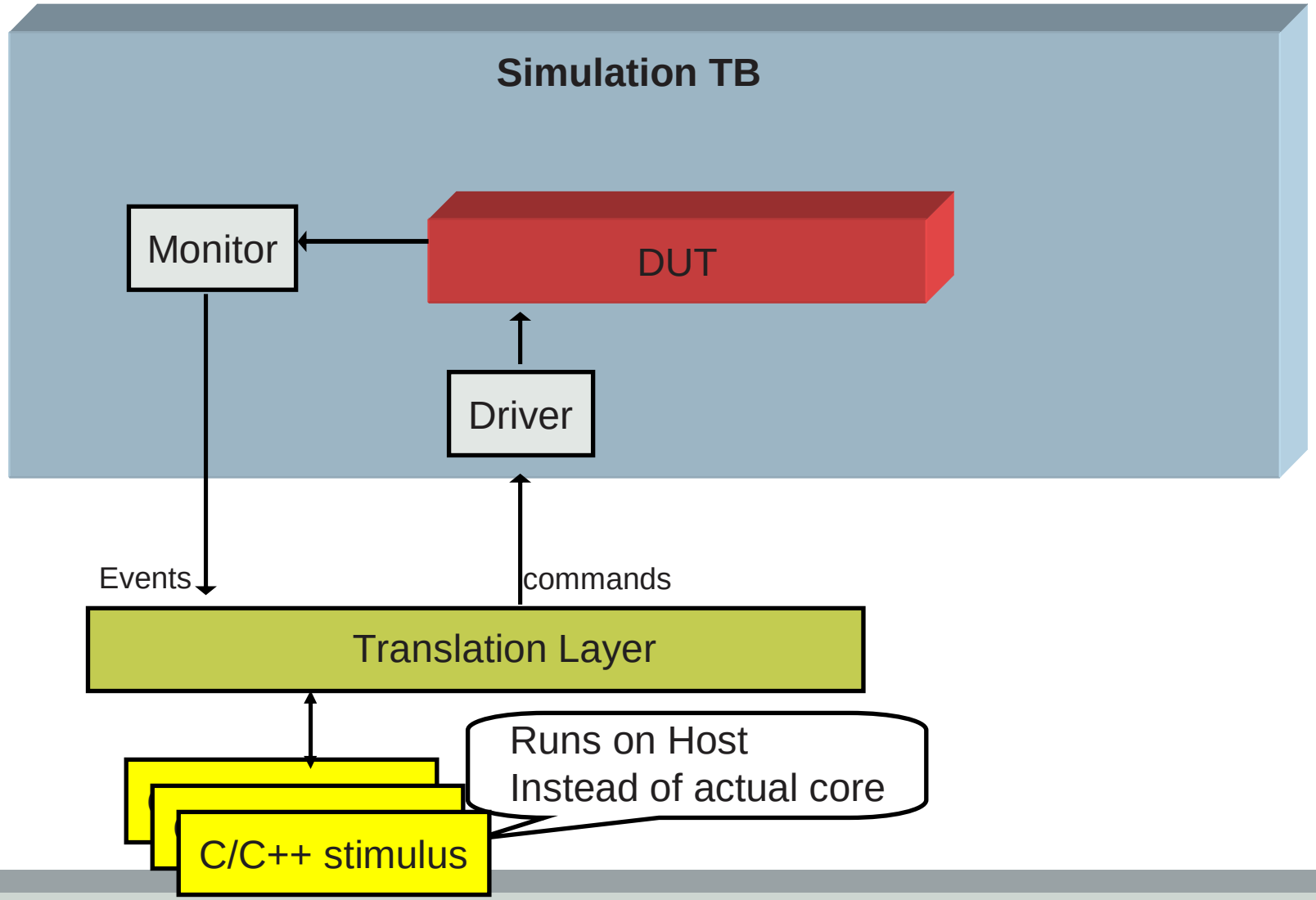
- Simulation – The back bone of the verification process
- Emulation – Needed to fill the gap in the verification of complex SOCs.
- SW model – Used by customers for SW development.

- Can we manage all and meet our R&D budget?

Finding a common ground for simulation, emulation, and SW models

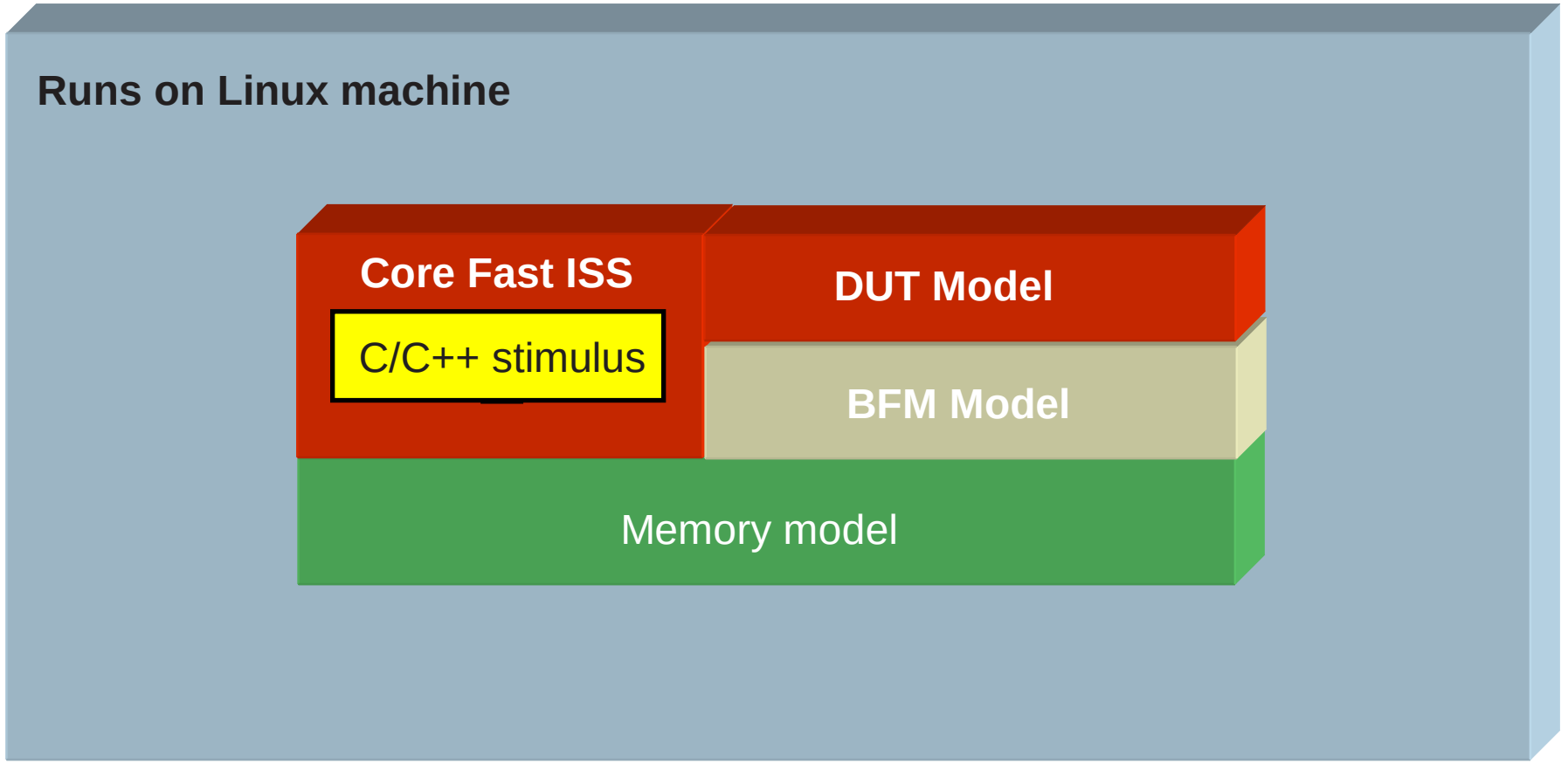
- **Reusable Stimulus**
  - ***C/C++ based*** - Can compile on core (for emulation) or run on host for simulation (using a systemC interface).
  - Stimulus has a ***programming view only***.
  - ***RTOS like scheduler*** or a SystemC based for simulation.
- **Common API for Simulation/Emulation BFM's**
  - Stimulus **randomizes a common API**. For each model there is a different implementation.
  - **Synthesizable BFM** for emulation.
- **SVA based monitors**
  - Provide information and coverage for emulation and simulation.

Reusable stimulus: Simulation view

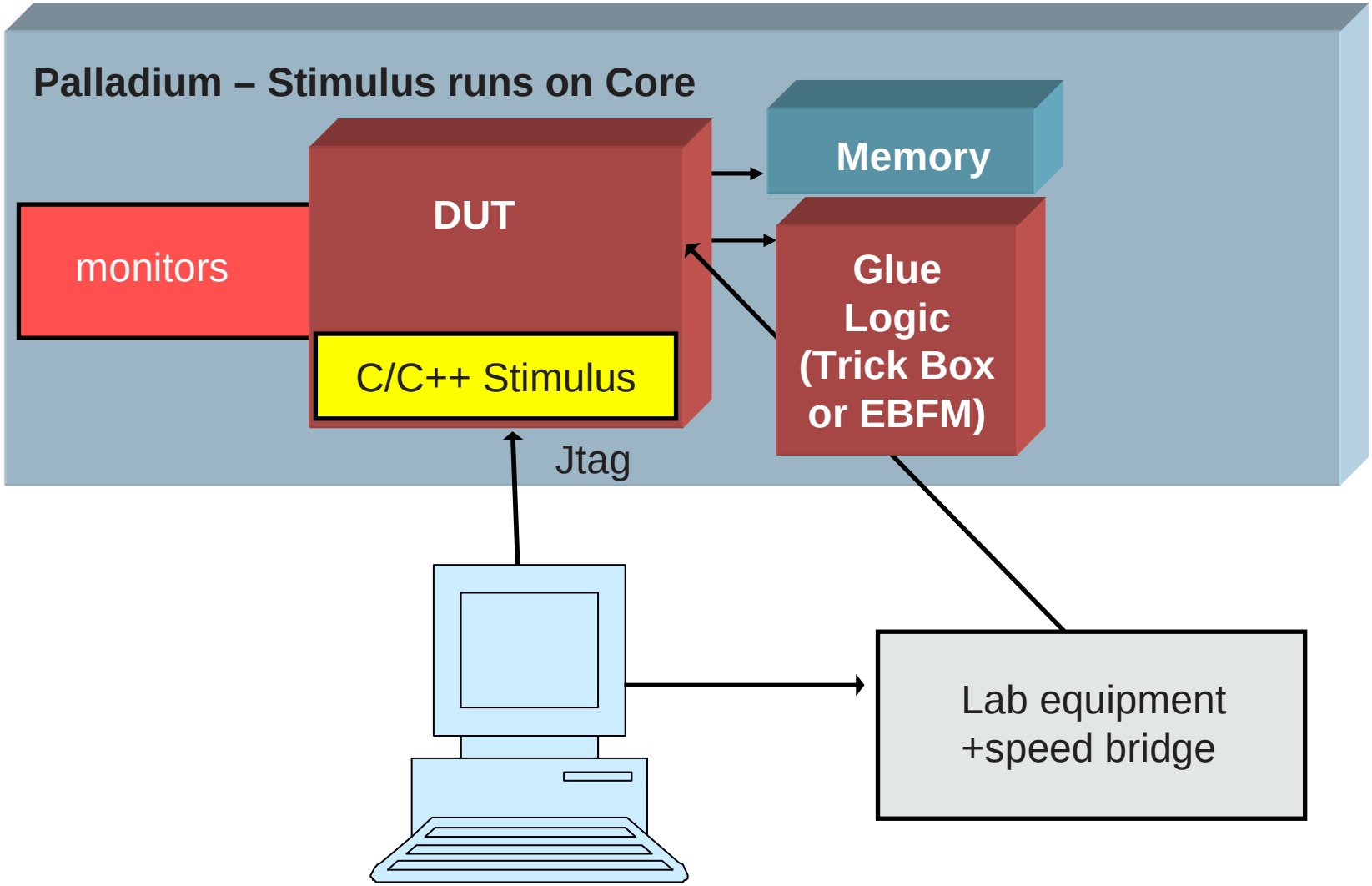


freescale™ semiconductor

Reusable stimulus: ‘C’ model Simulation view

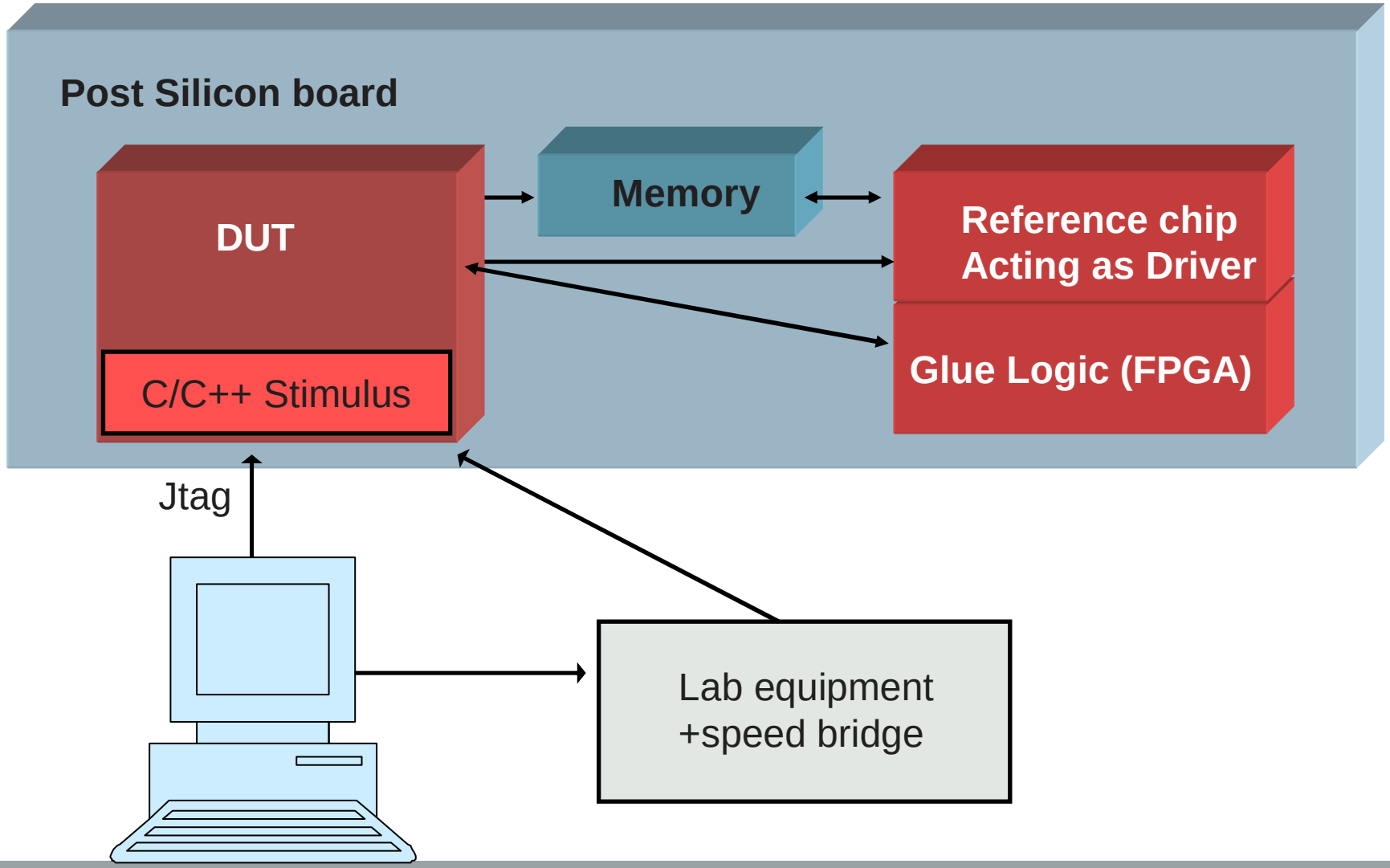


Reusable stimulus: Emulation model view



freescale semiconductor

Reusable stimulus: Post Silicon view



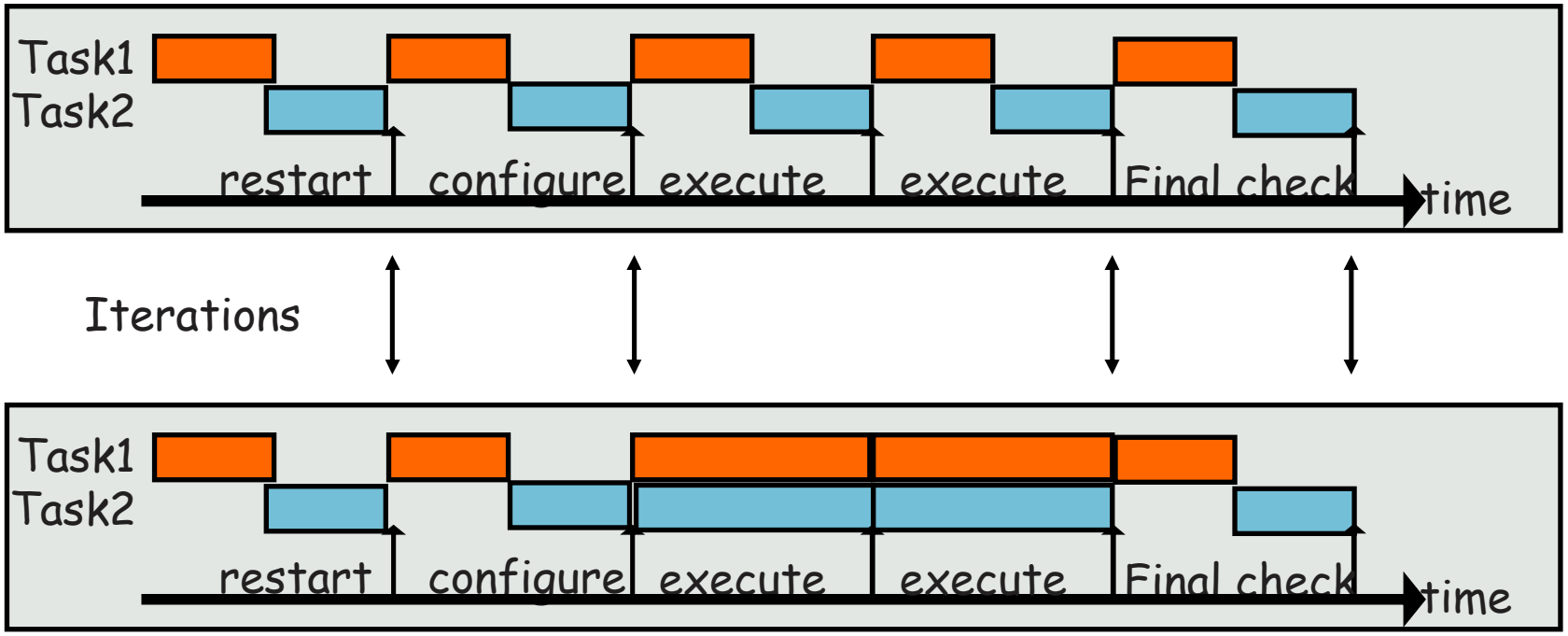
freescale semiconductor

Stimulus structure.

- **RTOS like scheduler** used to imitate simulation concurrency runs.
- Four Scheduling stages insures synchronization between different tasks:
  - **Restart**
  - **Configure**
  - **Execute**
  - **Final check**
- Scheduler waits for all tasks to finish a stage before it continues to the next
- An interrupt routine in each task

RTOS like Scheduling

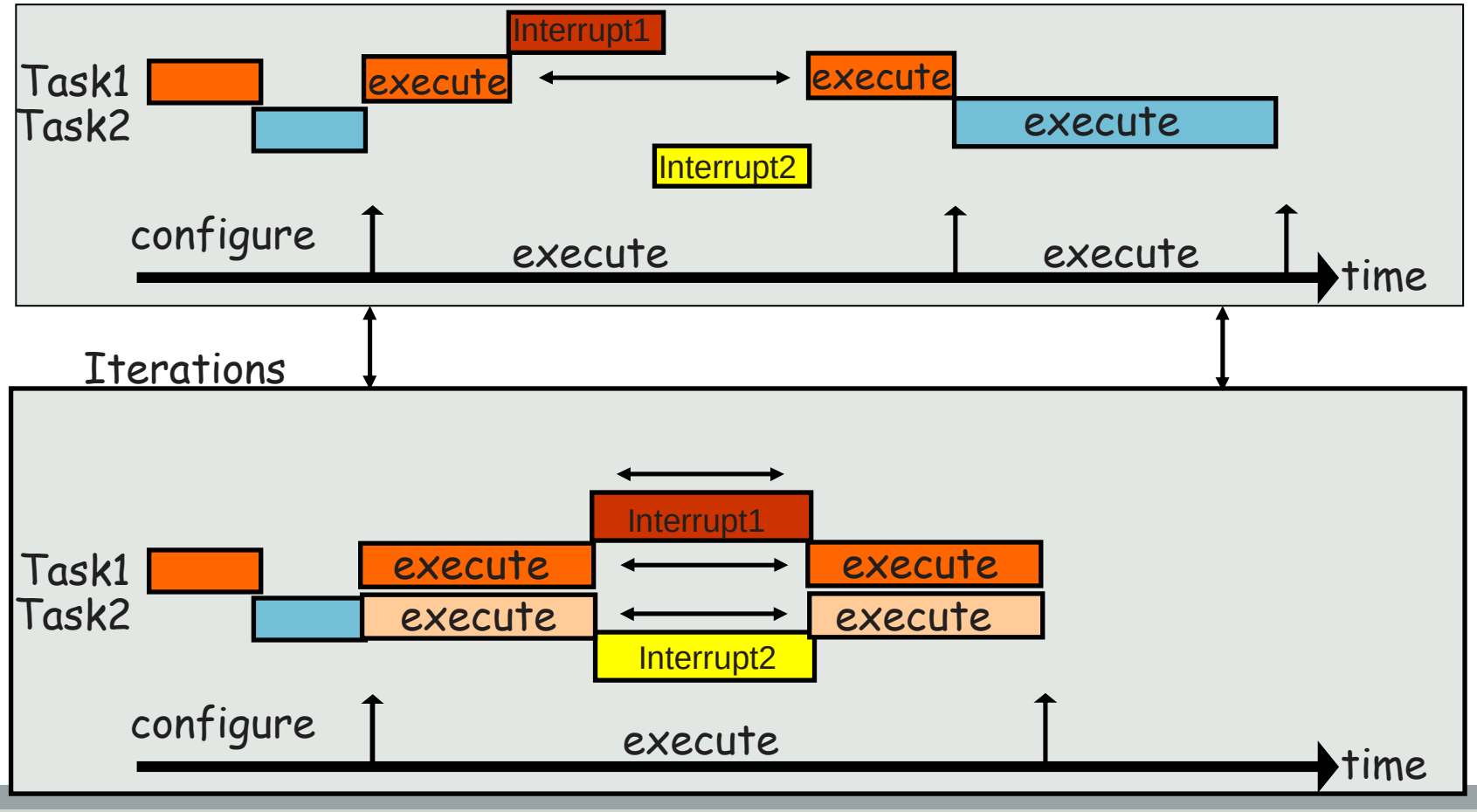
- Concurrent execution in simulation.
- User control over scheduling:
  - Every iteration or at each task call.



freescale™ semiconductor

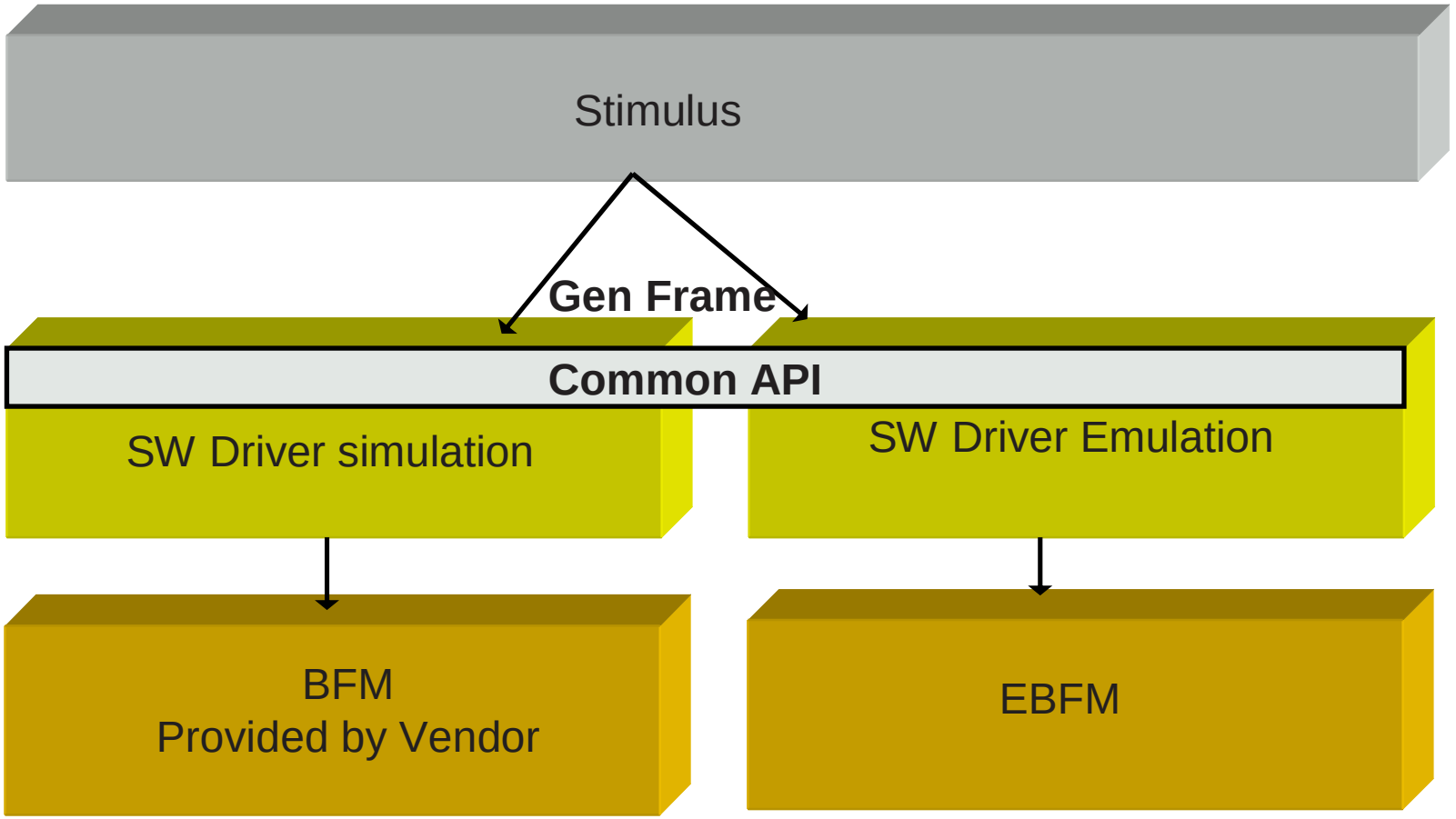
RTOS like : Scheduling - Interrupt

- Interrupt routine in each task is called.



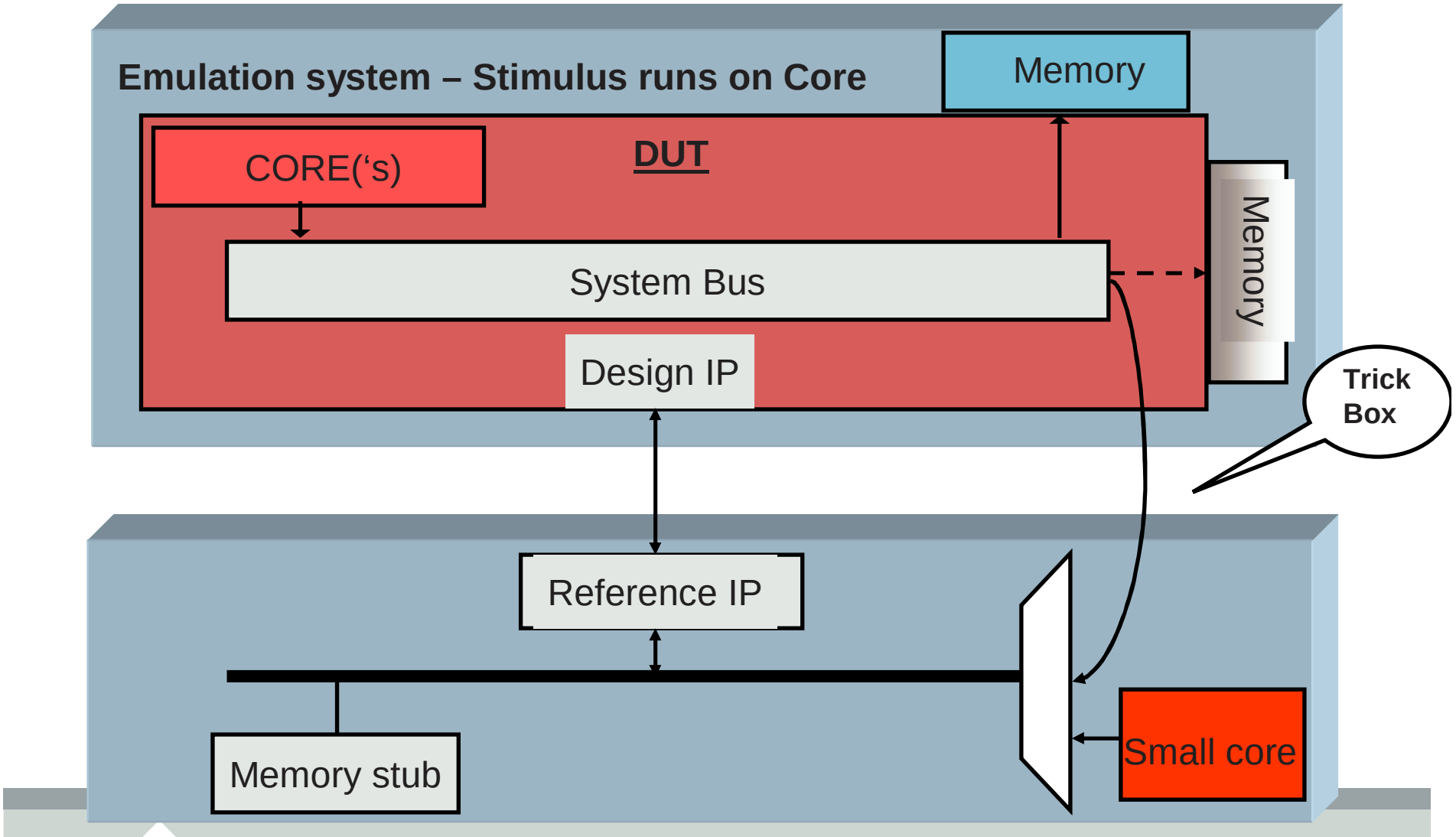
freescale semiconductor

Common API for BFM's: Stimulus Structure SW layers



freescale™ semiconductor

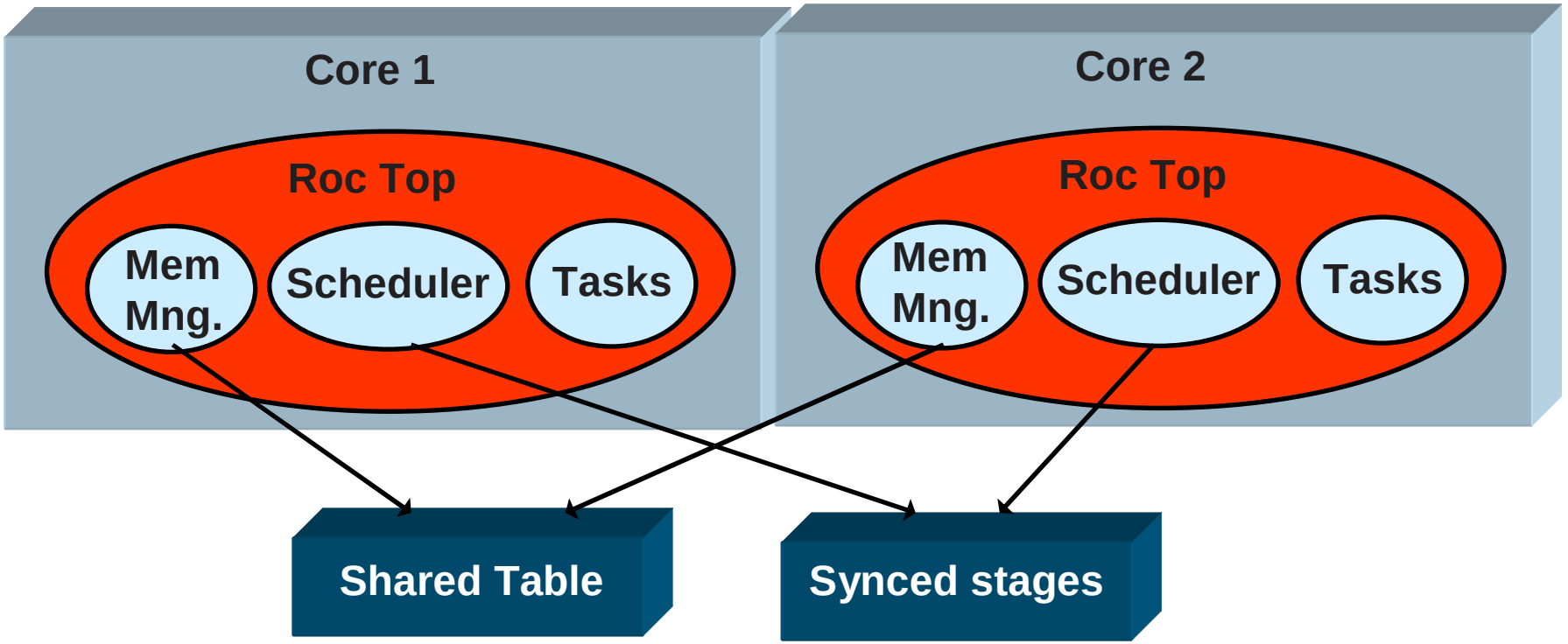
Emulation EBFM structure



freescale semiconductor

Test Bench Architecture Multi core

- Asymmetric Multi processing RTOS style
- Synced Scheduler stages between cores
- Shared memory allocation table



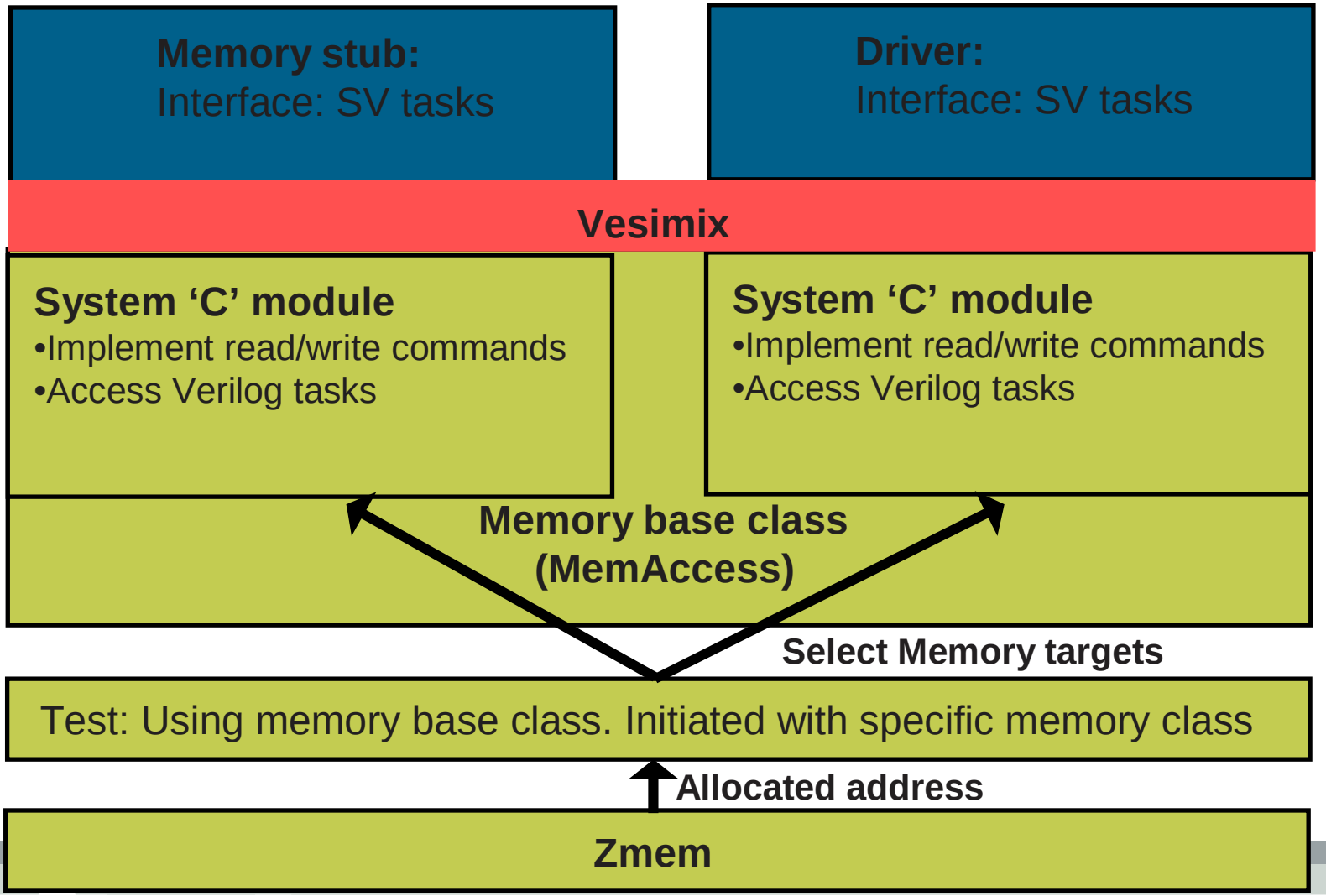
freescale™ semiconductor

Summery

- Current design challenges require to exercises multiple techniques.
- Solutions must fit into schedule and resource restrictions.
- Reusing code across different environments allows to meet design goals and development price.

BACK UP SLIDES

Memory Access



freescale™ semiconductor

Abstract

- Delivering an Embedded design to a customer goes hand in hand with delivering its software.
- The HW-SW product must work efficiently together, be ready on time and with outstanding quality without duplicating R&D effort.
- To achieve this extreme goal there is a need to check the SW-HW handshake using several platforms:
  - HW Simulation (RTL model)
  - SW simulation (SW model)
  - Emulation
  - Post silicon.
- In this presentation a tool set and methodology allowing to achieve all of the above will be presented:
- I will give an example from Freescale's new Multi core product : (http://www.freescale.com/files/32bit/doc/white_paper/MULTICOREFTFWP.pdf) – Where RTL simulation, SW simulation (using Virtutech model) and emulation (using Palladium) is used.


freescale™
semiconductor