DVCon 2013
Design & Verification Conference & Exhibition

February 25-28, 2013
DoubleTree, San Jose

accellera SYSTEMS INITIATIVE

Memory Subsystem verification – Can it be taken for granted ?

by
Shivani Upasani
LSI Bangalore

LSI®

Storage. Networking. Accelerated.™

Sponsored By:

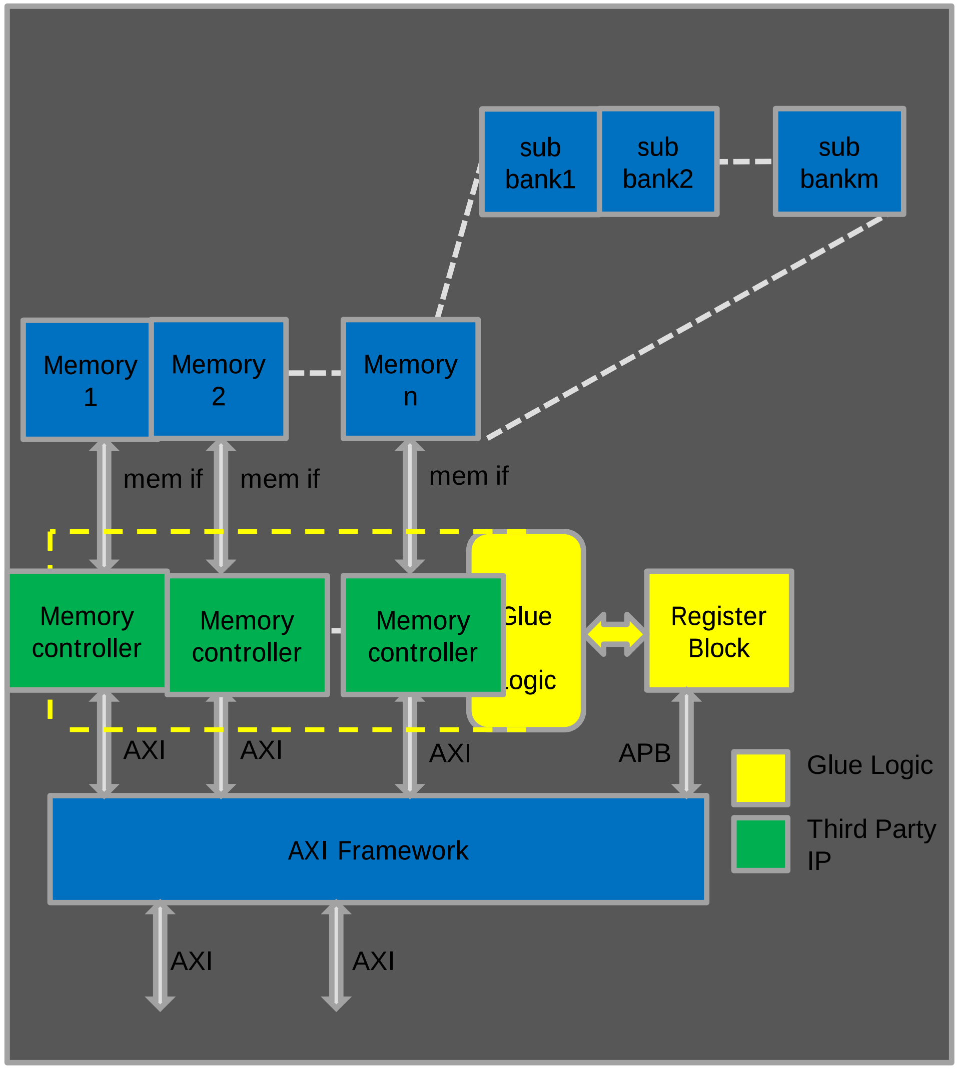
SYSTEMS INITIATIVE

Agenda

- Introduction to the Design under test (DUT)
- Scoreboard approach
- Interrupt and Exception Checker
- Low-power mode verification of memory
- Results and Conclusions

Sponsored By:

DUT overview



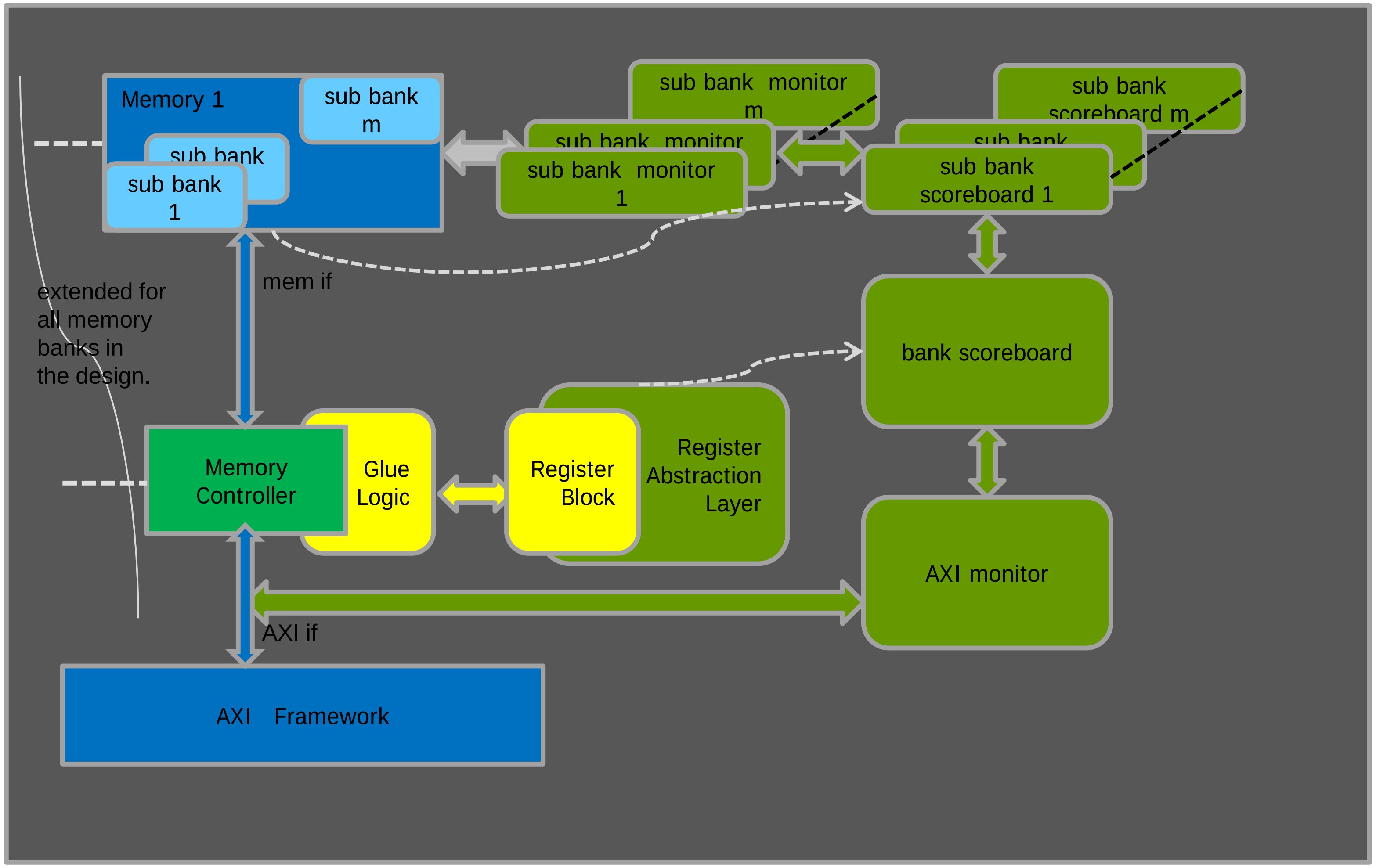
- Scalable architecture
  - Multiple memory instances
  - Each memory consisting of multiple sub-banks
  - Memory controller
- AXI Framework
- Glue logic to connect to Register Block
- 128 bit AXI Interfaces
- APB Interface

Verification Requirement of the Memory subsystem

Sponsored By:

- Data Integrity Check
  - To confirm that the data on the AXI Framework is correctly getting written into the memory and data read from memory is correctly being returned to the AXI framework.
- ECC functionality
  - We need to verify the ECC functionality of the third party IP.
  - We need to verify the Glue logic written to update the ECC errors, address, and syndrome values into the register block.
- Low-power mode features
  - The light sleep, deep sleep, and shut down features of the memory needs to be verified.

Verification Environment



Sponsored By:

Data Integrity Scoreboard Approach

- Two methods can be adopted.
  - Reference model approach.
  - Non-reference model approach (byte level scoreboard, packet based scoreboard, or backdoor read scoreboard)

- In Reference model approach,
  - DUT functionality needs to be modeled.
  - Exact number of reads and writes are predicted.
  - Effort is more in this approach.

Data Integrity Scoreboard approach continued

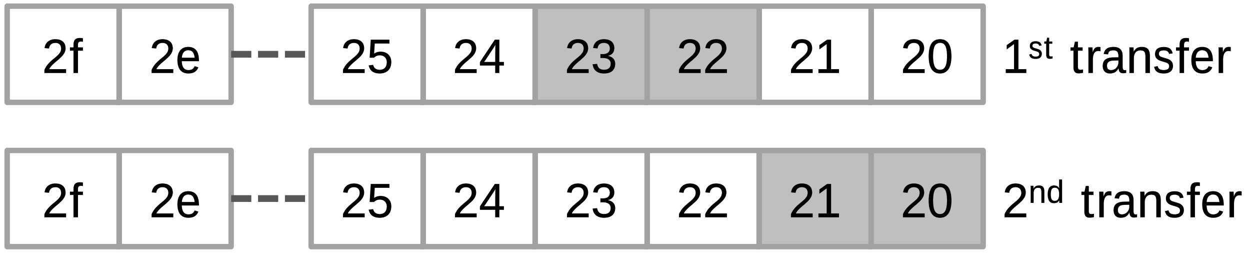
- In a non-reference model approach

- We don’t need to model the DUT behavior.
- Data can be tapped from the interface using monitors either in a packet format or byte-by-byte
- It is easier to implement
- Exact number of reads and writes are not predicted.
- Issues like redundant reads can be missed.

Example of Redundant Read

- In case of AXI wrap read transfers, if the address wraps back to the same memory word boundary, two reads are initiated (first read for first beat and second read when it wraps back), though a single read is sufficient.

Address : 0x22
Transfer size : 8 bits
Burst type : wrapping
Burst length : 4



Aligned wrapping byte transfer on a 128 bit bus

Sponsored By:

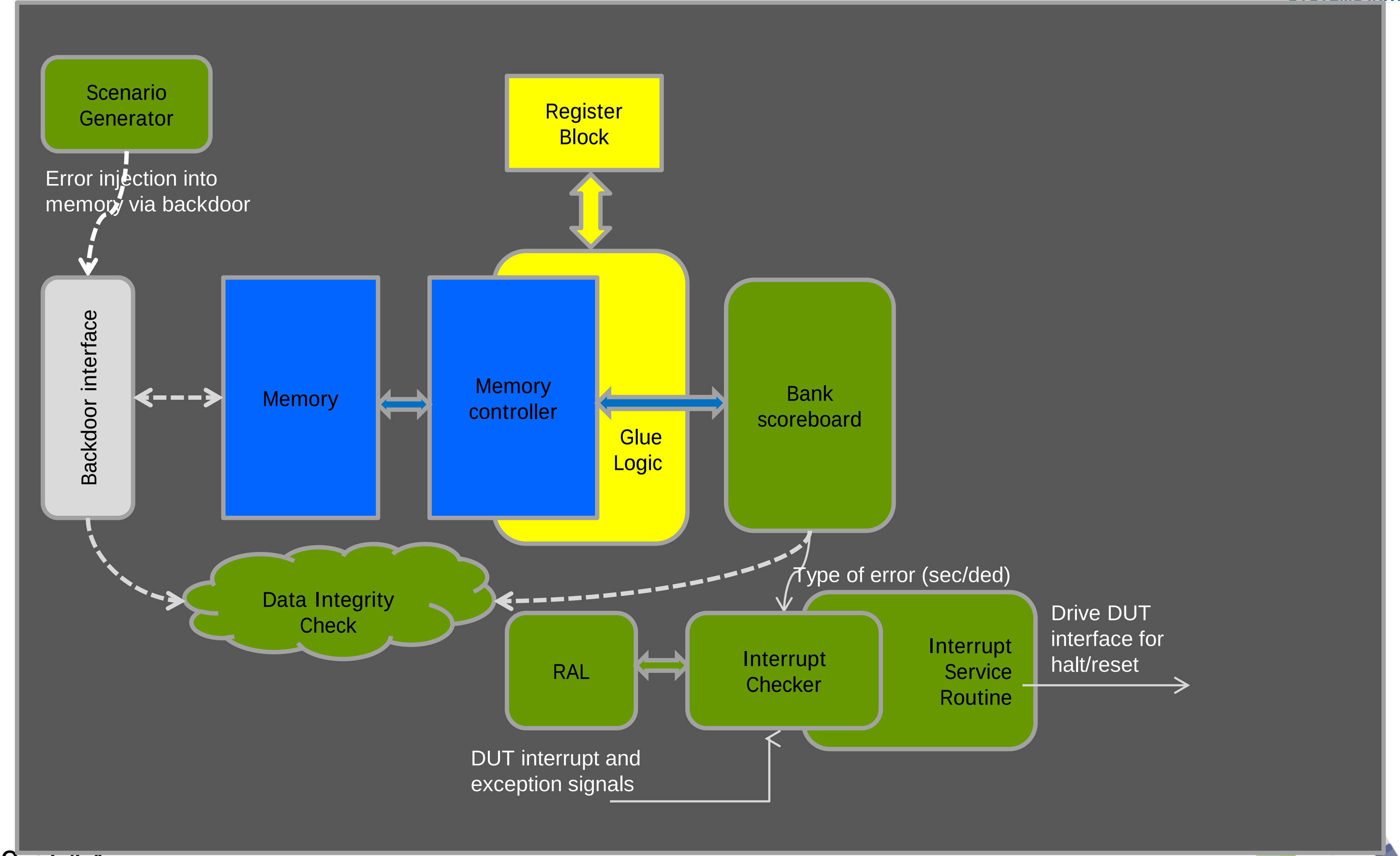
ECC Functionality check

- Requirement
  - Real time errors occurring in the memory can result in a system failure and hence it is important to verify them.

- Verification Strategy
  - Inject single bit and double bit errors in the memories via backdoor.
  - Generate access to those locations and check for data integrity.
  - Check for the interrupt/exception generation and handling at the system level.

ECC Interrupt and Exception Checker

Sponsored By:



Sponsored By:

Interrupt and Exception Checker

- Independent threads to check the interrupt and exception in case of single bit and double bit errors. This helps in case of back to back errors.
- Uses parameterized RAL register classes for passing the registers to the checker.
- Checks for the bank error address, syndrome, transfer id, transfer type, and error type correctly getting latched in the bank status reporting registers
- Uses the Register Abstraction Layer (RAL) in the environment for reading all these register values using backdoor to avoid any delays in the checker.

Parameterized RAL register type

```
`define STATUS_BANK_REG_1
ral_reg_addr_map_register_file_reg1
`define STATUS_BANK_REG_2
ral_reg_addr_map_register_file_reg2

class bank_checker #(type
STATUS_REG1 = `STATUS_BANK_REG_1,
STATUS_REG2 = `STATUS_BANK_REG_2)
extends bank_checker_base
....
function new(
STATUS_REG1 reg1,
STATUS_REG2 reg2
);
....
endfunction
....
endclass
```

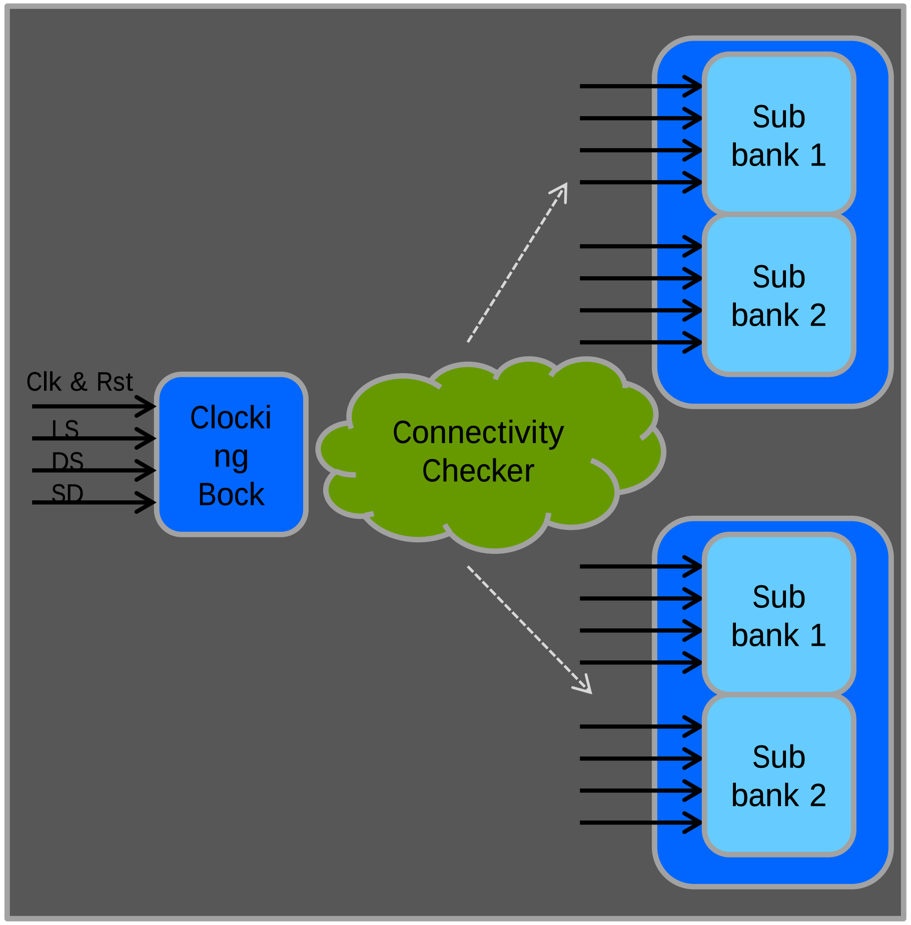
Interrupt Service Routine(ISR)

Sponsored By:

- Uses RAL backdoor to simultaneously read the status register while the interrupt while exception checker is also operating on the same register
- Mimics system level behavior and issues halt or reset to the DUT
- Operates on bit wise status register and clears them one after the another. Keeps a check that unless all the status bits are cleared, the interrupt and exception should remain asserted.

Low Power Mode verification

- Light sleep, deep sleep, and shut down modes of a memory are verified
- Connectivity checks are carried out to confirm the TOP level power-mode signal correctly connected to the memory bank and sub-bank power mode signals.
- Functional checks are carried out by writing to and reading from the memory during low power and comparing actual versus expected behavior.



Sponsored By:

Conclusions

- We are able to find power efficiency and performance issues with the third party IP by using the reference model scoreboard approach.
- The environment is scalable and can be upgraded easily using few defines to support any new memory configuration change.
- Low-power modes of the memory are verified by carrying out connectivity checks and functional checks.