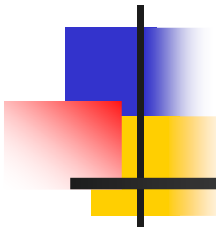
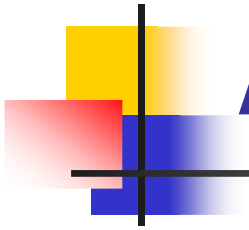


Verification Environment Architecture

Amir Bar-Dor – Texas Instruments

Ronen Shoham – Siverge Networks

May - 2006



Agenda

- Components of the Verification Environment
- Heritage in verification IPs
- Module Level to Chip Level
- I/F Verification IPs architecture

Texas Instruments

Siverge networks

Components of the Verification Environment

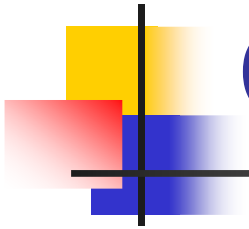
- System - test flow, resets, clocks, etc.
- I/F IPs
- Reference Model IPs

Environment system architecture

- The “system” of a VE should handle:
  - Clocks
  - Resets
  - Test flow mechanism
  - End of Test mechanism

Texas Instruments

Siverge networks



Clocks and Reset

- All clocks are generated in RTL
  - Verification environment can influence:
    - Clock period
    - Clock duty cycle
- There are two sorts of resets: RTL resets and "e" resets.
  - The RTL reset is not used much in the bfms and monitors.
  - Bfms and monitors synchronize on a reset in "e". In this manner, one can better control the start of each verification component.

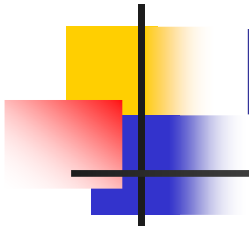
Test Flow Mechanism

- Benefits of structured test flow mechanism:
  - Easy synchronization between different parts of the VE
  - Easier reuse of scenarios between tests and different stages in the flow
  - Each user need to access only to necessary phases
  - Helps in debug
  - If not needed a phase can be skipped
- Example of test phases

  reset_done, init_dut_ended, start_flow_ended
  scenario_start, scenario_end, stop_activity
  test_done

Texas Instruments

Siverge networks

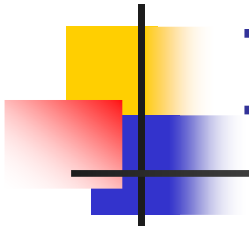


End Of Test Mechanism

- Components that should "object" to end of test:
  - Generators
  - Monitors
  - Scoreboards
- When to stop the test:
  - "IDLE" state
  - N items done
  - Reached X number of errors
- DUT state at end of test:
  - Drained
  - Stable
  - Not important

Texas Instruments

Siverge networks



I/F IPs

- All of the eVCs we develop use a common skeleton and philosophy.
- We take a very strict approach to eVC implementation.
  - There are naming conventions for tcms, methods
  - Only elements that are related to standards are allowed in the eVC.
- Advantages:
  - Common structure, easier adaptation to Chip Level, easier inter-engineer understanding, standard-compliant.
- Disadvantages:
  - Learning curve for new employees, more files (is it a disadvantage?)

Texas Instruments

Siverge networks

Module Environment vs eVC

- In our architecture, there is a difference between the Module Environment and the eVC.
- The Module Environment contains an instantiation of the eVC and extensions with all the design-specific elements needed for operation and verification of the eVC.
- Among the main elements of our module environments are:
  - Scoreboards
  - Register Files
  - Higher layer data formats (i.e. packets)
  - Non-standard additions/modifications
  - Signal pointers

Heritage in verification IPs

- Using common Structs/Units/Classes in VE can help in:
  - Easily adding new features to all IPs
  - Using generic objects (e.g Scoreboards)
  - Maintaining lists of objects from different IPs
  - Better understanding by non-IP-designers
- Creating Templates/Utility IPs allows:
  - Implementing/Enhancing/Maintaining common mechanisms once
  - Better connectivity between IPs (Important in Producer-Consumer protocols)
  - Faster development time

Module Level to Chip Level

- A good verification environment is key for a smooth and productive transition from Module Level to Chip Level.
- If the environment is thought-out and eVCs are properly designed (as touched upon previously), building the Chip Level environment takes much less time than otherwise. A good, consistent architecture saves time and resources by:
  - Avoiding the need for re-writing code
  - Coverage metrics that can be imported
  - Updates in the eVC are immediately seen in the CL env

Texas Instruments

Siverge networks

I/F Verification IPs architecture

- When planning an I/F IP it is important to consider the following points:
  - What is the architecture of the DUT?
    - Point to Point Bus
    - Serial (Rx/Tx)
    - Single Master – Multiple Slaves bus
    - Multiple Masters/Slaves bus

I/F Verification IPs architecture cont.

- What are the I/F components needed in the VE:
  - Masters
  - Slaves
  - Arbiters
  - Decoders
- Scaling of the IP Vs. the DUT
- Which components must be implemented in the IP
- Future uses (in chip level, other projects…)

Example of an eVC architecture (TI)

- In designing our eVCs, our frame-of-reference is the guideline and methodology developed by Cadence (Verisity).
- All our eVCs have the following elements:
  - An agent
  - Within the agent there is (at least):
    - a bfm
    - a sequence driver
    - a monitor
    - A signal map (empty)
- Our eVCs do not have the following elements:
  - Scoreboards
  - Pointers to Dut signals & ports
  - Anything that is implementation specific

Texas Instruments

siverge networks

An eVC example