UVM usage for selective dynamic re-configuration of complex designs

Kunal Panchal

Pushkar Naik

applied micro

accellera SYSTEMS INITIATIVE

2014 DESIGN AND VERIFICATION DVCON CONFERENCE AND EXHIBITION INDIA

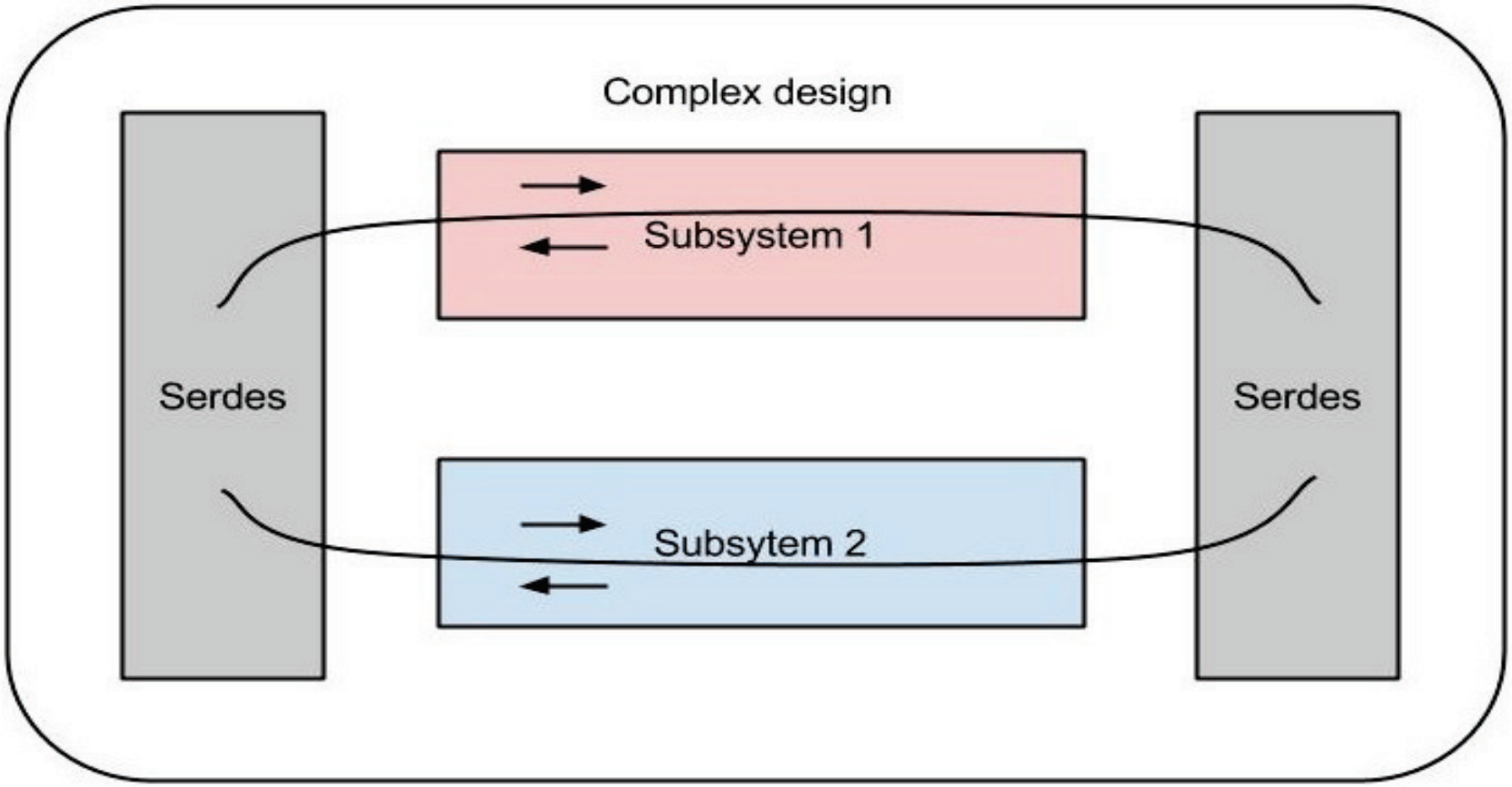
Agenda

- Introduction
- Sample DUT
- Verification Considerations
  - Basic Recommendations
  - Limitations Faced
- Solution Proposed
  - Flow Chart
- Conclusion

Introduction

- Verification of single sub-system chip
- Demand for multi-subsystem chip
- Verification complexity of multi-subsystem chip
- Re-use of proven verification env(s)

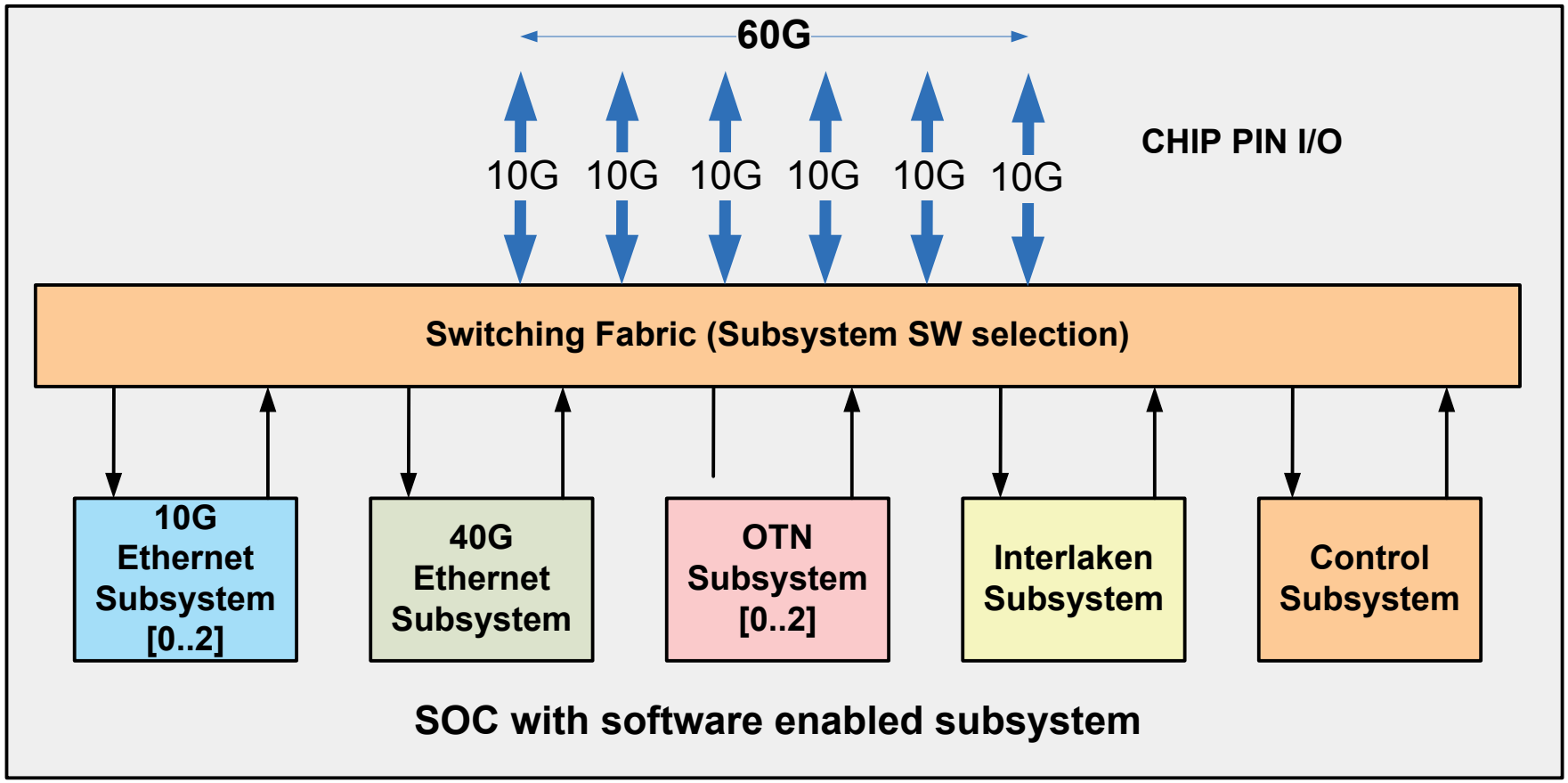
Multi-Subsystem Chip



Sample DUT

- Subsystem variation parameters
  - Rate Variation (10G, 40G ETH)
  - Protocol Variation (ETHERNET, OTN, INTERLAKEN)
- Switching Fabric for Subsystem to I/O's selection using software
- Total 6 basic I/O channels of 10G rate each to provide 60G aggregate BW
- Multiple channels clubbed to form higher BW pipe

10G/40G Networking PHY



Sample Test Scenario

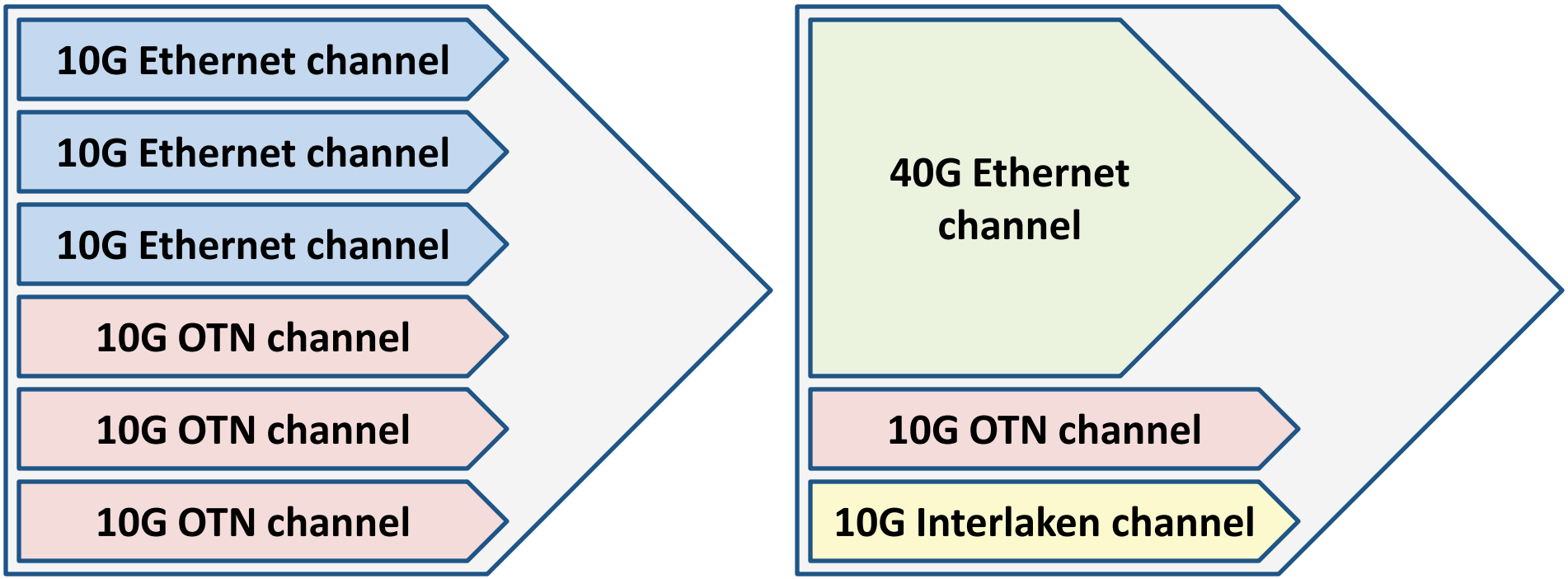
- Initial Setup
  - 3 x 10G ETH Channels
  - 3 x 10G OTN Channels
- Setup after reconfiguration
  - 1 x 40G ETH Channel
  - 1 x 10G OTN Channel (Untouched)
  - 1 x 10G INTERLAKEN Channel

BW allocation for Dynamic Reconfig

1st Setup:
60G Chip supporting 6x10G protocols

TO

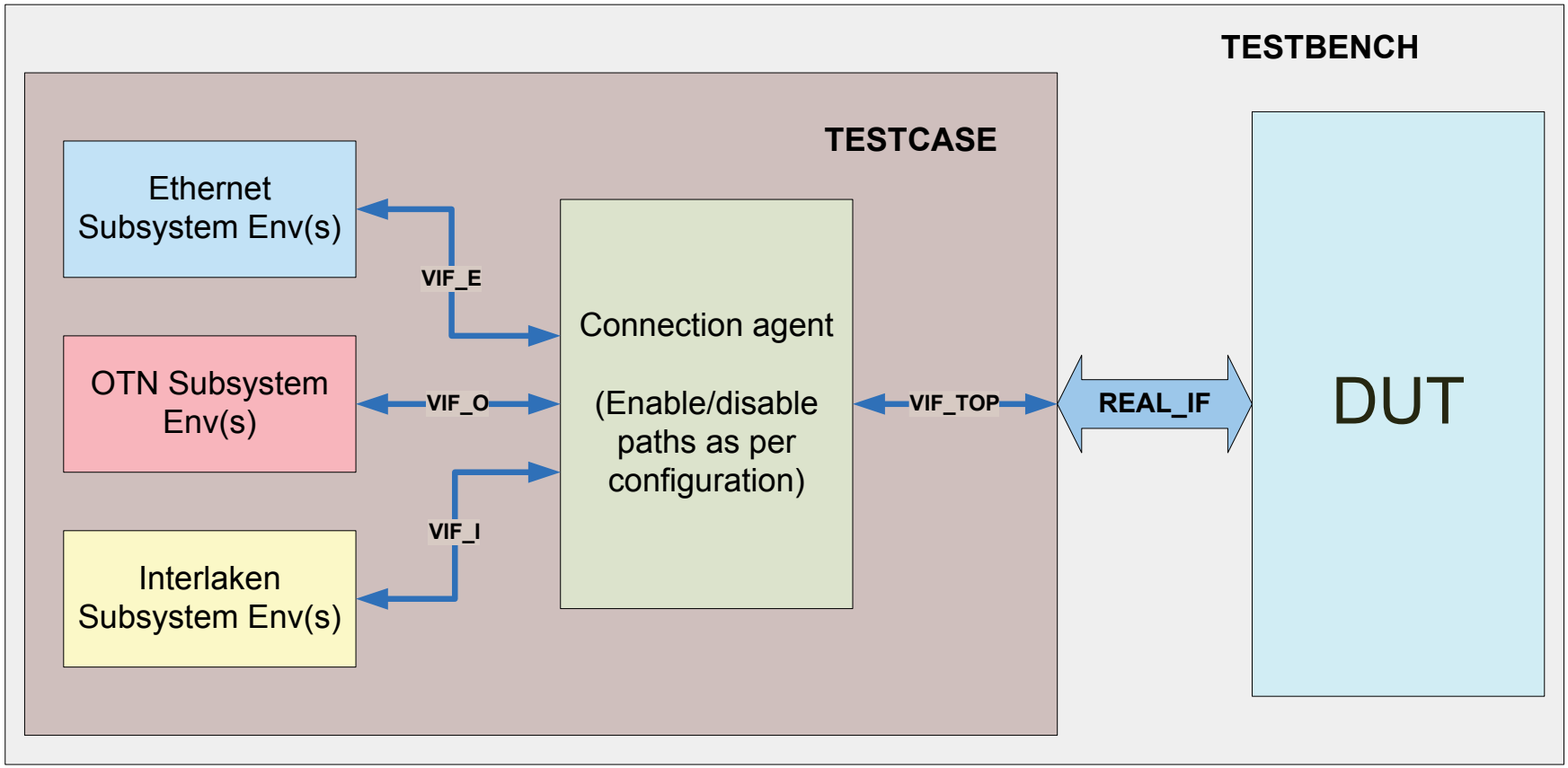
Setup after reconfiguration:
60G Chip supporting 1x40G & 2x10G protocols



Verification Considerations

- A DUT Subsystem with given Rate/Protocol variation forms one UVM environment
- A test case will create as many Env's as required by test scenario's Rate/Protocol requirement

Multi-Environment Test Bench



Verification Considerations

- Interface connections
  - An interface with all signals across all DUT features is created
  - Each Env has its own Virtual Interface instance created of above Interface type
  - Idea is that respective Env touches only those signals that it is supposed to
  - A top level Virtual Interface instance is created, again of same interface type, that finally connects to DUT's Real Interface
  - As per Env's "active" status, its local VIF is connected to the top level VIF, using the connection agent shown

Basic Recommendations

- Use *uvm_config_db* for Env components config
- Env components with enable/disable config as per datapath(sb, mon, drv, etc)
- Configurable clocks as per Reconfig requirements
- Proven tests of every subsystem env, tested individually, forms base for Dynamic Reconfig Test Scenario's
- Reset to all TB components and DUT must be in sync, and placed in *reset_phase*
- Any DUT specific reset requirements at the time of reconfig should be carefully taken care of

Basic Recommendations

- Data sequences lifetime need to be adjusted correctly
  - Env's untouched need to have sequence(s) running till end of test
  - Env's being phased out need to have their sequence(s) end right at Reconfig time (Graceful exit)
  - New Env's being created need to start sequence(s) after they are correctly configured/connected.
- Common Reconfig tasks to be placed in *base_test* for all test scenario's to leverage upon
- Untouched channels should not loose data integrity, ever
- Newly made active channels should eventually establish data integrity

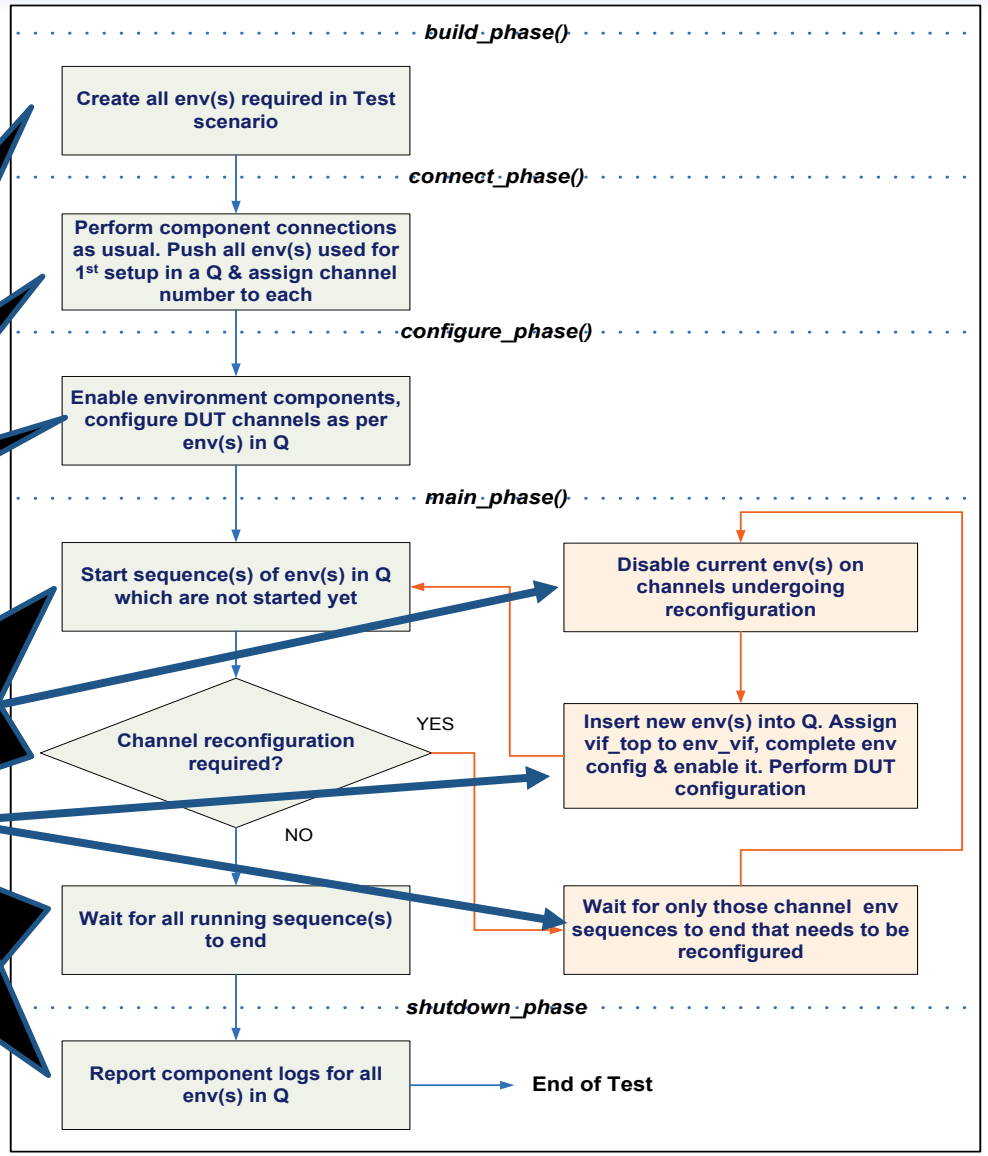
Limitations Faced

- Large configuration requirement of components against standard approach
- UVM's constraint on creation of components outside *build_phase*
- UVM's restriction on connection of TLM ports outside *connect_phase*
- UVM's phase looping feature cannot be used for Reconfig since few Env's are required to continue their data sequences 'untouched' while few new are required to be created

Solution Proposed

- As per test scenario, all the required environments (both pre and post reconfig) will be created in *build_phase*
- TLM port connections, if any, will be done in *connect_phase*
- Switching of Env's is done using dynamic connection of Env's local VIF to the top level VIF during the *main_phase*
- Logs reported from all env(s) to conclude test Pass/Fail

Multi-Channel Reconfig Flow Chart



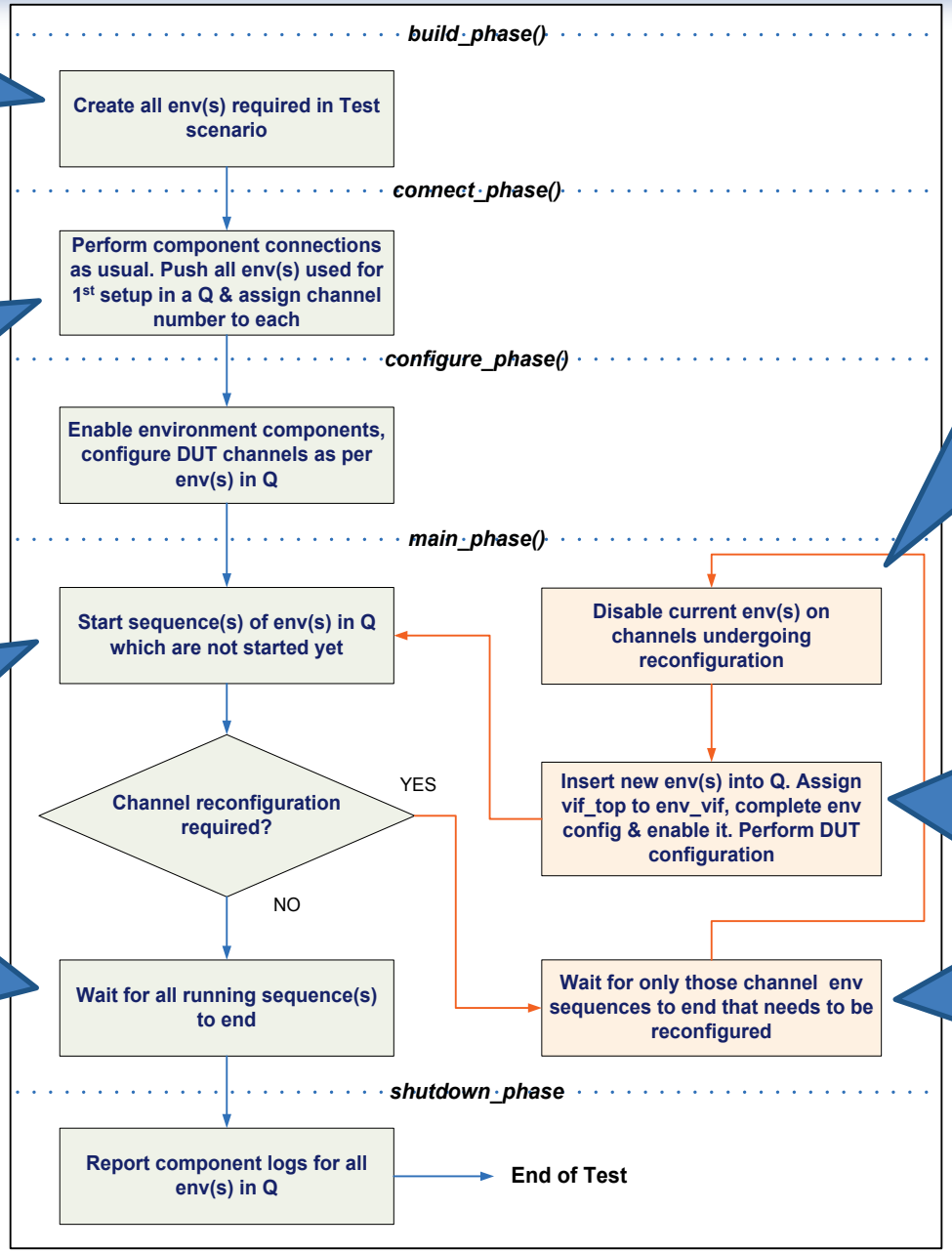
Report component logs for all env(s) in Q

8 env: 3x10G enet, 3x10G otn, 40G enet, 10G ikn

1st Setup for 6 channels: 3x10G enet, 3x10G otn

Start seq(s) of env in Q (3x10G enet, 3x10G otn env)

Wait for seq(s) to end for inst 40G enet on ch0, 10G otn on ch4 and 10G ikn on ch5



Disable 3x10G enet on ch 0-2 and 2xotn on ch 3&5. (stop clocks, mon, drv, etc)

Add 40G enet on ch 0 & 10G ikn on ch 5. Assign env_vif to top_vif

Wait for seq(s) on ch 0-3 and 5 to end (ch 4 is untouched)

Conclusion

- Multi-subsystem Chips with Dynamic Reconfig requirement
- UVM limitations due to phasing structure
- Components require heavy configurability
- Paper proposes a work-around :

  Create all required env(s) at start and switch environments by controlling the Virtual Interface connection(s) during reconfiguration

Questions