Interconnect Verification Challenge and the need for a generic scoreboard

François Cerisier
Verification Expert
Test and Verification Solutions France

Verification Futures 2013
SoC Interconnect

- Core 0
- Core 1
- DSP
- DMA
- Codec (h265)
- GPU / 3D Motion
- Cache
- AXI
- AXI
- AXI
- OCP
- AHB
- AHB
- AHB
- AXI
- AXI
- AXI
- AXI
- AXI
- AXI
- OCP
- OPB
- AHB
- User Bus
- User Bus
- Wishbone
- USB
- Display
- Audio HWA
- UART
- DDR
Main Interconnect Characteristics

• Communication of transactions between Masters and Slaves
  – Protocols (AXI, AHB, APB, OCP, PLB, OPB, DCR, Wishbone, company corporate bus, …)
  – Bus Widths (8/16/32/64/128)
• Memory Maps
  – Shared memory map for all masters ?
  – Memory map clusters ?
  – One Memory map per master ?
• Address Space
  – Physical Address Space
  – Virtual Address Space (MMU)
  – System Virtual Memory Address Space (System MMU)
• Cache Coherency
  – ACE, ACE-Lite ?
  – Others ?
• SoC specific features
  – Dynamic re-configuration (address space, security, routes, …)
  – Error management
    → invalid requests, unmapped addresses
  – Security, DRM, Firewall
    → invalid request depending on security attributes, software execution level
  – Power management, Master/Slave power up/down
    → interrupts, invalid request
The Interconnect Verification Plan

- **Address Map**
  - Are all masters able to access all possible slaves?
  - ... under virtual address mode?
  - Errors on invalid addresses

- **Protocol Sanity**
  - Are all kinds of transactions supported on each route?
  - Are bursts/locks supported on each route?
  - Protocol not broken under stress conditions

- **SoC features**
  - Security
    - Can secure transactions access to all slaves?
    - Are unsecured transactions getting errors from secured slaves?
  - Power Management Use cases
    - Are we getting error from power off slaves?
    - Are we able to wake up a slave?
  - Cache Coherency

- **Performance Analysis**
  - Latency, Bandwidth

- **Interconnect integration**
  - Are the different CPU clusters and IPs well connected with the interconnect?

- **Use cases**
Routes & Address Map

Slaves Memory Space

- Slave 5 AXI 0x7000_3FFF
- Slave 4 APB 0x7000_2000
- Slave 4 OCP 0x7000_1FFF
- Slave 4 APB 0x7000_0000
- Slave 3 AHB 0x5FFF_FFFF
- Slave 3 AHB 0x5000_0000
- Slave 2 OCP 0x401F_FFFF
- Slave 2 OCP 0x4000_0000
- Slave 1 OCP 0x2FFF_FFFF
- Slave 1 OCP 0x2000_0000
- Slave 0 AXI 0x1FFF_FFFF (cacheable)
- Slave 0 AXI 0x0000_0000

Master 5 AXI
Master 4 AXI
Master 3 AHB
Master 2 OCP
Master 1 CPU1 - AXI
Master 0 CPU0 - AXI

Coherency Domain

Routes:
- Route M5 to S5
- Route M4 to S5
- Route M4 to S3
- Route M0 to S4
- Route M0 to S2
- Route M1 to S0
- Route M0 to S0
Protocol conversion issues

AXI transaction

Request transfer

 AXI burst len=3
 Size = WORD
 Address = 0x5
 Kind = WRAPPED LOAD

Response transfer

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>E</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
</tbody>
</table>

Possible Converted transaction in protocol X

On a 64 bit bus

Request transfer

 LD16
 Addr = 0

Response transfer

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>E</td>
<td>D</td>
<td>C</td>
<td>B</td>
<td>A</td>
<td>9</td>
<td>8</td>
</tr>
</tbody>
</table>

On a 32 bit bus

Request transfer

 LD16
 Addr = 0

Response transfer

<table>
<thead>
<tr>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>F</td>
<td>E</td>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>17</td>
<td>16</td>
<td>15</td>
<td>14</td>
</tr>
</tbody>
</table>
Testbench Requirements

- **VIPs**
  - For each protocol (master and slaves)
  - With good protocol checkers
  - With full transaction monitoring sent to analysis port

- **Virtual Sequences**
  - Enable controlling all masters from one main sequence
  - Enable complex scenarios

- **NoC / Interconnect Scoreboard**
  - End to End Transaction Checking

- **Standalone or within SoC**
Standalone Testbench or with the SoC

• **Standalone**
  – Interconnect available before integration in the SoC
  – Only verify the interconnect out of context

• **Headless Within Integrated SoC**
  – Allows to find integration issues
  – Requires to black box all IPs
Interconnect TestBench - standalone

SoC Interconnect!

Test Bench

UVM AXI Agent -> AXI
UVM AXI Agent -> AXI
UVM OCP Agent -> OCP
UVM AHB Agent -> AHB

NoC or Interconnect Scoreboard

UVM AXI Agent
Monit or
Driver

UVM AXI Agent
Monit or
Driver

UVM OCP Agent

UVM AHB Agent

OPB
AHB
AXI
User Bus
Wishbone
AHB
AXI

Monit or
Driver

Driver

Driver
Interconnect TestBench in headless SoC context
Interconnect Scoreboard Requirements

- Connect to any bus protocol VIP
- End to End transaction checking
  - Data, direction, attributes, response, atomicity
- Support for:
  - Multiple address maps, Virtual address space
  - Address map reconfiguration, MMU
  - Security, Power management
  - Cache Coherency features (support for ACE)
  - User defined security/filtering (DRM, …)
- Comparison policies
  - Strict:
    - one to one transaction comparison
  - Permissive:
    - Allow transaction address realignment, dummy reads, nops
  - Per checker configuration
    - User switch on/off each checker (per path)
Simple Scoreboard principles – data flow design

Test

sequences

sequencer

driver

monitor

assertions

Bus A VIP Master Agent

DUT

sequences

Bus B VIP Slave Agent

driver

monitor

assertions

trans

Transaction Predictor

Ref trans

Storage

Match ?

Compare / Search

Scoreboard

Response Scoreboard
Interconnect Scoreboard Structure

- **Internal Generic transaction**
  - Extendable with attributes using UVM factory
- **Adapter between VIPs and generic port**
- **One inner simpler scoreboard per route**
- **Reconfigurable Address Map table**
Interconnect Scoreboard Structure

DUT

NoC Scoreboard VIP

Address Map & Configuration

VIP to Scbd adapters

Coherency Domain

AXI Monitor

AXI adapter

Master 0 to S0

Slave I/F 0

AHB adapter

AHB Monitor

UVM Agent

Coherency Domain

AXI Monitor

AXI adapter

Master 1 to S1

Slave I/F 1

AHB adapter

AHB Monitor

UVM Agent

Coherency Domain

AXI Monitor

AXI adapter

Master 2 to S2

Slave I/F 2

AHB adapter

AHB Monitor

UVM Agent

Coherency Domain

AXI Monitor

AXI adapter

Master 3 to S3

Slave I/F 3

AHB adapter

AHB Monitor

UVM Agent

Coherency Domain

AXI Monitor

AXI adapter

Master 4 to S4

Slave I/F 4

AHB adapter

AHB Monitor

UVM Agent

Coherency Domain

AXI Monitor

AXI adapter

Master 5 to S5

Slave I/F 5

AHB adapter

AHB Monitor

UVM Agent

slave

VIP Monitors

VIP to Scbd adapters

NoC

APR

Slave 4
Adding Cache Coherency – ACE Support

- ACE support
  - Requests may not reach the targeted memory
  - Responses may come from another coherent master

→ Breaks the general master/slave scoreboard principles
Interconnect Scoreboard Structure (with ACE)
Scoreboard transaction flow – Request

Master

Monitor (Master AXI)  Monitor (Master ACE)

noc_scbd_adapter  noc_scbd_adapter
(Master AXI)  (Master ACE)

noc_scbd_adapter  noc_scbd_adapter
(master)  (slave)

Monitor (master)  Monitor (slave)

DUT/VIP  TB  NoC_scbd

Slave

Monitor (master)  Monitor (slave)

noc_scbd_adapter  noc_scbd_adapter
(master)  (slave)

noc_scbd_master  noc_scbd_slave  noc_scbd_path

DUT/VIP  TB  NoC_scbd
// instantiate the scoreboard
scbd0 = noc_scbd::type_id::create("scbd0",this);

// create two address domains
scbd0.add_address_domain ("CPU0");
scbd0.add_address_domain ("CPU1");

// address domain 0 configuration
scbd0.add_address_segment("CPU0","","RAM","h0000_0000","h2000000",BUS);
scbd0.add_address_segment("CPU0","","UART","h9000_0000","hFFFF",BUS);
scbd0.add_address_segment("CPU0","","ID0","hFFFF_FFF0",1,BUS);

// address domain 1 configuration
scbd0.add_address_segment("CPU1","","RAM","h0000_0000","h2000000",BUS);

//Masters declaration
scbd0.add_master("CPU0_instr","CPU0");
scbd0.add_master("CPU0_data","CPU0");

//Slaves declaration
scbd0.add_slave("ROM");
scbd0.add_slave("RAM");
scbd0.add_slave("UART");
//Paths declaration
scbd0.add_path("CPU0_data", "ROM");
scbd0.add_path("CPU0_data", "RAM");
scbd0.add_path("CPU0_data", "UART");

// Instantiate Adapaters
noc_scbd_axi_adapter CPU0_instr_adapter;
CPU0_instr_adapter.noc_item.connect(scbd0.masters[0].adapter_export);

// connect monitors to scoreboard adapters
axi_env.agents[0].monitor.collected_request.connect(this.CPU0_instr_adapter.monitor_export_REQ);

//...
Advanced features

• **Protocol conversion to any VIP**
  – thru adaptors

• **Dynamic address map**
  – Thru configuration API

• **Attributes**
  – AXI attributes provided
  – Use the factory to extend the generic type

• **Security & Power Management**
  – Thru user defined methods

• **Functional Coverage**
  – path, address map, security, …

• **Performance**
  – Latency and bandwidth extraction available (requires proper use cases)

• **ACE & Cache Coherency**
  – Internally supported

• **Others**
  – Virtual method place holder can be user extended
User’s experience

• **3 derivatives of a SoC Interconnect**
  – Approx 40 masters, 60 slaves with over 200 paths
  – 5 protocols, 3 different bus sizes
  – Security Management
  – Power Management features
  – Dynamic address translations

• **Scoreboard Developments**
  – Right architecture choice is key
  – Generic features / Generic Adapters
  – Search and comparison algorithms

• **Verification results**
  – Address map specification
  – Wrong protocol translations of AXI FIXED from 64 to 32 bit buses
  – Deadlock in some traffic congestions involving bursts
  – Deadlock in power management
Interconnect Verification Summary

• SoC Interconnect needs to be verified from end to end

• Verification Environment should address
  – Complex scenarios
  – Stress/congestion conditions

• Interconnect SoC scoreboard should be generic & highly configurable
Generic Scoreboard VIP summary

• **Allows quick testbench setup**
  – fully configured in minutes

• **Compatible with any VIPs**
  – Only requires proper monitor using uvm_analysis_port

• **Allow verification of security features, power management, ACE/Cache Coherency and more user defined features**
• Questions?