Leveraging IP-XACT Standardized IP Interfaces for Rapid IP Integration

David Murray, CTO, Duolog Technologies
Overview

• What is Interface Standardization
• Why Standardize?
• Leveraging Standards
• IP-XACT as an interface standard
• Mechanics
• Flows
• Summary
WHAT IS IP INTERFACE STANDARDIZATION?

• Standardization of
  – HW Interface
    • Ports
  – SW Interface
    • Memory Maps
    • Registers
Why Standardize

‘INTEROPERABILITY’

IP Interface Standards [HW + HW/SW]

IP Providers

EDA Industry

IP Consumers
Leveraging Standards

• Leveraging interface standards for IP Creators
  – Standards help utilise automation to create IP Collateral e.g. documentation, design implementations or verification environments.

• Leveraging standards for IP Consumers
  – Standards help to automate IP integration and verification. IP blocks with known good interfaces are easier to integrate both from the hardware and software side.

• Leveraging standards by EDA industry
  – Standards are not enough! They must be automatable and automated!
  – EDA Industry delivers value by providing automation of complex design and verification flows,
IP-XACT Standard

IP-XACT

• IP-XACT is an XML schema that defines and describes electronic components and their designs.

• Goals
  – to ensure delivery of compatible component descriptions from multiple component vendors
  – to enable exchanging complex component libraries between EDA tools for SoC design
  – to describe configurable components using metadata
  – to enable the provision of EDA vendor-neutral scripts for component creation and configuration
IP-XACT Content

IP Metadata
- Components
- Ports
- Bus Interfaces/Definitions
- Filesets/files
- Memorymaps/registers
- ComponentGenerators

SoC/Sub-system

Design/Integration Metadata
- Component Instances
- Interface connections
- Port-level connections
- Design Configurations

Generators/Flows
- Component Generators
- Generator Chains
Bus Definitions

Defines how bus ports should appear on interfaces

<table>
<thead>
<tr>
<th>Port Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
</tr>
<tr>
<td>Description</td>
</tr>
<tr>
<td>Type</td>
</tr>
<tr>
<td>Default</td>
</tr>
<tr>
<td>Qualifier</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On Master</th>
<th>Presence</th>
<th>Direction</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>Required</td>
<td>out</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On Slave</th>
<th>Presence</th>
<th>Direction</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>Required</td>
<td>In</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>On System</th>
<th>Presence</th>
<th>Direction</th>
<th>Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>Presence</td>
<td>illegal</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Bus Abstraction

- Defines the main characteristics of an overall bus and how it should appear on an IP

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Type</th>
<th>Default</th>
<th>Presence</th>
<th>Direction</th>
<th>Width</th>
<th>On System</th>
<th>On Slave</th>
<th>On Master</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR</td>
<td>Address</td>
<td>Wire</td>
<td>0</td>
<td>Illegal</td>
<td></td>
<td></td>
<td>Required in</td>
<td>Required out</td>
<td>Required out</td>
</tr>
<tr>
<td>WDATA</td>
<td>WriteData</td>
<td>Wire</td>
<td>0</td>
<td>Illegal</td>
<td></td>
<td></td>
<td>Required in</td>
<td>Required in 32</td>
<td>Required out 32</td>
</tr>
<tr>
<td>RDATA</td>
<td>ReadData</td>
<td>Wire</td>
<td>0</td>
<td>Illegal</td>
<td></td>
<td></td>
<td>Required out</td>
<td>Required in 32</td>
<td>Required in 1</td>
</tr>
<tr>
<td>WR_N</td>
<td>Write Enable</td>
<td>Wire</td>
<td>1</td>
<td>Illegal</td>
<td></td>
<td></td>
<td>Required out</td>
<td>Required out 1</td>
<td>Required in 1</td>
</tr>
<tr>
<td>ERR</td>
<td>Error</td>
<td>Wire</td>
<td>0</td>
<td>Illegal</td>
<td></td>
<td></td>
<td>Optional out</td>
<td>Optional in 1</td>
<td>Required in 1</td>
</tr>
</tbody>
</table>
Mapping Ports to Interfaces

COMPONENT

Model

PORTS

- m0_addr[11:0]
- m0_wrdata[31:0]
- m0_read_data[31:0]
- m0_wren_l

Bus Interface

- Bus Type
- Abstraction Type

Logical Ports

- ADR
- WDATA
- RDATA
- WR_N
- ERR

Abstraction Definition

Physical Ports

- m0_addr[11:0]
- m0_wrdata[31:0]
- m0_read_data[31:0]
- m0_wren_l

Mapping Ports to Interfaces
• Defines Memory maps and registers in an standard format
Interconnects/Bridges

• Defines Bus interconnect structures and bridges
Defining Connected Systems

- Interface on the component boundary
- Hierarchical Connection
- Interconnection
- ComponentInstance
- adHoc connection
Utilizing the Standard
IP Qualification

- **Bus Definitions**
  - IP-XACT
  - XML

- **IP-XACT Creation**
- **IP Qualification**

**Specification Checks**
- Syntax, SCR + Custom

**RTL Coherency Checks**
- HW/SW Interface verification
- HW Interface

**IP Integration Process**
- ARM
- DUOLOG
- CADENCE
- JASPER
IP Integration and Verification

- ARM
  - Bus Definitions
    - ARM IP
      - (e.g. NIC400, CN504)
    - IP-XACT
    - XML
  - Other IP/Bus definitions
    - (In IP-XACT Format)

- IP Integration Creation
- Performance Verification
  - CADENCE Interconnect Workbench
- DUOLOG Socrates
  - IP-XACT
  - XML
  - RTL
Summary

• The growth of 3rd party IP usage and the lack of hardware and Hardware/Software interface standardization are causing a lot of integration issues which remain a key challenge of SoC design

• IP-XACT has emerged as a clear leader for the IP interface specification standards
  – It has comprehensive capabilities when it comes to defining the intricacies of RTL port interfaces
  – Possible Hardware/Software elements such as registers and memory maps.

• The standardization of IP interface helps
  – To improve IP quality and automation
  – To streamline IP integration and verification

• IP-XACT adoption in the industry has led to high-value multi-vendor, interoperable flows that provide integration ready-IP, accelerate IP integration and enable high levels of verification automation.
Thanks our industry partners who have collaborated with us in the past number of years to make these highly interoperable IP-XACT Flows.

ARM : Paul Martin and William Orme
Jasper : Norris Ip and Shaun Giebel
Cadence: Nick Heaton and Adam Sherer
Questions!
Thank You!