Interconnect Design in SoC

Within the increasing complexity of SoC design, bus-interconnect is a key component which has led to evolution in the design of interconnect with a new socket-based approach. The socket is defined as: the decoupling of IP core and interconnect functionality. The socket-based approach provides interconnect IP to reuse without rework and it provides a bus-independent interface. This allows on-chip interconnect to provide application-specific features. The socket also isolates the cores from the internal switching logic and provides the following advantages:

- IP core decoupling with protocol conversion
- Command translation
- Clock domain crossing
- Width conversion
- Security management features for protecting a specific memory region from different cores (i.e., protecting Instruction Fetch area from Write commands)
- Configuration-based error handling support
In socket-based design approach, interconnect interfaces are very generic and developed based on protocol configuration as per specific system requirements. Standalone interconnect verification with 3rd-party IP test suite and verification environment is not sufficient and does not cover all main features as per system configuration. Hence one should check whether interconnect sockets are generated based on proper configuration. Apart from interconnect configuration checks, the interconnect needs to be checked for performance, error handling and security management which we need to address while verifying complete SoC.

Reduced time-to-market offers a very narrow time frame for thorough interconnect verification on SoC. To address this challenge, we have developed a configurable verification environment and coverage model (to check the quality of verification performed), which can also be reused in different SoC verifications.

**Verification Environment**

The verification environment for interconnect fabric requires VIP monitor integration to monitor transactions on different bus interfaces. In a bench test, VIP monitors are connected on each Socket interface of interconnect to collect transactions. To boost the verification time, configurable support is provided to plug VIPs on socket interface to drive (along with monitor) the stimulus directly on interconnect and bypass the actual bus master(s).

With different protocol sockets (like OCP, AXI, APB, AHB) are present on interconnect, a protocol independent scoreboard has been developed to check data integrity and verify different data path transactions to and from the bus masters/slaves connected to the interconnect fabric. Figure 2
explains the generic scoreboard architecture and connection with Interconnect fabric.

The main features of the implemented protocol independent interconnect scoreboard are:

- Request and Response routing as per connectivity configuration
- Address and Data width translations with control signals mapping
- Transaction splitting and merging from master to slave based on type of transaction and master slave configuration
- Separate data checkers for request and response data path
- In-order and Out-of-order request and response checking
- Security management support
- Response checking on reserved address space
- Configurable optional support for performance measurement and analysis
- Functional coverage metrics on routes, memory segments, transaction kinds, responses and cross coverage metrics (can also include IP protocol coverage)

The implementation of an interconnect scoreboard needs to consider each possible route of the Interconnect fabric in SoC and the address map. Routing and address map configuration is therefore a key feature of a interconnect scoreboard.

![Scoreboard Architecture for Interconnect](image)

Due to different bus interfaces at input & output of interconnect fabric, protocol conversion (PRC) has been done at each VIP monitor connected to interface, based on master and slave configuration. In protocol conversion, transactions are mapped as per slave interface configuration, to reduce the complexity in response path. Because of diverse addresses and data width of each master and slave, transaction realignment between master and slave interfaces needs to be taken care during protocol conversion with splitting of requests and merging of responses.

In SoC, multiple masters can communicate to multiple slaves and socket protocols introduce two-way communication. We have developed separate scoreboards for request and response per slave. The Scoreboard is kind of a data checker, which checks data integrity with all attributes present on slave interface.

Requests and responses, collected from multiple bus master VIP monitors, are routed through transaction router to the scoreboard based on slave address and connectivity table. The transaction router also has an error reporting mechanism, when transactions are not sent to the correct interface or when transactions, generated to an unmapped address, do not get an error response
from interconnect fabric. It filters transactions when transactions are sent to protected area, and checks the response of those filtered transactions from interconnect. Transaction filtering is done based on protection register modelling in transaction router. These registers are continuously updated in transaction router whenever write transactions are successfully completed as per the connectivity table. Also, the functional coverage model is implemented for features to be verified and requires information samples from the transaction router.

Continued in Part 2.

Authors Janak Patel & Bhavin Patel are with eInfochips.

Author Ramneek Real is with Toshiba.