Infineon Inhouse IP Connectivity Platform

Anand Patil, Infineon Technologies India Pvt Ltd, Bangalore, India (anand.patil@infineon.com)

Maheshkumar Simpy, Infineon Technologies India Pvt Ltd, Bangalore, India (maheshkumar.simpy@infineon.com)

Leily Zafari, Infineon Technologies AG Neubiberg, Germany (leily.zafari@infineon.com)

Michael Velten, Infineon Technologies AG Neubiberg, Germany (michael.velten@infineon.com)

Erkan Akbay, Infineon Technologies AG Neubiberg, Germany (erkan.akbay@infineon.com)

***Abstract*— In today's SoC design there is an increasing demand to integrate more and more functionality resulting in huge functional blocks termed as IPs. The wiring of these functional blocks through the hierarchical layers of a design is a tedious and error prone task. These interconnections are concurrently done by concept and design engineers, which might result in an inconsistent connectivity specification. In addition, errors in signal naming conventions, incorrectly mapped bit positions and many other fault conditions produce seemingly simple errors that can waste hours of debugging time during SoC verification. It might also result in an expensive device re-spin if these errors are missed completely during the verification process.**

In order to accommodate the complexity and also to prevent slowdown of the development process, an Infineon in-house SoC connectivity platform was developed which is introduced in this paper.

Keywords— ***IP, SoC, Connectivity, Platform, Database, Web Frontend, HTML***

I. Introduction

Infineon started with specifying the connectivity directly inside the Hardware Description Language (HDL) [3]. Later, domain specific languages (e.g. Ganove) were used for specifying the connectivity. The limitation of both approaches is that they do not provide the possibility of concurrent connectivity entry for multiple users distributed at multiple sites.

This paper mainly focuses on providing a common connectivity platform which enables multiple users with different roles/views (e.g. concept and design) to specify interconnection of IPs of a SoC. The complexity of a SoC is handled using divide and conquer mechanisms applied across the views. Furthermore, it provides a flexibility to interconnect IPs concurrently in real time at multiple sites. It also triggers semantic checks during connectivity specification, thus providing debugging options to the user.

II. Objective

The main objective is to provide a consistent concept and design solution for SoC connectivity specification to reduce complexity using divide and conquer mechanisms applied to all SoC architectures at Infineon. Furthermore, a single source connectivity database shall provide consistency between documentation and design implementation.

III. Overview

The connectivity platform provides a web-interface as frontend for the users to enter the connectivity specification. This web-frontend communicates to a central single source database which stores information about the Bill-Of-Material (BOM) and connectivity information of a design project.

Furthermore, a user management is provided which handles access permissions and enables the multi user/role support by assigning users to their corresponding products and views. Hence, if a concept engineer logs into the system, he will see a flat top-down view (Fig 1) on the complete SoC.

Design engineers have a more bottom-up view (Fig 2) on the SoC, because divide and conquer mechanisms can be applied to partition a complex SoC design into several sub-systems. Hence, a sub-system owner will only see the IP instances of his sub-system which are his responsibility to interconnect. The sub-system owner can also control the interfaces/ports [2] of the boundary of his sub-system. The SoC architect will only see the top-level IPs and sub-system boundaries which he needs to interconnect.

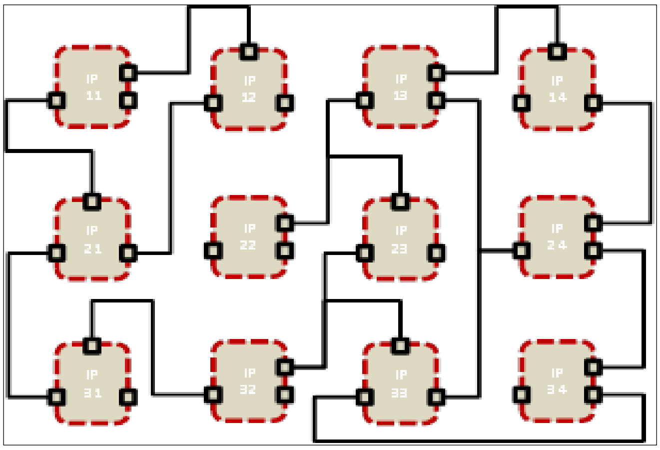

Fig 1.Concept View

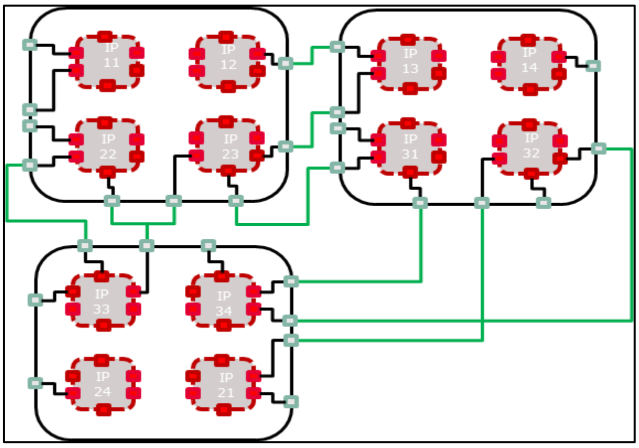

Fig 2.Design View

The special feature about this connectivity platform is that, the described top-down and bottom-up views are combined in one database using different connectivity layers. This has several advantages:

- Connectivity specified by concept engineers is directly visible in the design view and vice versa
- Concurrent connectivity specification and debugging for a multi-user/site environment
- Conflicts between design and concept specification directly result in change requests
- Incompatible interface items cannot be interconnected with each other
- SoC can be repartitioned to a different sub-system structure without losing connectivity

Furthermore, it supports batch mode connectivity upload and download using the Ganove SIG format as textual scripting interface. The downloaded connectivity is further processed to generate consistent documentation and implementation of the design.

IV METHOD STEP BY STEP

The entire method to realize this connectivity platform may be broken down into 2 steps

A. Meta-Model Design and SQL API Generation

B. Server Client Interaction

Each of these steps is discussed subsequently.

A. Meta-Model Design and SQL API Generation

Modern application development processes always start with defining a model (Meta-Model here!) of the application before implementation. This is in particular interesting if the application is being developed by several people since a layer of abstraction is required due to the fact that everyone is not able to know all the aspects of the tool development.

The Meta-Model [1] gives us a common base of development which greatly increases the coding efficiency. The Connectivity platform design starts with modeling the Meta-Model in a UML class diagram. From the designed Meta-Model, the SQL API's are generated. So defining a class in the Meta-Model leads to the generation of a Python class in the SQL API's, this then maps onto a database table as shown in the Fig 3.

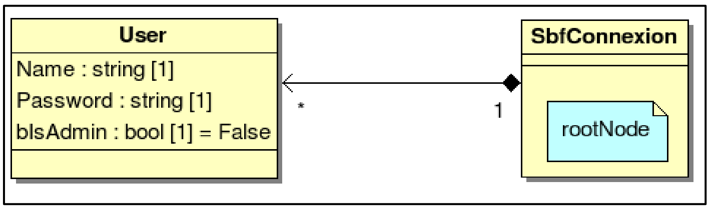

Fig 3. Meta-Model Design and SQL code generation

B. Server Client Interaction

As described in the earlier sections, multisite connectivity specification is one of the critical requirements of this connectivity platform. Synchronizing the connectivity specification with concurrent developers is a challenge. So a server client setup will solve the synchronization problem which is explained in the following paragraphs.

At the backend, the web server is running the Pyramid [4] application which interacts with the MYSQL database using the SQLAlchemy API's generated from the Meta-Model. The Pyramid framework will also interact with the frontend (Web browser) in HTTP/JSON format. All static assets which are included in the source code file (CSS, JS, fonts, pictures) are served by an Apache server as shown in fig 4.

The communication between server and client is done using AJAX (Asynchronous JavaScript and XML). This helps in dynamically updating a web page without reloading the entire page but only updating certain aspects. The Pyramid framework also helps in maintaining the user sessions and the authentication [5] of users is done using LDAP (Light Weight Directory Access Protocol).

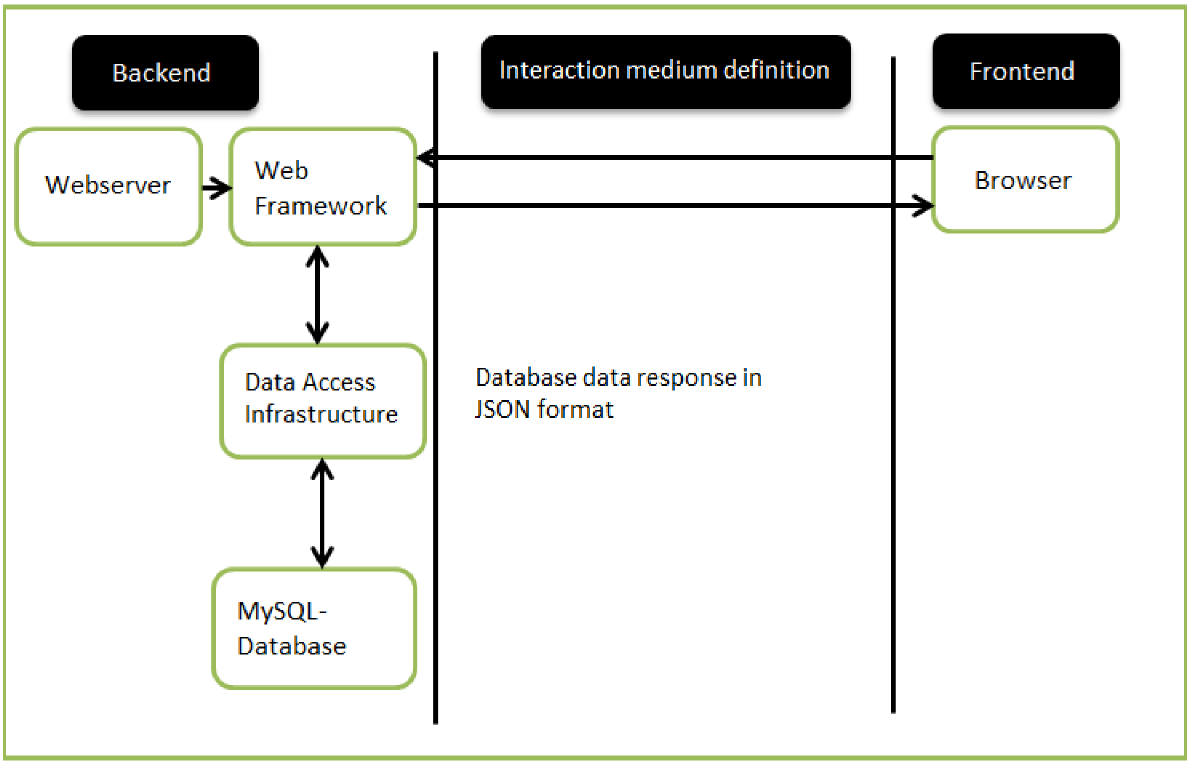

Fig 4. Server Client Interaction

V SUPPORTED FEATURES

A. User management

The user management is of the utmost importance and was defined as a primary requirement. The design process of a chip is handled by many engineers with different responsibilities starting with the concept engineers specifying the requirements, down to the actual hardware implementation. User management system is advantageous for a design automation platform, which will be concurrently used by many engineers. In order to assure accountability and division of responsibilities, not all resources should be available for everyone. Also the user management assures that an engineer is only displayed as much information as needed to limit the flow of information and focus on his/her tasks.

This is achieved by using authentication [6] and authorization techniques applied to users based on his responsibilities. One or more users are provided with admin rights to manage the authorization of other users. Also the admin is responsible for defining or creating a new product and assign users to access the required IP/Subsystem of the added product.

B. Multiple product support

The connectivity platform provides support and management of more than one product in a single database. This also includes managing different steps of a product.

C. Import/export

Since Infineon develops entire microcontrollers and not only part solutions, this platform needs to be embedded in a flow which generates RTL at the end. The platform is capable of importing hierarchy information to supply the engineer with a proper representation of the chip and to provide him with interface items for connectivity. Additionally old connectivity information has to be handled also for developing new products based on existing ones. That finalizes the crude migration and data import requirements. The other input of this tool is handled by the core interface, the web application. The output is restricted (apart from backups) to a signal list output, returning all connectivity information and exports from subsystems of the different connectivity layers.

D. Multi-site

Infineon has many different office locations spread all over the world. The multi-site requirement is specified as the need to access this tool in different places, time zones and environments. This problem is solved by using a web interface accessing a central MySQL database which is also a requirement deduced from the multi-site need.

E. Multi-role

At Infineon we have different types of engineers with different responsibilities. The most common segmentation into the biggest groups is to talk about concept, design, layout, and test engineers. The concept engineers are responsible for specifying requirements for products and functional requirements of IP’s. Design engineers are responsible for implementing the specification provided by the concept engineers (and layout). Previously one design engineer who was in charge of designing a specific IP got the specifications from the other departments by email, calls etc. This is solved by using our platform as connectivity specified by concept engineers is directly visible in design view and vice versa.

F. Connectivity layers

This multi-role requirement implies that the different departments can insert their specifications and/or connectivity independently. This again requires a layer system that connections can be inserted on different levels since the design development status might not match the specification status and vice versa. The engineer should be given the opportunity to work asynchronously which removes all the dead-lock issues one might face if working on a multi user environment with different purposes.

G. Connectivity report

This connectivity platform enables the user to check the percentage of connectivity of an IP instance in the form of a report. This report provides the connectivity information of each port of an IP. This is very helpful for the engineers to get an overview of the complete connectivity.

H. Connectivty specification

The most important feature of this connectivity platform is to specify the connectivity for interface/port items. This is realized by selecting the interface/port items of IP’s to be connected. Connectivity specification includes broadcasting of output pins, binding the input pins to 0/1 tie value in the connectivity page. The connected interface/port items are colored green and driver/receiver of the connected pins can be visualized by clicking on the corresponding pin.

VI RESULTS

A proof of concept implementation of this connectivity platform was already used for the development Infineon Automotive microcontrollers. Using this approach, 100% connectivity coverage was achieved. Because of this positive result, the full featured version of connectivity platform (presented in this paper), was developed using modern development frameworks based on Meta-Modelling methodology. This next generation connectivity platform is rolled out for the Infineon Automotive Microcontroller product and is in use. Furthermore, a pilot phase is executed for Infineon's Power Microcontroller.

VII CONCLUSION

The approach presented in this paper has been tried and tested with products from Infineon Automotive and Power Microcontroller. So far, the results are very positive. The approach increases the efficiency of the user as the time taken to specify the connectivity information against traditional effort is reduced from days to minutes. This approach also eliminates any errors caused by traditional effort as there are inbuilt semantic checks incorporated inside the framework.

The pilot project for this technology was done using Infineon Automotive microcontrollers. But the developed infrastructure can also be easily extended to support products of other Business units for specifying connectivity information.

VIII FUTURE ENHANCEMENT

This paper demonstrates an approach for handling connectivity of IPs using Server client infrastructure with Web front end as interface. In future, the presented connectivity platform can be further extended to support features like mutli derivative handling, notification handling for requesting unavailable connectivity information on different layers, direct interaction with configuration managements systems etc...

IX REFERENCES

[1] OMG. (2014, Feb.) Omg's Meta objects facility @ONLINE. [Online]. Available:
[2] http://www.omg.org/mof/
[3] IEEE. (2014) IEEE standard for ip-xact @ONLINE. [Online]. Available: https://standards.ieee.org/findstds/standard/1685-2014.html
IEEE (2008) VHDL Reference Manual @ONLINE. Available:
[4] http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4772740&filter=AND(p_Publication_Number:4772738)
[5] Pyramid Framework @ONLINE Available: http://docs.pylonsproject.org/en/latest/docs/pyramid.html#pyramid-documentation
[6] Pylon. (2015 Mar.) Authentication policies @ONLINE. [Online]. Available:
http://docs.pylonsproject.org/projects/pyramid/en/stable/api/authentication.html