Nano Scale Disruptive Silicon-Plasmonic Platform for Chip-to-Chip Interconnection

DDCM with electrical PHY design and verification

Milestone no.:	MS28
Due date:	10/31/2012
Actual Submission date:	10/31/2012
Authors:	ST
Work package(s):	WP5
Distribution level:	CO[1] (NAVOLCHI Consortium)
Nature:	Document, available online in the restricted area of the NAVOLCHI webpage

List of Partners concerned

Partner number	Partner name	Partner short name	Country	Date enter project	Date exit project
1	Karlsruher Institut für Technologie	KIT	Germany	M1	M36
2	INTERUNIVERSITAIR MICRO-ELECTRONICA CENTRUM VZW	IMCV	Belgium	M1	M36
3	TECHNISCHE UNIVERSITEIT EINDHOVEN	TU/e	Netherlands	M1	M36
4	RESEARCH AND EDUCATION LABORATORY IN INFORMATION TECHNOLOGIES	AIT	Greece	M1	M36
5	UNIVERSITAT DE VALENCIA	UVEG	Spain	M1	M36
6	STMICROELECTRONICS SRL	ST	Italy	M1	M36
7	UNIVERSITEIT GENT	UGent	Belgium	M1	M36

[1] **PU** = Public
PP = Restricted to other programme participants (including the Commission Services)
RE = Restricted to a group specified by the consortium (including the Commission Services)
CO = Confidential, only for members of the consortium (including the Commission Services)

Milestone Responsible

Organization: STMicroelectronics
Contact Person: Alberto Scandurra
Address: Stradale Primosole, 50 – 95121 Catania
Italy
Phone: +39 095 740 4432
Fax: +39 095 740 4008
E-mail: alberto.scandurra@st.com

Executive Summary

This document describes the activity carried out by ST to design and verify the Dual Die Communication Module (DDCM).

Change Records

Version	Date	Changes	Author
0.1 (draft)	2012-11-19	Start	Alberto Scandurra
1 (submission)	2012-11-19	Final version	Alberto Scandurra

1. *Contents*

1. *CONTENTS* **3**

1. INTRODUCTION **4**

2. DDCM DESIGN **6**

1. DDCM design flow 6
2. DDCM data base 6

3. CONCLUSION **8**

1. Introduction

The **Dual Die Communication Module** (abbreviated **DDCM**) is the building-block responsible for the interconnection of different dice within a so called Network in Package (NiP), the communication system enabling inter dice data transmission in the context of Systems in Package (SiP) technology.

According to a widely used approach, the DDCM is seen composed of two main building blocks:

- the DDCM **controller**, responsible for managing incoming/outgoing STNoC/SBus/AMBA-AXI traffic, generating IDN segments through encapsulation and preparing them to be sent to the PHY transmitter, as well as collecting them from the PHY receiver;
- the DDCM **PHY**, responsible for transmitting output phyts across the physical link and collecting inputs phyts from the physical link.

As shown in figure 1.1, the DDCM top level in each die consists of a transmitter (DDCM Tx) and a receiver (DDCM Rx).

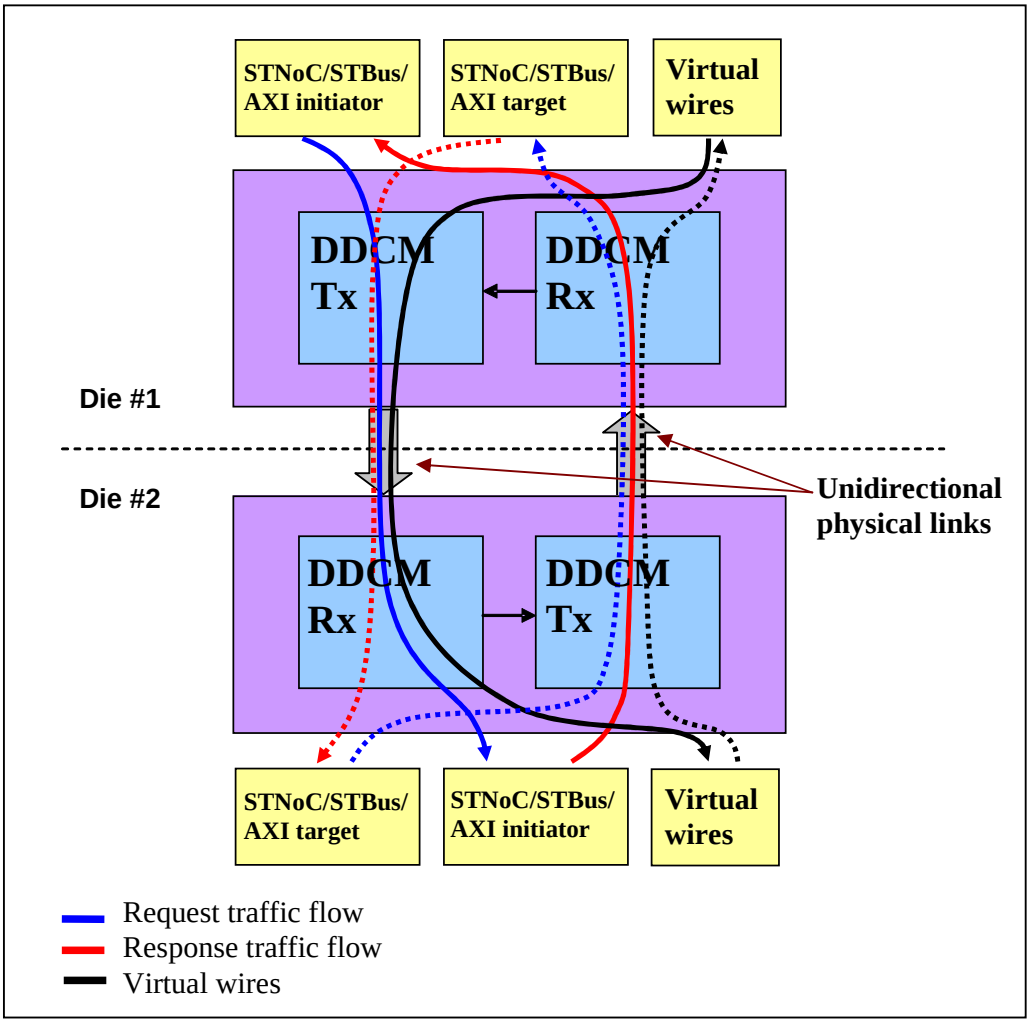

Figure 1-1: DDCM top level architecture and information flow

In such a figure it's possible to see the two information flows supported by a complete DDCM architecture, i.e.

- requests from STNoC/STBus/AMBA-AXI initiators in chip 1 to STNoC/STBus/AMBA-AXI targets in chip 2, responses from STNoC/STBus/AMBA-AXI targets in chip 2 to STNoC/STBus/AMBA-AXI initiators in chip 1, virtual wires from chip 1 to chip 2 (continuous lines);
- requests from STNoC/STBus/AMBA-AXI initiators in chip 2 to STNoC/STBus/AMBA-AXI targets in chip 1, responses from STNoC/STBus/AMBA-AXI targets in chip 1 to STNoC/STBus/AMBA-AXI initiators in chip 2, virtual wires from chip 2 to chip 1 (dotted lines).

Figure 1-2 shows a full architectural view of an DDCM, highlighting the separation between an DDCM transmitter and an DDCM receiver.

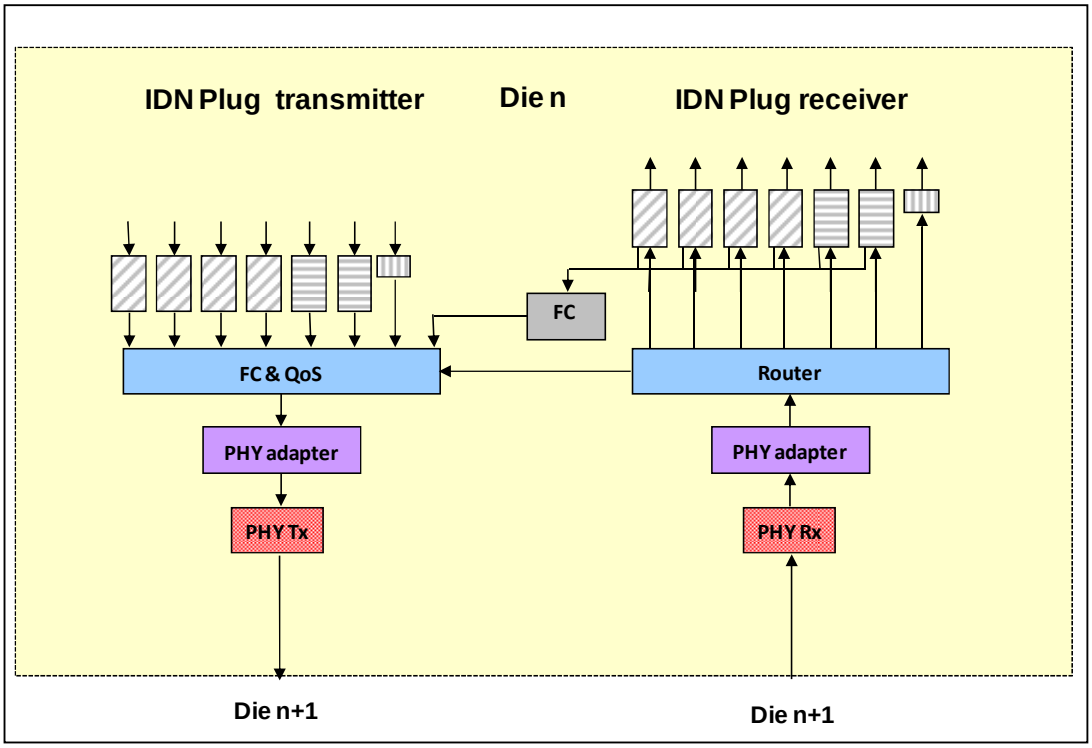

Figure 1-2: DDCM detailed architecture

The DDCM is a **parametric** design that, depending on the SoC where it is used, can be configured properly in order to meet system requirements and needs in terms of interfaces, FIFOs sizes, clock domains synchronization and functionality.

The next section describes the design flow used to implement the block.

2. DDCM design

The DDCM block has been designed following a typical design flow usually adopted for digital modules.

1. DDCM design flow

The design flow of the DDCM block consists of the steps described in the following.

- **Rtl encoding**: the structure and the functionality of the DDCM top level and all its building-blocks is described at rtl (registers transfer level) level in VHDL (Very High Speed Integrated Circuits Hardware Description Language) language. The VHDL code is parametric, meaning that the design can be properly configured according to specific needs so to obtain different structure and functionality depending on different systems requirements.
- **CoreKit generation**: the generic VHDL code is packaged into an entity called coreKit, allowing the DDCM user to generate a specific view of the DDCM by assigning the proper values to design parameters, with the support of a graphical user interface (GUI).
- **Functional verification**: different configurations of the DDCM are stimulated by means of proper traffic generators to check the behaviour is correct according to functional specifications (deliverable D5.1).
- **Logic synthesys**: once the correct functionality is got by simulations, the VHDL code is translated into a circuit, or logic netlist, exploiting a proper CMOS standard cells library.
- **Netlist verification**: the netlist implementing the DDCM is simulated with the same tests used for VHDL verification. All the specified tests have to pass in both cases in order to ensure a) correct functionality and b) consistency between VHDL description and circuit netlist.

2. DDCM data base

The DDCM data base is located in the ST Interconnect System Group server design area and is described in the document related to deliverable D5.2.
In summary the data base, seen as a library located in the directory **ddcm_lib,** is organized in the following two subdirectories:

- **dev** (development) containing the generic design and the generic verification environment;
- **run** containing the simulation area for a set of specific configurations of the design.

In turn the directory dev contains the following subdirectories:

- **doc** containing the functional specification of the block (deliverable D5.1 in NAVOLCHI project context);
- **rtl_vhdl** containing the VHDL files representing the rtl description of the DDCM top level and all its building-blocks;
- **corekit** containing the generic view of the DDCM;
- **verif_env** containing the generic verification environment, i.e. testbench and stimuli sources;
- **verif_run** containing the tests to verify the different functionality of the DDCM;
- **synth** containing the area for the logic synthesis of the DDCM.

For any details about the DDCM data base refer to the D5.2 deliverable document.

3. Conclusion

This document describes the DDCM design and verification flow.