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

Digital domain to plasmonic domain interface specification and VHDL modelling

Milestone no.:	MS5
Due date:	07/31/2013
Actual Submission date:	09/20/2013
Authors:	ST
Work package(s):	WP5
Distribution level:	RE[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	IMEC	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)

Deliverable 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 in detail the microarchitecture and the function of the *PHY adapter* of the Dual Die Communication Module (DDCM) allowing chip-to-chip interconnection exploiting plasmonic devices as components of the physical layer (PHY), as well as the design flow employed to implement the overall subsystem.

Change Records

Version	Date	Changes	Author
0.1 (draft)	2013-07-29	First version	Alberto Scandurra
1 (submission)	2013-10-04	Final version	Alberto Scandurra, Valentina Cernuto

1. *Contents*

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).

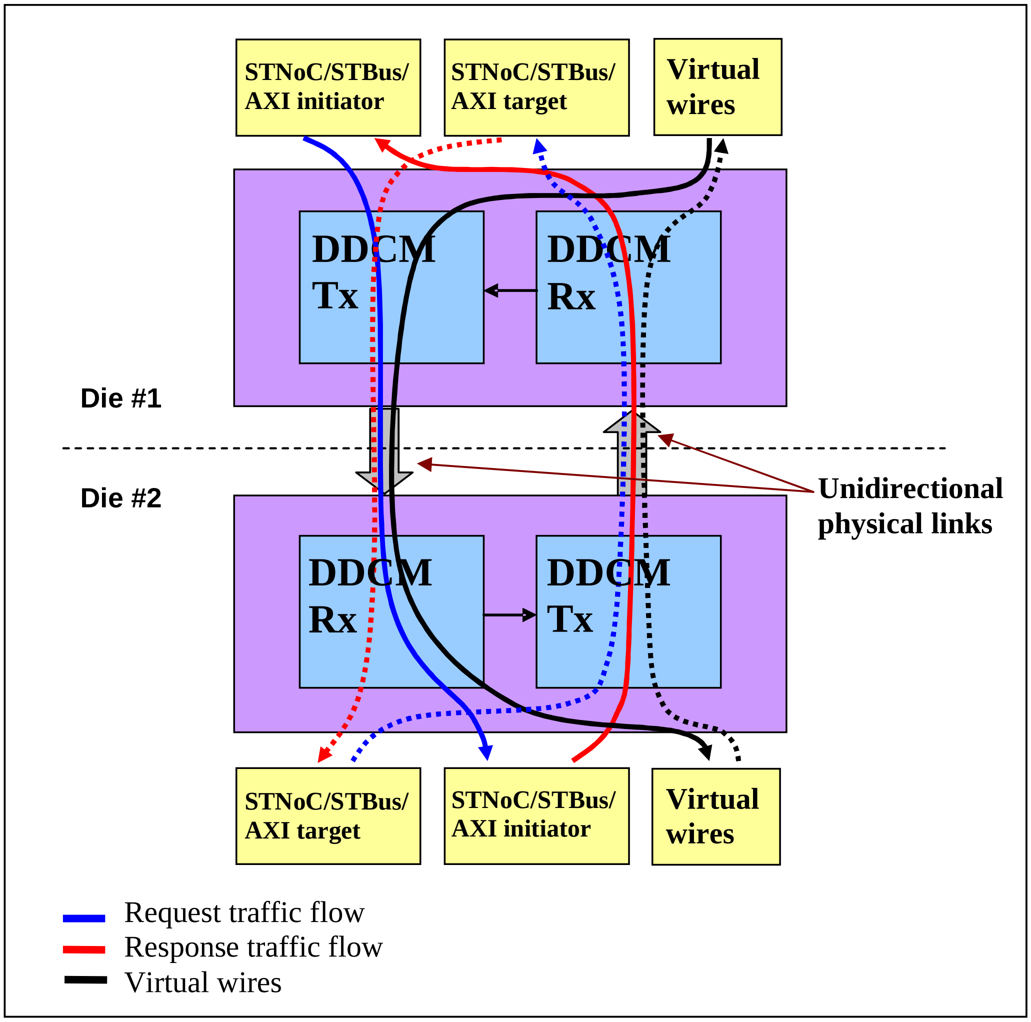

Figure 1-1: DDCM top level architecture and information flow

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 a DDCM, also called IDN Plug (i.e. Inter Dice Network Plug), highlighting the separation between an DDCM transmitter and an DDCM receiver.

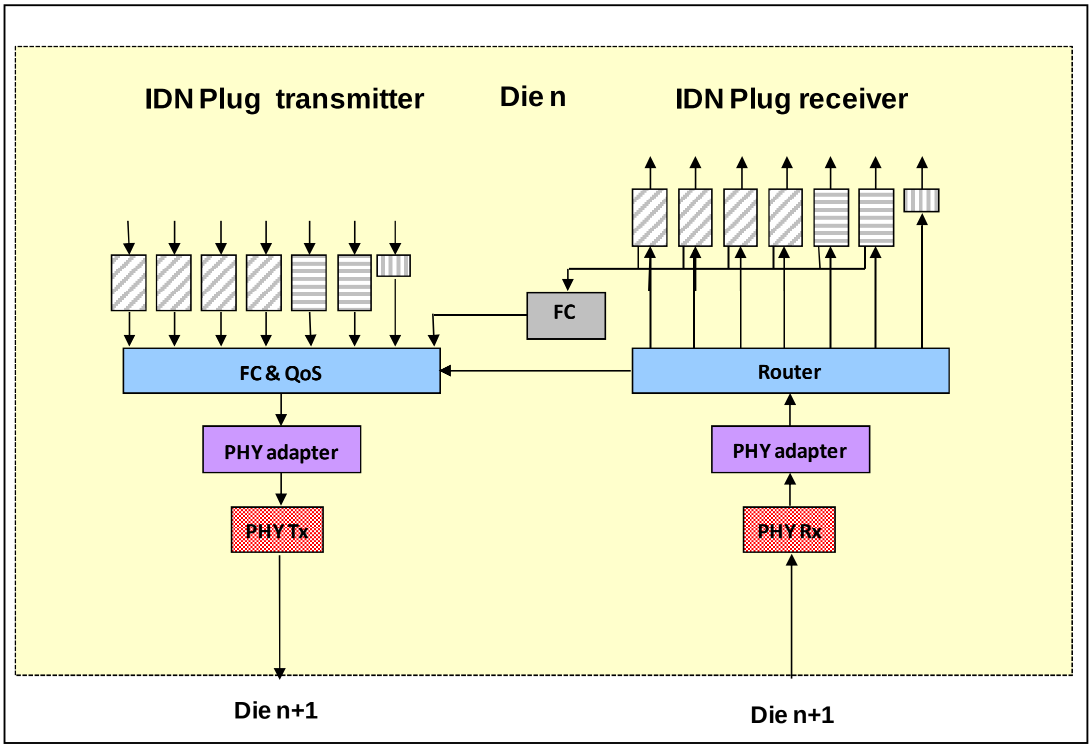

Figure 1-2: DDCM detailed architecture

The next section describes the functionality and the microarchitecture of the PHY adapter intended to exploit the plasmonic devices as physical layer (PHY).

2. DDCM PHY adapter

The DDCM PHY adapter is responsible for adapting the System on Chip (SoC) traffic in such a way to be transferred between the two chips of the system across the physical channel.

Transmitter

The PHY adapter transmitter is responsible for transforming digital data into a format used to modulate the output of the plasmonic LASERs in order to transmit the data across the physical channel (plasmonic waveguide).
Figure 2-1 shows the microarchitecture of the DDCM PHY adapter transmitter specified for the NAVOLCHI demonstrator.

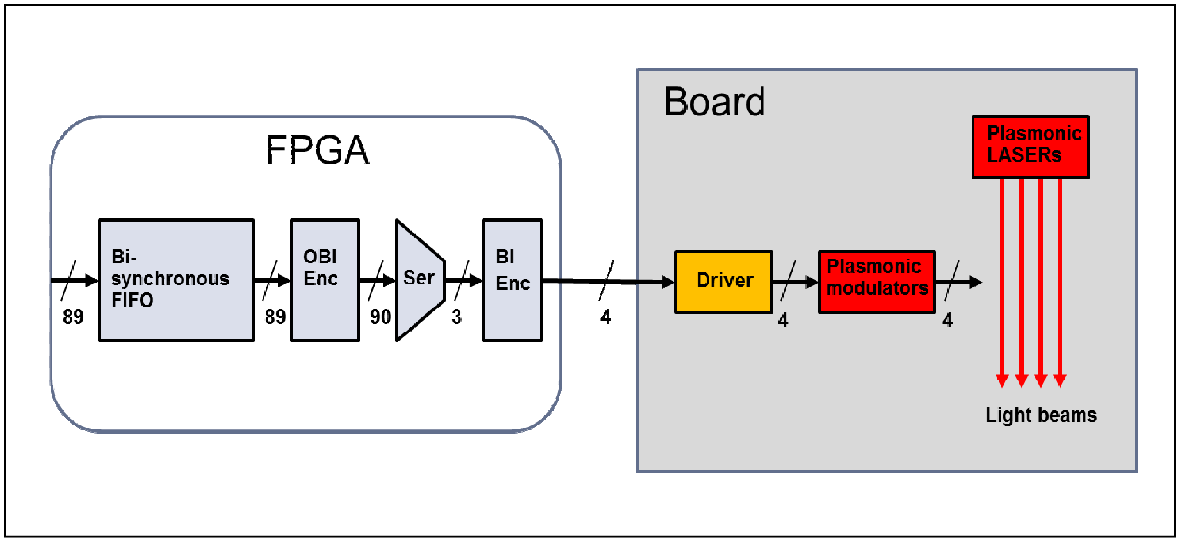

Figure 2-1: Demonstrator DDCM PHY adapter transmitter microarchitecture

The NAVOLCHI demonstrator DDCM PHY adapter transmitter is composed of the following building-blocks:

- a **bi-synchronous FIFO**, playing the twofold role of retiming stage and storage element;
- a data encoder for the minimization of the number of ‘1‘ to be transmitted within a data word, in order to keep turned on the minimum number of plasmonic emitters (**Optical Bus Inverter Encoder**);
- a **serializer** to transmit the N-bits data as chunks of M-bits, exploiting the M plasmonic emitters; in the system used for the demonstrator, the input and the output of the serializer are 90 and 4 bits wide respectively.
- a **data encoder** for the minimization of the Hamming distance between two back-to-back data words, in order to minimize the switching activity of plasmonic emitters (**Bus Inverter Encoder**);
- a set of M **modulator drivers**, each generating the proper control signals for plasmonic LASER modulators;
- a set of M **modulators**, each modulating the output of the related plasmonic LASER.

Receiver

The PHY adapter receiver is responsible for the transformation of the plasmonic information got from the physical channel (plasmonic waveguide) into a format suitable to be used by the digital parts of the SoC.
Figure 2-2 shows the microarchitecture of the DDCM PHY adapter receiver specified for the NAVOLCHI demonstrator.

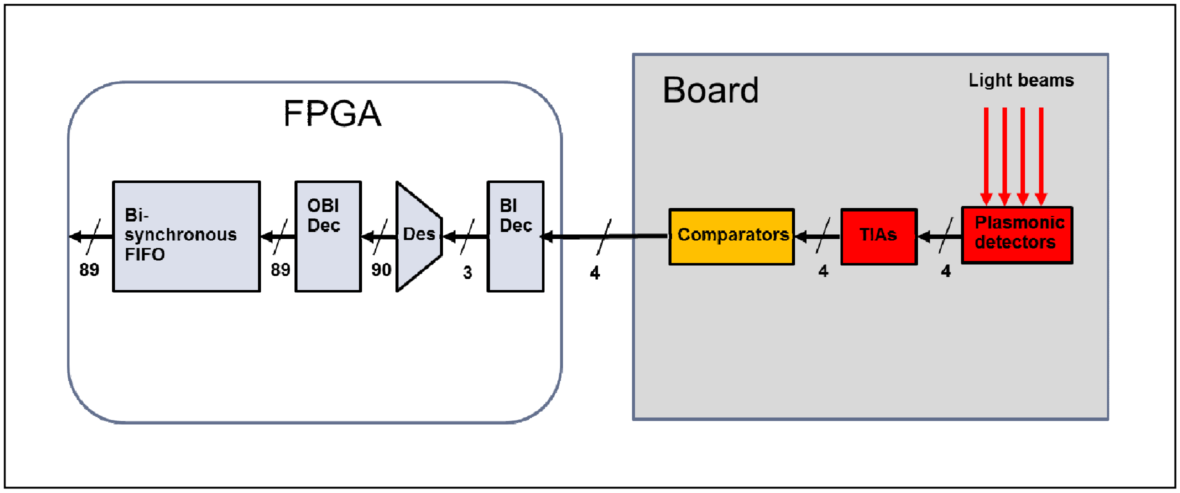

Figure 2-2: Demonstrator DDCM PHY adapter receiver microarchitecture

The NAVOLCHI demonstrator DDCM PHY adapter receiver is composed of the following building-blocks:

- a set of M **Trans-Impedance Amplifiers** (TIAs) responsible for translating the output currents generated by the plasmonic photodetectors into voltage levels;
- a set of M voltage **comparators** responsible for associating the proper logic value (‘0‘ or ‘1‘) to the output of the TIAs;
- a **data decoder** for performing the inverse transformation applied to data words by the Bus Inverter Encoder in PHY adapter transmitter (**Bus Inverter Decoder**);
- a **deserializer** to merge the chunks of M-bits, coming from the M plasmonic photodetectors into a N-bits;
- a **data decoder** for performing the inverse transformation applied to data words by the Optical Bus Inverter Encoder in PHY adapter transmitter (**Optical Bus Inverter Decoder**);
- a **bi-synchronous FIFO**, playing the twofold role of retiming stage and storage element.

3. Building-blocks

In this section each DDCM PHY adapter building-block specified and implemented is described in terms of functionality and, when required, structure.

Transmitter

The DDCM PHY adapter transmitter is composed of some digital blocks, implemented as synthesizeable RTL, and some analog blocks, modeled as behavioral RTL blocks that will be implemented as custom design at transistor level.
Specifically, the blocks interacting with the system are digital, while the blocks interacting with plasmonic devices are analog.

Digital blocks

The PHY adapter transmitter digital blocks are the bi-synchronous FIFO, the encoders for power consumption reduction and the serializer.

Bi-synchronous FIFO

The bi-synchronous FIFO plays the twofold role of storage buffer, where incoming data are stored before being processed for transmission, and retiming stage, breaking critical paths between data sources and off-chip transmitter.
Data are stored into the FIFO at the speed of the digital system, and are taken from the FIFO at the speed of the off-chip transmitter.

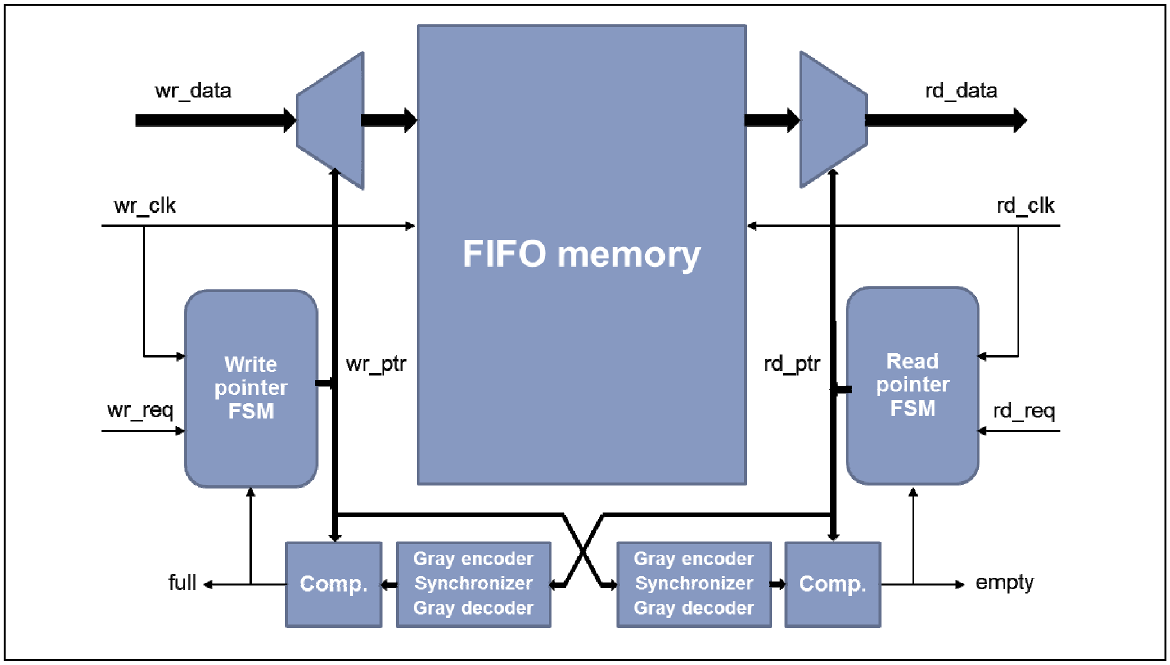

Figure 3-1 : Bi-synchronous FIFO micro-architecture

Optical Bus Inverter Encoder

The Optical Bus Inverter (OBI) Encoder aims at minimizing the number of ‘1‘ in the data word to be transmitted, in order to minimize the number of emitters turned-on.

The algorithm implemented by this block consists in counting the number of ‘1‘ in the data word, and inverting the data itself if such a number is greater than half data size; in case of inversion a specific flag (inv) is set.

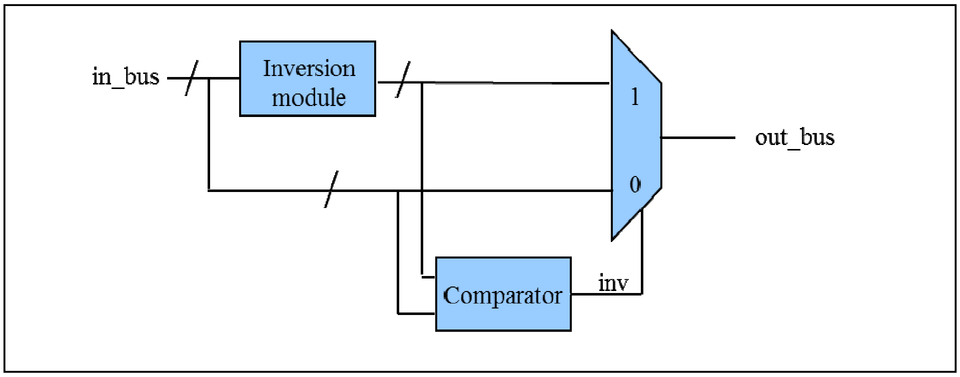

Figure 3-2 : Optical Bus Inverter Encoder microarchitecture

Serializer

The serializer performs the segmentation of the incoming data according to the selected output data size.

It is implemented as a parametric block that can be properly configured depending on the requirements of the system; in the case of the NAVOLCHI demonstrator the incoming data is 90 bits wide, and the output data is 3 bits wide, meaning that a data word is segmented into 30 smaller chunks of bits.

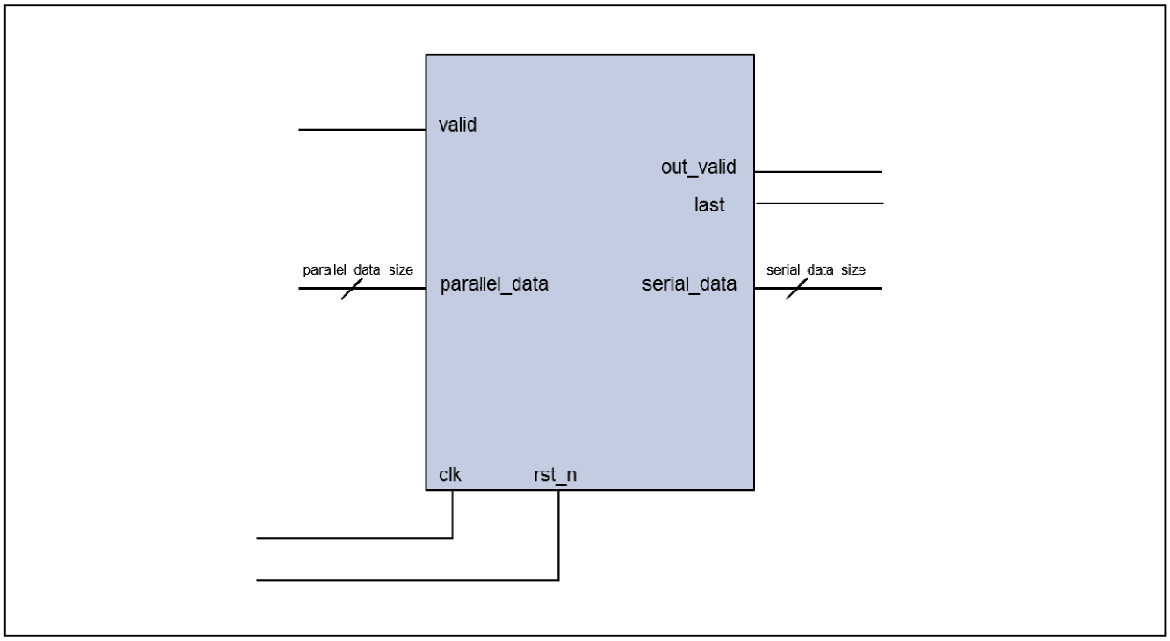

Figure 3-3 : Serializer top level

Bus Inverter Encoder

The Bus Inverter (OI) Encoder aims at minimizing the Hamming distance between two consecutive data words to be transmitted, in order to minimize the switching activity of emitters.
The algorithm implemented by this block consists in evaluating the Hamming distance between two back-to-back data words, i.e. the current and the previous ones, and inverting the current one if such a distance is greater than half data size; in case of inversion a specific flag (inv) is set.

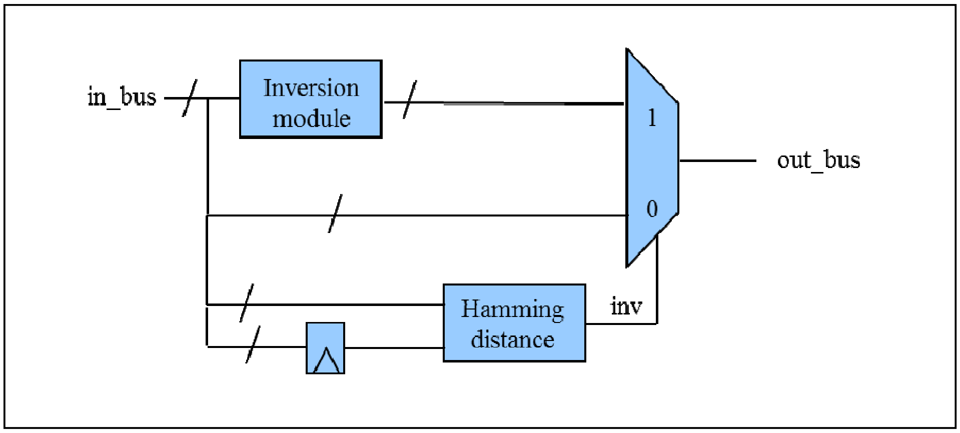

Figure 3-4 : Bus Inverter Encoder microarchitecture

Analog blocks

The PHY adapter transmitter analog blocks are the modulator driver and the modulator.

Modulator driver

This block has the task of shaping the voltage levels generated by the digital parts according to the used CMOS technology in order to be able to drive the analog parts controlling the plasmonic emitters.
Just the VHDL behavioural model has been implemented for this block at the moment.

Modulator

This block has the task to generate the proper modulation voltage for the plasmonic modulator according to the stream of bits to be transported across the optical network.
Also for this block just the VHDL behavioural model has been implemented for this block at the moment.

Receiver

Also the DDCM PHY adapter receiver is composed of some digital blocks, implemented as synthesizeable RTL, and some analog blocks, modeled as behavioral RTL blocks that will be implemented as custom design at transistor level.
The blocks interacting with the system are digital, while the blocks interacting with plasmonic devices are analog.

Analog blocks

The PHY adapter receiver analog blocks are the Trans Impedance Amplifier (TIA) and the comparator.

Trans Impedance Amplifier

This block acts as a current-to-voltage converter, taking the photocurrent generated by the plasmonic detector and generating a proper voltage, representing a logic ‘0’, a logic ‘1’ or simply noise (i.e. no detection).
Just the VHDL behavioural model has been implemented for this block at the moment.

Comparator

This block has the task of shaping the voltage levels generated by the TIA according to the used CMOS technology. It has also the task of generating a start signal to when actual information is received, allowing keeping the digital parts of the receiver idle when the TIA input current is simply noise and not a signal carrying actual information.
Also for this block just the VHDL behavioural model has been implemented at the moment.

Digital blocks

The PHY adapter receiver digital blocks are the deserializer, the decoders for power consumption reduction and the bi-synchronous FIFO.

Bus Inverter Decoder

This block relies on the 4th bit of the incoming data (*inv* flag) to determine whether the remaining 3-bits have to be inverted or not, according to what has been done by the related encoder in the transmitter, in order to obtain the original 3-bits data.

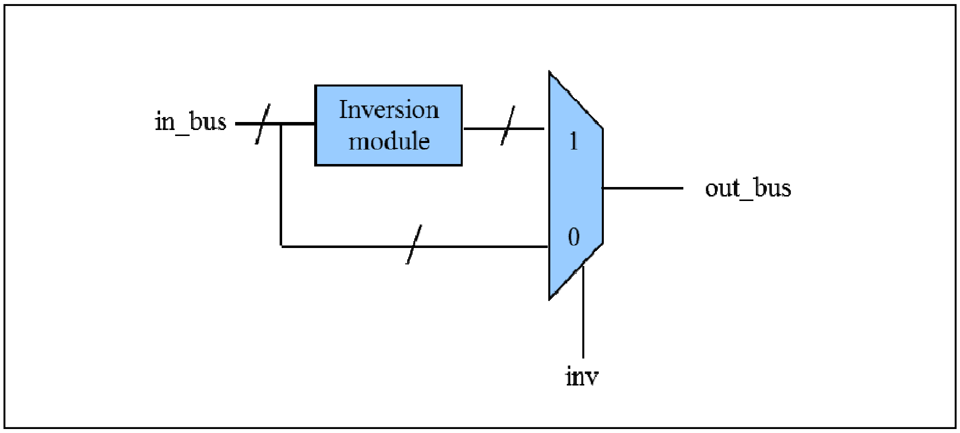

Figure 3-5 : Bus Inverter Decoder microarchitecture

Deserializer

The deserializer performs the reassembly of the incoming data according to the selected output data size.
It is implemented as a parametric block that can be properly configured depending on the requirements of the system; in the case of the NAVOLCHI demonstrator the incoming data is 3 bits wide, and the output data is 90 bits wide, meaning that an output data is built using 30 input 3-bits words.

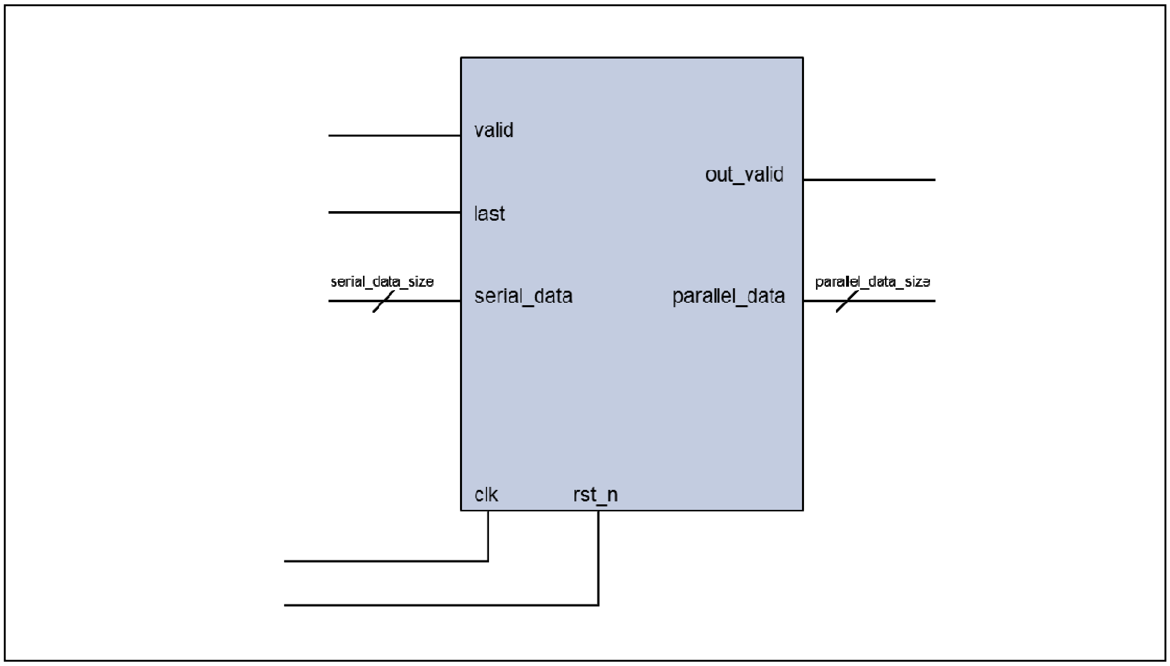

Figure 3-6 : Deserializer top level

Optical Bus Inverter Decoder

This block relies on the 90th bit of the incoming data (*inv* flag) to determine whether the remaining 89-bits have to be inverted or not, according to what has been done by the related encoder in the transmitter, in order to obtain the original 89-bits data.

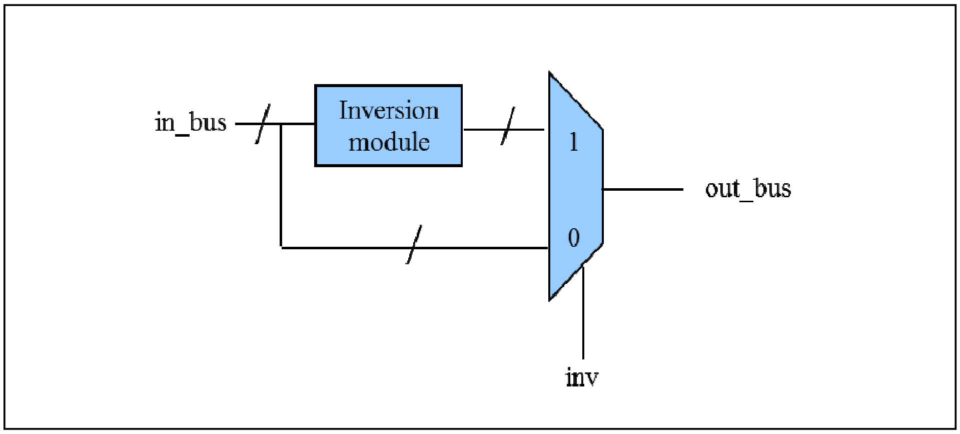

Figure 3-7 : Optical Bus Inverter Decoder microarchitecture

Bi-synchronous FIFO

Also in the receiver the bi-synchronous FIFO plays the twofold role of storage buffer, where incoming data are stored after they have been processed after reception, and retiming stage, breaking critical paths between off-chip receiver and data destinations.

4. Design flow

This section describes the different steps followed for moving from the idea to the logic implementation of the DDCM PHY adapter; such steps are common for the design of any digital system.

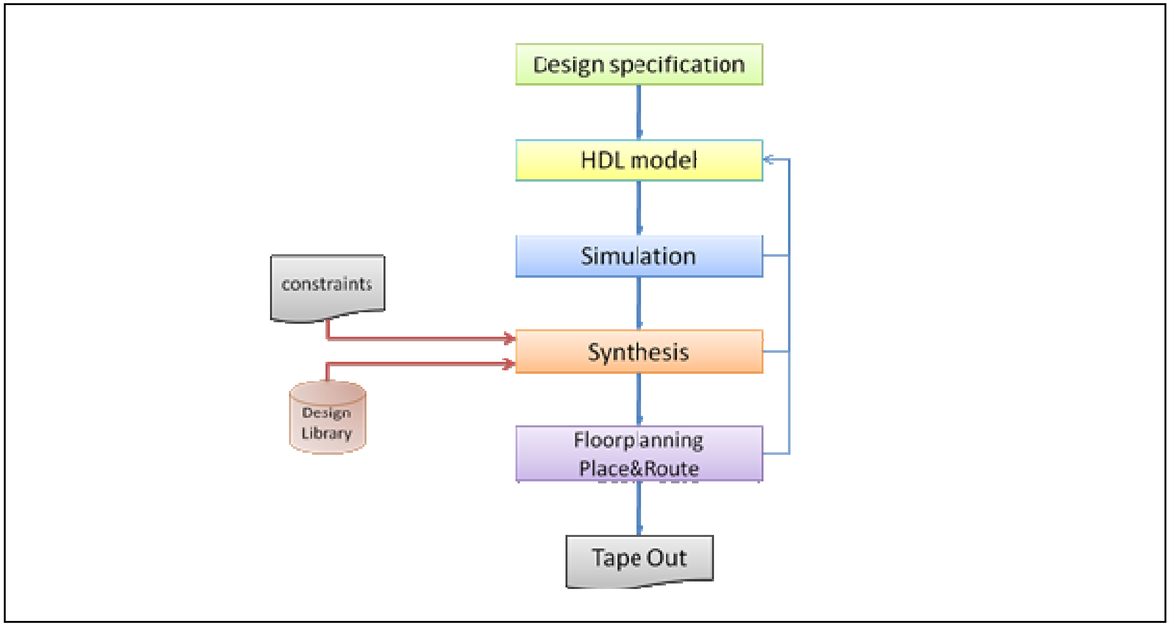

Figure 4-1 : Digital design flow

RTL Design

In this phase the function of a digital block is described by means of an RTL (Register Transfer Level) language, i.e. a high level language allowing to describe the behaviour and the structure of hardware blocks. The language used for the DDCM is VHDL (Very High Speed Integrated Circuits – Hardware Description Language).

Functional verification

Once the description of the structure and the functionality of a digital block are complete, the block is functionally verified exploiting an environment able to inject stimuli into its inputs, and to watch its outputs in order to check if the reaction of the block to its input is the expected one.
The tool used for functional verification is NCSIM from Cadence.

Synthesis

Logic synthesis is the process that translates the RTL description of the block into a technology-dependent gate level netlist, that is a logic schematic of the structure of the digital block exploiting logic components (standard cells) belonging to a specific technology library.
The tool used for logic synthesis is Design Compiler from Synopsys. The technology library is ST proprietary, and the one used for first trials is CMOS 32nm.

Formal verification

Formal verification or equivalence check is the process that guarantees that the gate level netlist obtained by the synthesis phase has a functionality equivalent to the one of the RTL description of the digital block.

The tool used for formal verification is Formality from Synopsys.

The outcome of the described design flow is functional gate level netlist, characterized in terms of area occupancy, timing (i.e. operating speed) and power consumption. Depending on the objectives, such a netlist can be considered as the starting point for the back-end phase of the design, i.e. the phase leading to the generation of the layout and then the masks for the manufacturing of the system, or it can be mapped onto an FPGA equipment in order to have an early physical implementation of the system itself, to be used for validation purposes and for other activities depending on the system complexity, such as prototyping, early software development, and more.

5. Conclusion

This document summarizes the work carried out at ST within NAVOLCHI project for workpackage 2.

The building-blocks of the Dual Die Communication Module PHY Adapter transmitter and receiver have been specified and implemented; the digital parts have been implemented as synthesizeable VHDL, that in next phases of the work will be mapped onto FPGA, while the analog parts have been modeled exploitintg behavioral VHDL in order to run simulations of the whole system so to be able to characterize it at an early stage.