



Deliverable – D17

Feasibility study for using GALS NoC in the GALS system implementation

| | |
|---|---|
| Grant Agreement No: | 214364 |
| Project acronym: | GALAXY |
| Project title: | GALS InterfAce for CompleX Digital System Integration |
| Funding Scheme: | STREP |
| Date of latest version of Annex I against which the assessment will be made: | 28.10.2008. |
| Contractual Date of Delivery to the EC: | 31. Oct. 09 |
| Actual Date of Delivery to the EC: | 31. Oct. 09, revision 25.Nov.10 |
| Author(s): | Milos Stanisavljevic, Yusuf Leblebici (EPFL) |
| Participant(s): | EPFL, UNIBO |
| Work Package: | WP8 |
| Security: | Public |
| Nature: | Report |
| Version: | 2 (updated) |
| Total number of pages: | 25 |

Abstract:

This report investigates the effectiveness of using NoCs in the target hardware accelerator design and assesses the applicability of the NoC solution. The system planned to be GALSified is investigated in the scope of trade off between NoC implementation difficulties and benefits. An alternative hardware accelerator design that could benefit more from NoC implementation is discussed. Finally, the optimal GALS interface architecture for targeted application is analyzed.

- **Keyword list: point-to-point topology, multi-point topology, OFDM baseband processor, deinterleaver, Viterbi decoder, LDPC decoder, Reed-Solomon decoder**



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

| <i>Function</i> | <i>Responsibility</i> | <i>Date</i> | <i>Signature</i> |
|---------------------|-----------------------|-------------|------------------|
| <i>Written by:</i> | Milos Stanisavljevic | 28.10.2009 | |
| | Yusuf Leblebici | 22.11.2010 | |
| <i>Checked by:</i> | Milos Krstic | 25.11.2010 | |
| <i>Approved by:</i> | | | |

Reserved to EC

| | | | |
|---------------------|--|--|--|
| <i>Approved by:</i> | | | |
|---------------------|--|--|--|



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Never stop thinking



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

CHANGE RECORDS

| <i>ISSUE</i> | <i>DATE</i> | <i>§ : CHANGE RECORD</i> | <i>AUTHOR</i> |
|--------------|-------------|--------------------------|----------------------|
| 1 | 28-Oct-09 | Main evaluation | Milos Stanisavljevic |
| 2 | 22-Nov-10 | Readdressing the study | Yusuf Leblebici |



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

BIBLIOGRAPHIC RECORD

| | |
|---|--|
| Project Number: | 214364 |
| Project Title: | GALAXY |
| Deliverable Type: | Report |
| Deliverable Number: | D17 |
| Contractual Date of Delivery: | 31. Oct. 2009 |
| Actual Date of Delivery: | 31. Oct. 2009, rev 25.Nov.2010 |
| Title of Deliverable: | Feasibility study for using GALS NoC in the GALS system implementation |
| Work package contributing to the Deliverable: | WP8 |
| Authors: | Milos Stanisavljevic, Yusuf Leblebici |
| Abstract | This report investigates the effectiveness of use of NoCs in the target hardware accelerator design and assesses the applicability of the NoC solution. The system planned to be GALSified is investigated in the scope of trade off between NoC implementation difficulties and benefits. An alternative hardware accelerator design that could benefit more from NoC implementation is discussed. Finally, the optimal GALS interface architecture for targeted application is analyzed. |
| Keywords | point-to-point topology, multi-point topology, OFDM baseband processor, deinterleaver, Viterbi decoder, LDPC decoder, Reed-Solomon decoder |
| Confidentiality Level | Public |
| Name of Client: | EC |
| Distribution List: | GALAXY, EC, internet |
| Authorised by: | |
| Issue: | 2 |
| Document ID: | D17 |
| Total Number of Pages: | 25 |
| Contact Details: | krstic@ihp-microelectronics.com |



TABLE OF CONTENTS

| | | |
|----------|---|-----------|
| 1 | INTRODUCTION | 6 |
| 2 | REFERENCES | 12 |
| 2.1 | ACRONYMS | 12 |
| 2.2 | REFERENCE DOCUMENTS..... | 12 |
| 3 | GALS HARDWARE ACCELERATOR..... | 14 |
| 3.1 | THE CHOICE OF DESIGN..... | 14 |
| 3.2 | BENEFITS AND DIFFICULTIES IN IMPLEMENTING NOC IN DESIGN..... | 19 |
| 3.3 | ASSESSING SYNCHRONIZATION INTERFACES FOR GALS BASED NETWORK-ON-CHIP DESIGN..... | 22 |
| 4 | FINAL DISCUSION AND CONCLUSION | 25 |

LIST OF FIGURES

| | | |
|-------------------|--|-----------|
| Figure 1: | Generic GALS architecture..... | 7 |
| Figure 2: | Typical NoC architecture..... | 7 |
| Figure 3: | Typical topology and the global node layout in a NoC architecture.... | 8 |
| Figure 4: | Architecture of the FAUST NoC solution [BEI06], [LAT07]..... | 7 |
| Figure 5: | Top level block diagram of OFDM processor..... | 15 |
| Figure 6: | FFT block diagram..... | 16 |
| Figure 7: | Moonrake TX test chip floorplan implementation..... | 17 |
| Figure 8: | Architecture of the Moonrake transmitter..... | 17 |
| Figure 9: | GALS Partitioning in the Moonrake TX des..... | 18 |
| Figure 10: | Architecture of 64-point pipelined GALS FFT processor..... | 20 |
| Figure 11: | Amplitude spectrum of supply current | |
| | (a) without clock modulation | |
| | (b) with clock modulation (fc=100MHz). | 20 |
| Figure 12: | Block diagram of the NoC sub-system test design..... | 24 |



1 INTRODUCTION

Very Large Scale Integrated (VLSI) circuits designed using modern Computer Aided Design (CAD) tools are becoming faster and larger, incorporating millions of smaller transistors on a chip. VLSI designs can be divided into two major classes: Synchronous and Asynchronous circuits. Synchronous circuits use global clock signals that are distributed throughout their sub-circuits to ensure correct timing and to synchronize their data processing mechanisms [SU02, SA04]. Asynchronous circuits contain no global clocks. Their operation is controlled by locally generated signals [OH02]. Asynchronous circuits [EBE04] have many potential advantages over their synchronous equivalents including lower latency, low power consumption, and lower electromagnetic interference [EBE04][SAK07].

Synchronous design methodologies have been plagued by the necessity to distribute the centralized clock all throughout the chip with an acceptably low skew. This task has become even more difficult with the ever decreasing feature size of modern technologies. The recent trend to integrate more and more functions on a single chip, called System-on-Chip (SoC), provides additional challenges. SoCs require a great deal of modularity, where previously designed hardware modules can easily be embedded in a larger design. This typically results in a system that consists of several hardware blocks that have been designed to operate with different clock frequencies. To derive synchronous local clocks from a centralized clock in such a system is an involved task without a trivial solution [GUR02].

Asynchronous design techniques have always attracted attention as an alternative, especially for SoC integration, as they do not rely on a centralized clock. This alleviates problems related to clock distribution and enables the integration of hardware blocks with different clock domains. However, asynchronous design techniques also have a number of well known shortcomings, most notably complicated design methodologies and lack of reliable tools and test methodologies [GUR02].

The Globally-Asynchronous Locally-Synchronous (GALS) is a relatively new VLSI system design methodology, that promises to combine the advantages of both synchronous and asynchronous operations [CHA84]. In GALS, rather coarse grained synchronous functional blocks are surrounded by self-timed wrappers that include a local clock generator and asynchronous communication ports. The functional blocks operate synchronous to the local clock but communicate asynchronously with similar blocks [GUR02].

The basic GALS paradigm is based on a system composed of number of synchronous blocks designed in a traditional way. However, it is assumed that clocks of such synchronous systems are not necessarily correlated and consequently that those synchronous systems communicate asynchronously using handshake channels. Locally synchronous modules are usually surrounded by asynchronous wrappers providing such inter-block data transfer. The principle architecture of GALS for point to point dataflow structures is given in Figure 1. Practical GALS implementations may form much more complex structures, such as bus or NoC structures for inter-block communications and use different data synchronization mechanisms, but all proposals are based on the simple structure shown in Figure 1, while the data transfer between the locally synchronous blocks can be implemented using pausable clocking, boundary synchronization or FIFO-like interface structures.

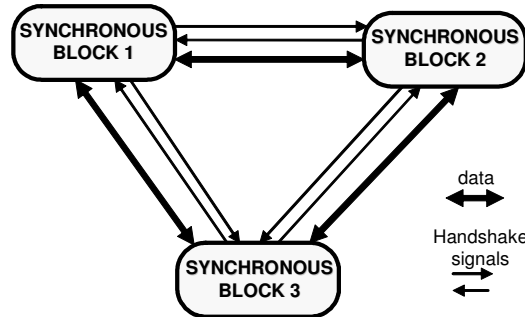


Figure 1: Generic GALS architecture

Globally-asynchronous locally-synchronous (GALS) systems may become a solution for nowadays challenges in the field of VLSI design. The ITRS road-map [SIA06] predicts that, as a solution to the clock distribution problem, GALS will become mainstream design technique in the near future. In a GALS system, a number of synchronous islands of logic communicate asynchronously using a suitable interconnect. Fully synchronous chips are becoming not feasible anymore due to clock distribution and power consumption problems. The value of GALS lies in combination of well-known synchronous design methods and relative simple asynchronous communication channels. The key components are the communication control ports around the synchronous modules and the stretchable clock also called a wrapper. This clock has an unbound delay and is controlled by events the asynchronous channel [BL02]. GALS is a solution to combine the advantage of asynchronous and synchronous circuits design.

Recently, Networks on Chip (NoC) [DEM06] have emerged as one of the important techniques to develop complex System on Chip designs. The NoC paradigm seems to be very attractive solution for the future chip interconnects since NoC aims to provide scalable communication bandwidth through a modular interconnect design. The typical NoC architecture is illustrated in Figure 2.

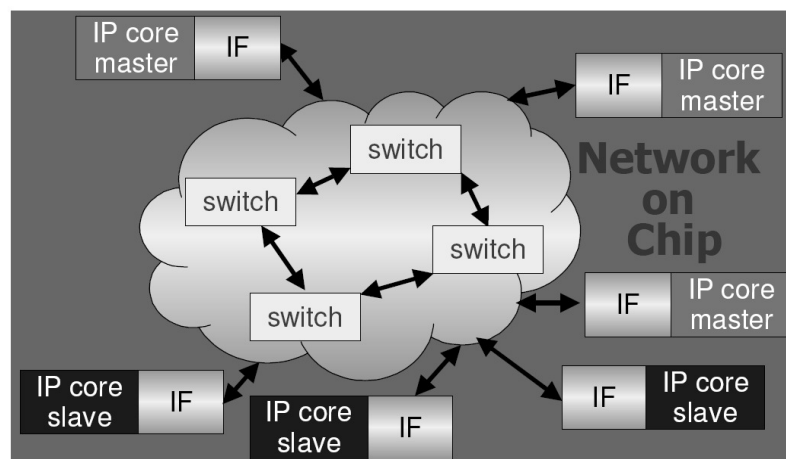


Figure 2: Typical NoC architecture



All NoCs have three fundamental building blocks, namely, switches (also called routers), Network Interfaces (NIs) (also called network adapters) and links. The NoC is instantiated by deploying a set of these components to form a topology and by configuring them in terms of buffer depth, etc. The backbone of the NoC consists of switches, whose main function is to route packets from sources to destinations. Some NoCs rely on specific topological connectivity, such as octagon or ring to simplify the control logic, while others allow for arbitrary connectivity providing more flexible matching to the target application. NoCs can be based on circuit or packet switching, or a mix of both; the former is aimed at providing hard QoS guarantees, while the latter optimizes the efficiency for the average case. When packet switching is chosen, switches provide buffering resources to lower congestion and improve performance. They also handle flow control issues, and resolve conflicts among packets when they overlap in requesting access to the same physical links. Two of the most usual flow control protocols involve switch-to-switch communication and are retransmission based (i.e., packets are optimistically sent but a copy of them is also stored by the sender, and, if the receiver is busy, a feedback wire to request retransmission is raised) or credit-based (i.e., the receiver constantly informs the sender about its ability to accept data, and data are only sent when resources are certainly available). End-to-end flow control schemes where peripheral NIs directly exchange flow control information with each other, are more rarely used because of their buffering requirements; the most common usage scenario involves NoCs that implement circuit switching.

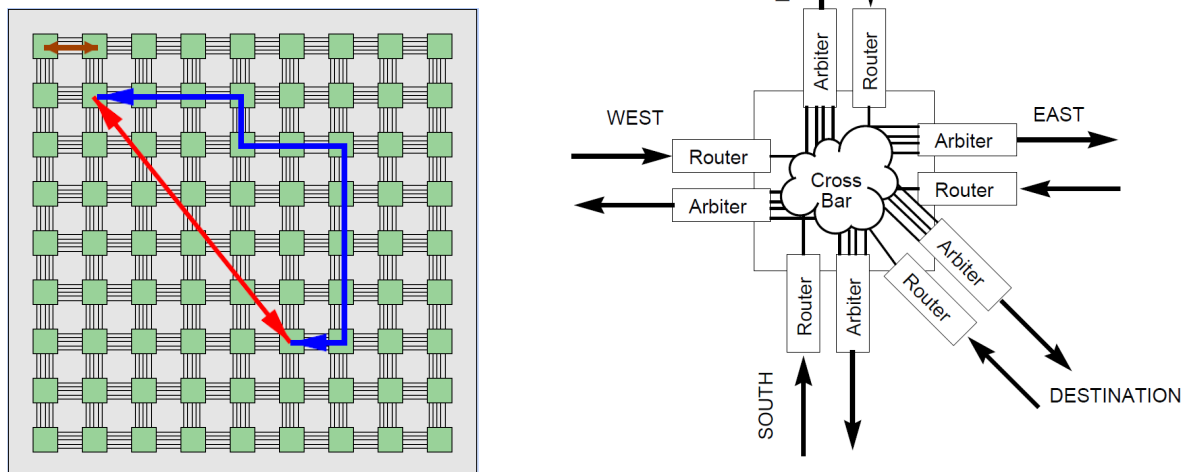


Figure 3: Typical topology and the global node layout in a NoC architecture

The lesson learned from state-of-the-art NoC prototypes is that almost 50% of total NoC power is drained by the clock tree (clock distribution and flip-flops). Moreover, the problem of distributing the global clock in a chip with minimal clock skew is getting difficult to solve due to increases in clock frequencies, smaller feature sizes and growing design complexities, which indicates that a global timing notion will not be feasible for highly integrated nanoscale designs.



GALAXY

GALS InterfACE for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

An NI is needed to connect each core to the NoC. NIs convert transaction requests/responses into packets and vice versa. Packets are then split into a sequence of Flow control unITS (FLITS) before transmission, to decrease the physical wire parallelism requirements. NIs are associated in NoCs to system masters and system slaves. Many current NoC solutions leverage static source routing, which means that dedicated NI Look-Up Tables (LUTs) specify the path that packets will follow in the network to reach their final destination. This type of routing minimizes the complexity of the routing logic in the NoC. As an alternative, routing can be performed within the topology itself, normally in an adaptive manner; however, performance advantages, in-order delivery and deadlock/livelock freedom are still issues to be studied in the latter case.

In general, two different clock signals can be attached to NIs: the first one drives the NI front-end, the side to which the external core is attached, and the second one drives the NI back-end, the internal NoC side. These clocks can, in general, be independent. This arrangement enables the NoC to run at a different (and potentially faster) clock than the attached cores, which is crucial to keep transaction latency low.

GALS technology is generally viewed to be a breakthrough technology for future system interconnect designs. There are many parallels between the design approach of GALS systems and NoCs are generally accepted as a great application for GALS systems. GALS-based NoCs would allow the synchronous design of network nodes at their optimum clock frequency, while facilitating asynchronous communication between modules. This would reduce the timing convergence constraints during back-end physical design steps, remove the power-hungry global clock tree and pave the way for new variation-tolerant on-chip interconnection schemes. Unfortunately, although these considerations would appear to be common sense, they have not been enough to remove the barriers to the adoption of GALS interconnects due to the lack of convincing analysis and exploration frameworks, crossbenchmarking with synchronous solutions, proven robustness against nanoscale physics effects and tool support.

Even though the majority of GALS implementations are point-to-point GALS chips, there are already a few NoC platforms based on the GALS architecture [DOB05, BEI05, BEI06, BJE05].

In general, results show a strong case in favor of NoC design over the standard shared bus in large or high-bandwidth systems, as long as the data flow architecture is not strictly point-to-point. For NoC systems, the performance break-even size (number of nodes) compared to a shared bus can be as low as 5 masters, while at a size of 15 nodes the NoC is already about 300% faster than the shared bus. In cases where the system architecture favors the NoC approach, GALS NoC typically shows a 25-30% improvement in the delay time compared to the synchronous NoC.

For example, Beignet et al. have implemented On-chip and Off-chip Interfaces for a GALS NoC architecture [BEI06]. By using an existing multi-clock synchronization FIFO based on gray code, GALS interfaces were designed which can be managed by standard implementation tools. Moreover, for the targeted NoC protocol, the GALS interfaces also handle NoC FLIT priorities in order to preserve the NoC QoS. For Off-chip interfaces, a new concept of mixed synchronous/asynchronous dual-mode NoC port was proposed for efficient Off-chip NoC interfaces for NoC based open-platform prototyping. These interfaces have been



successfully implemented, and integrated in a complete NoC prototype chip of about 4.5 Million gates for Telecom applications (802.11a and MCCDMA standards). The architecture of the FAUST NoC is shown in Figure 4.

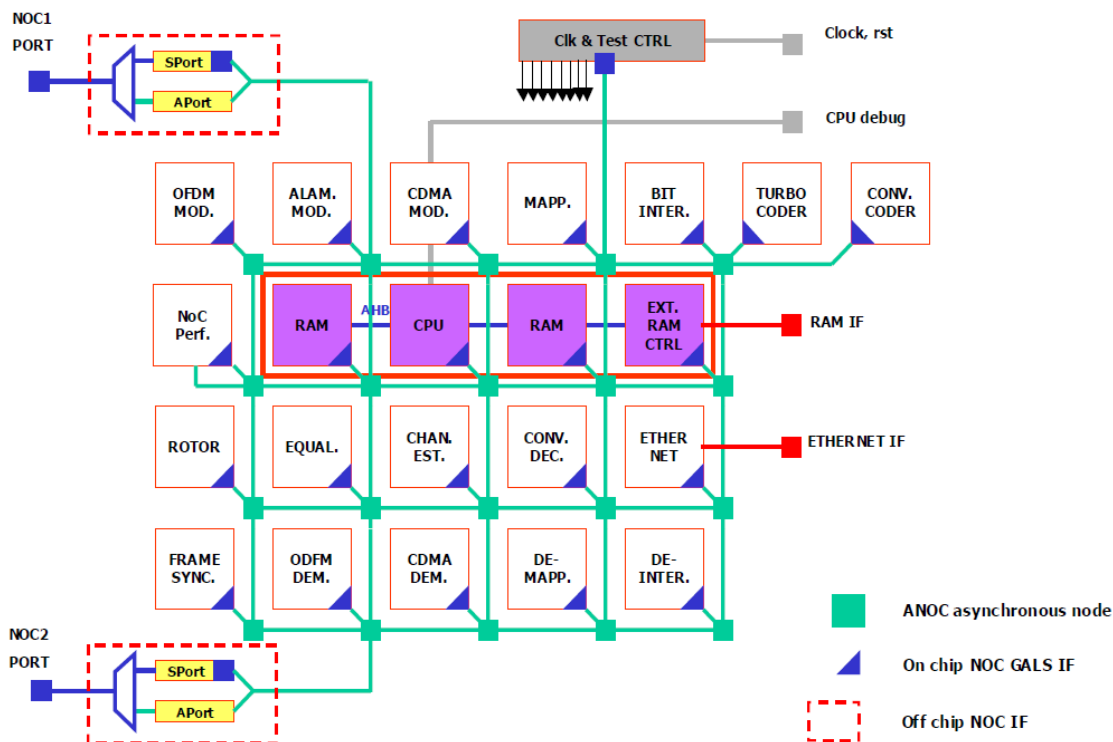


Figure 4: Architecture of the FAUST NoC solution [BEI06], [LAT07]

The NoC On-Chip and Off-Chip interfaces have been fully implemented using a 0.13 μm technology from STMicroelectronics. The standard cell library provided by the foundry was enlarged with some specific cells. This library contains mostly optimized C-elements (2/3/4 inputs, with set/reset, and asymmetric gates) and arbiter cells (Mutex and Synchronizer cells). Without such optimized C-elements, pure standard-cell design can still be achieved by mapping the C-elements to existing and/or gates. Both the On-chip and Off-chip NoC interfaces have been written in VHDL RTL, to which have been added the required C-elements and synchronizer instances [BEI06].

Asynchronous-to-synchronous (A-S) and synchronous-to-asynchronous (S-A) network interfaces have been designed with dedicated clock management, and the operation of the entire network has been validated.

This example has successfully demonstrated that the GALS NoC architecture can be implemented in an efficient manner, as long as the overall system architecture does not force a point-to-point solution.



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

Most recently, a low-overhead asynchronous interconnection network for GALS chip multiprocessors has been demonstrated by Horak et al. [HOR10]. In this work, mixed-timing interfaces with new custom protocol converters were proposed to provide robust communication between synchronous and asynchronous timing domains. Network primitives were assembled into a Mesh-of-trees topology for preliminary system-level performance evaluation against a synchronous MoT network, first in isolation and then with accompanying mixed-timing interfaces. Finally, the mixed-timing network is embedded and co-simulated with the XMT processor and the performance is evaluated by running several parallel kernels. The new GALS XMT processor provides comparable performance to the existing synchronous XMT except in the most challenging case of high clock rate and high traffic rate.

To summarize, recent research indicates that a strong case can be made for GALS implementation of NoC architectures, as long as the overall system architecture and data flow patterns are suitable for such implementations. Several examples have already been demonstrated for the feasibility of this approach, including various telecom-related applications.

In the following text we will investigate the effectiveness of using of NoCs in the target hardware accelerator design and assess the applicability of the NoC GALSification. This discussion will mainly focus on the specific architecture that has been selected for the target design, in the GALAXY project.



2 REFERENCES

2.1 ACRONYMS

| | |
|-------------|---|
| CAD | Computer Aided Design |
| CP | Content Protection |
| FEC | Forward Error Correction |
| FIFO | First-in First-out |
| GALS | Globally Asynchronous Locally Synchronous |
| IP | Intellectual Property |
| LDPC | Low-density Parity-check Code |
| MIMO | Multiple-Input and Multiple-Output |
| NI | Network Interface |
| NoC | Network on Chip |
| OFDM | Orthogonal Frequency Division Multiplex |
| QoS | Quality of Service |
| RS | Reed-Solomon |
| SoC | System on Chip |

2.2 REFERENCE DOCUMENTS

* Where we put the reference outside of the sentence at the end of a paragraph that means that the whole paragraph is cited from the referenced document

| Ref. | Document Title |
|-------------|--|
| [BEI05] | E. Beigne, et al. An Asynchronous NOC Architecture Providing Low Latency Service and its Multi-Level Design Framework. In Proc. of the 11th IEEE Int. Symp. on Asynchronous Circuits and Systems, 2005. |
| [BEI06] | E. Beigne, P. Vivet, Design of On-chip and Off-chip Interfaces for a GALS NoC Architecture, Proceedings of 12th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC'06), Grenoble, France, pp. 172-181, March 2006. |
| [BEI08] | Edith Beigné, Fabien Clermidy, Sylvain Miermont, Pascal Vivet: Dynamic Voltage and Frequency Scaling Architecture for Units Integration within a GALS NoC. NOCS 2008: 129-138 |



GALAXY

GALS InterfAce for CompleX Digital
System Integration

Confid. Level: Public
Date : 25/11/2010
Issue: 1

| | |
|---------|---|
| [BJE05] | T. Bjerregaard, et al, An OCP Compliant Network Adapter for GALS-based SoC Design Using the MANGO Network-on-Chip, Proc. of the International Symposium on System-on-Chip (SoC'05), pp. 171-174, 2005. |
| [BL02] | Blaauwendraad, B., Student thesis, Title: TIR, design and testing of a Simple GALS, Linköping University, Department of Electrical Engineering, 2002 |
| [DOB05] | R. Dobkin, et al, An Asynchronous Router for Multiple Service Levels Networks on Chip, Proc. of 11th IEEE Int. Symp. on Asynchronous Circuits and Systems (ASYNC 2005), New York, Mar 2005. |
| [EBE04] | Aristides Efthymiou, John Bainbridge, Douglas A. Edwards, Adding Testability to an Asynchronous Interconnect for GALS SoC, Asian Test Symposium 2004: 20-23 |
| [CHA84] | D. M. Chapiro, Globally Asynchronous Locally Synchronous. Systems. PhD thesis, Stanford University, 1984. |
| [DEM06] | G. DeMicheli, L. Benini, Networks on Chips: Technology and Tools (Systems on Silicon), Morgan Kaufmann, ISBN 978-0123705211, 2006 |
| [GRA07] | E. Grass, et al, 60 GHz SiGe-BiCMOS Radio for OFDM Transmission, The IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, USA, pp. 1979-1982, 27 - 30 May 2007. |
| [GUR02] | F. Gürkaynak et al., A Functional Test Methodology for Globally-Asynchronous Locally-Synchronous Systems, <i>Proc. of ASYNC</i> , pp. 181-189, 2002. |
| [GUR06] | F. Gürkaynak, GALS System Design: Side Channel Attack Secure Cryptographic Accelerators, PhD thesis, Hartung-Gorre Verlag, 2006 |
| [KRS06] | M. Krstic, E. Grass, C. Stahl, M. Piz, System Integration by Request-driven GALS Design, IEEE Proc. Computers & Digital Techniques, Vol. 153, Issue 5, September 2006, pp 362-372. |
| [LAT07] | D. Lattard, et al, "A Telecom Baseband Circuit-Based on an Asynchronous Network-on-Chip", Proc. of the International Solid State Circuits Conference, ISSCC'2007, San Francisco, USA, Feb 2007. |
| [NEX04] | Lines, A., Asynchronous interconnect for synchronous SoC design, IEEE Micro, 2004, 24, (1), pp. 32-41 |
| [OH02] | Eunjung Oh, Soo-Hyun Kim, Dong-Ik Lee, Ho-Yong Choi, High-Level Test Generation for Asynchronous Circuits from Signal Transition Graph, IEICE Trans on Fundamentals of Electronics, Communications and Computer Sciences Vol.E85-A No.12, 2002, pp. 2674-2683 |
| [SA04] | Mathew A. Sacker, Andrew D. Brown, Andrew J. Rushton, Peter R. Wilson, A behavioral synthesis system for asynchronous circuits, IEEE Trans. VLSI Syst. 12(9): 978-994 (2004) |
| [SIA06] | SIA, The International Technology Roadmap for Semiconductors: 2006 Update, Semiconductor Industry Association, San Jose, CA (http://public.itrs.net), 2006. |
| [SU02] | Yin-He Su, Ching-Hwa Cheng, Shih-Chieh Chang, Novel Techniques for Improving Testability Analysis, IEICE Trans on Fundamentals of Electronics, Communications and Computer Sciences Vol.E85-A No.12, 2002, pp. 2901-2912 |



3 GALS HARDWARE ACCELERATOR

3.1 THE CHOICE OF DESIGN

The main ASIC implementation within the GALAXY project is a complex target system using both a standard synchronous CMOS design flow and the GALS design flow developed in this project (WP3, WP4, WP5) in a cutting edge 40 nm CMOS process. Selecting a suitable target platform has been one of the most important decisions. Our main goal is to design a complex GALS system for the high-end communication application with supporting data rates up to 1 Gbps [GRA07]. The purpose of this system implementation is evaluation of the proposed novel GALS design flow, exploration of the GALS interfaces, and power and EMI reduction possibilities. To properly address this, in parallel, the purely synchronous version of the same system will be implemented.

Until now there have been only relatively few complex GALS demonstrators developed, such as the WLAN baseband processor from IHP (0.25um CMOS) [KRS06], the FAUST processor developed at LETI (0.13um CMOS) [LAT07] as well as ALPIN (2008) and Magali (2010) and circuits fabricated in GALS group at ETHZ (0.25um CMOS) [GUR06]. Even though this GALS system implementation is not the ultimate goal of the GALAXY project, we regard the GALS system implementation in this project a necessary vehicle to evaluate achieved improvements in the GALS design flow and architecture. However, the planned GALAXY GALS chip implementation will definitely have many competitive features in comparison with the previous approaches.

Since wireless communication is an application domain which is gaining more and more popularity and which poses significant technical challenges to system designers (performance- and power-wise), and since several group members have prior experience in this field, we have considered target platforms from this field. Our main candidate, on which this research is based, is an accelerator for an OFDM baseband processor with data rates up to 1 Gbps for communication systems in 60 GHz range which is currently under development in IHP [GRA07b]. Currently, the full processor integrates several Viterbi decoders, 256-FFT and IFFT, many complex interleavers/deinterleavers. The top level block diagram is depicted in Figure 5. The processor consists of transmitter and receiver, each of them having up to 12 parallel streams and multiple complex blocks.

Previously this processor was implemented in FPGA, achieving 1 Gbps throughput, with 100MHz operating frequency and 10 streams. The complexity of FPGA realization, and some specific blocks is given in Table 1.

The goal is to increase the throughput with ASIC solution, reduce the power, and area. New standards (802.15.4c) target throughputs of 5 Gbps (up to 10 Gbps). These standards have different modes with different decoding techniques, for example, using deinterleaving together with Viterbi decoding, using LDPC (and/or Reed-Solomon) decoding, combining deinterleaving together with LDPC (and/or RS) decoding. In order to increase throughput we could add more streams (instances) and we could use the higher operating frequency.

The system should support different data-rates, frame-sizes, streaming, MIMO, etc. It is possible to focus on particular decoding technique, and limited set of the data rates and



optimize the system for such implementation. Considering the specific application area of wireless telecommunications, and taking into account the various constraints on the overall system topology, it was decided to focus on a point-to-point GALS solution. In particular, the target design shall be utilized to demonstrate the low EMI properties of GALS architecture, as applied in the point-to-point topology.

However, dedicated test structures for generic NoC GALS implementation shall also be implemented in the frame of the target design, as explained in the following.

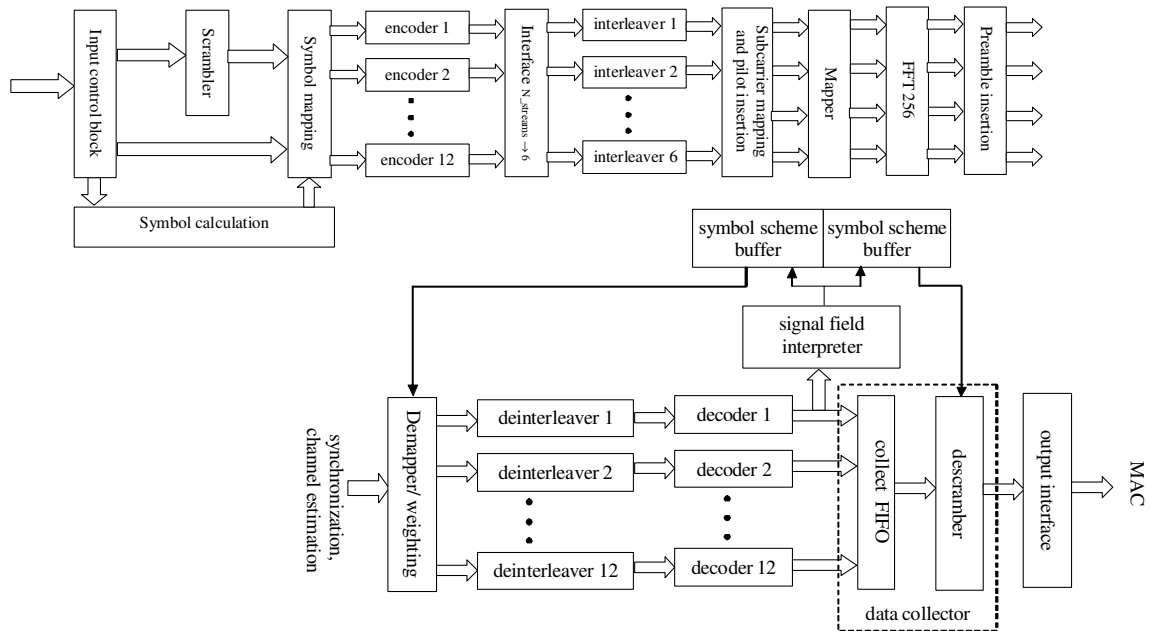


Figure 5: Top level block diagram of OFDM processor

Table 1: Complexity of FPGA realization of OFDM processor

| block | gates (Mil.) | slices | flip-flops | 4-input LUT | BRAM | MULT |
|-------|--------------|--------|------------|-------------|------|------|
| Tx | 7.8 | 12430 | 18069 | 16375 | 111 | 41 |
| Rx | 15.5 | 29142 | 27450 | 46632 | 225 | 56 |

Our OFDM scheme proposes application of 256-FFT. However, it is required to effectively run this FFT with clock frequency of 400 MHz. This is not possible by using the proposed FPGA platform [IAF], or even using more advanced FPGAs (Virtex IV and V).



Therefore we implemented this FFT structure as a combination of the four 64-FFT and 4-FFT. In this way we were able to parallelize the structure with processing 4 samples in on clock cycle with 4 different FFT-64. After that, the output four streams coming out from those 64-FFT are processed in a final 4-FFT stage (see Fig. 6). With this solution we can process samples arriving with 400 MSPS with a 100 MHz FFT processor. This solution, however, also increases the area by about a factor of 4 – which is deemed to be an acceptable trade-off considering the performance increase obtained in this implementation.

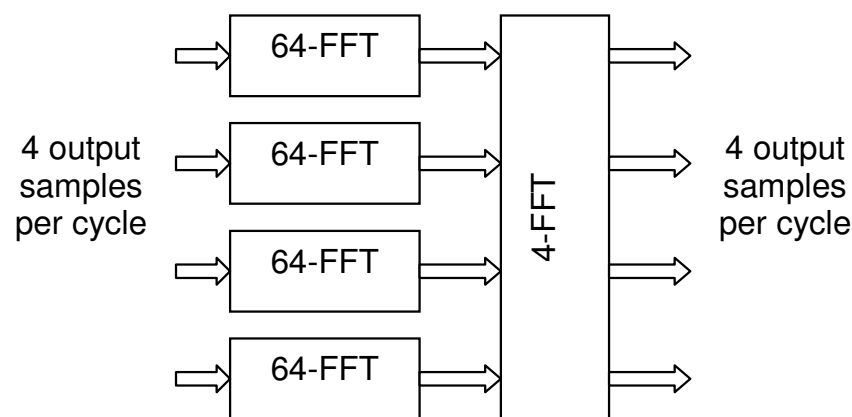


Figure 6: FFT block diagram

One of the goals of GALAXY project is to explore and improve state-of-the-art GALS methods and practically utilize this methodology by applying mature realistic design example. The GALS technique has been already several times implemented, firstly in the point-to-point designs (chips form ETHZ and IHP), and more recently for GALS NoCs (Nexus, and three LETI implementation FAUST (2006), ALPIN (2008) and MAGALI chip (2010). Both concepts are very interesting and we have decided to take the one that better corresponds to the target example platform.

As defined in the project proposal, we have in our disposal a state-of-the-art OFDM baseband processor that was chosen to be the best platform for target implementation. However, the architecture of this design, is not suitable for NoC application. Based on this constraint, we have decided to use the point-to-point GALS concept, meanwhile concentrating on the demonstration of specific benefits that relate to EMI reduction. Point-to-point GALS implementation in GALAXY project is challenging and it has significant impact, due to the real application field, applied technology, improved interfaces, and low-EMI features. Previously, several GALS designs were based on artificial platforms without real application.

In the context of GALAXY project we intend to implement the ASIC version of OFDM transmitter. The goal is to increase the throughput with ASIC solution, reduce the power, and area. This design will be implemented as both synchronous and GALS (globally asynchronous locally synchronous) system. Both designs are to be implemented on the same chip using the same pad ring structure. The block diagram of this test system is shown in Fig. 7 (code name Moonrake). Both versions of OFDM transmitter will be implemented using the same pad frame and basically as one single chip. Pads will be multiplexed and the chip will work in two modes:



GALS or synchronous. Target frequency for both core and pads is 200 MHz. Synchronous clock frequency is controlled by PLL provided by Infineon. Cell area of each Tx component is estimated to 2.2 mm².

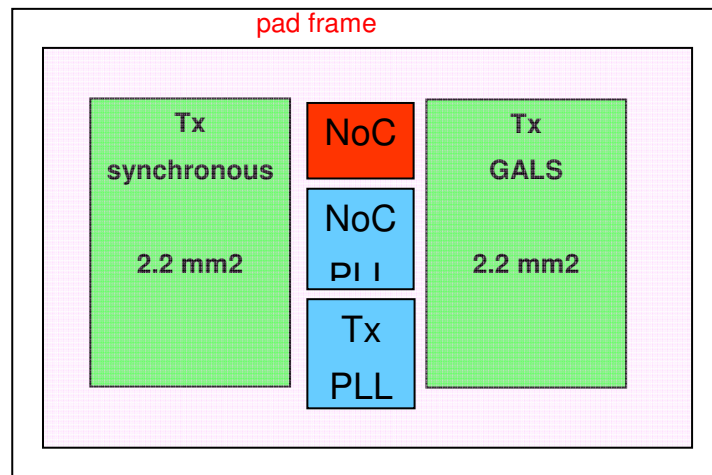


Figure 7: Moonrake TX test chip floorplan implementation

The more detailed internal architecture of Moonrake chip is given in Fig. 8. One can observe that both synchronous and GALS transmitter share the same input and output stage (based on FIFOs). The GALS transmitter bases its clocking on six pausable local clock generators (illustrated with LC). Synchronous transmitter is driven with the common clock sourced from PLL. For programming of Moonrake chip we are using clock independent JTAG interface. This interface is used for setting the parameters of local clock generators, PLL, etc. Additionally we have BIST logic that can perform functional BIST test of both synchronous and GALS transmitter. The BIST architecture is based on classical concepts of LFSR and MISR registers. Finally, a top level clock control block shall be implemented that performs clock division, clock multiplexing, and clock gating.

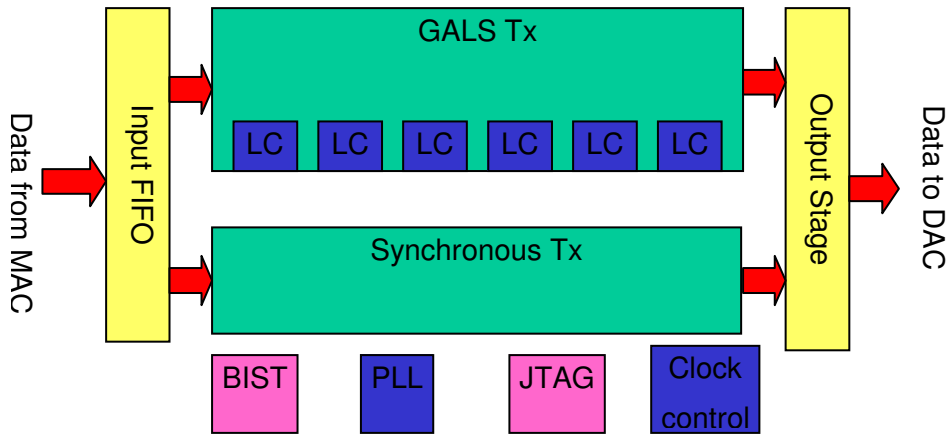


Figure 8: Architecture of the Moonrake transmitter

The GALS part of the design is implemented using point to point GALS approach. The architecture and GALS partitioning of this design is given in Fig. 9.

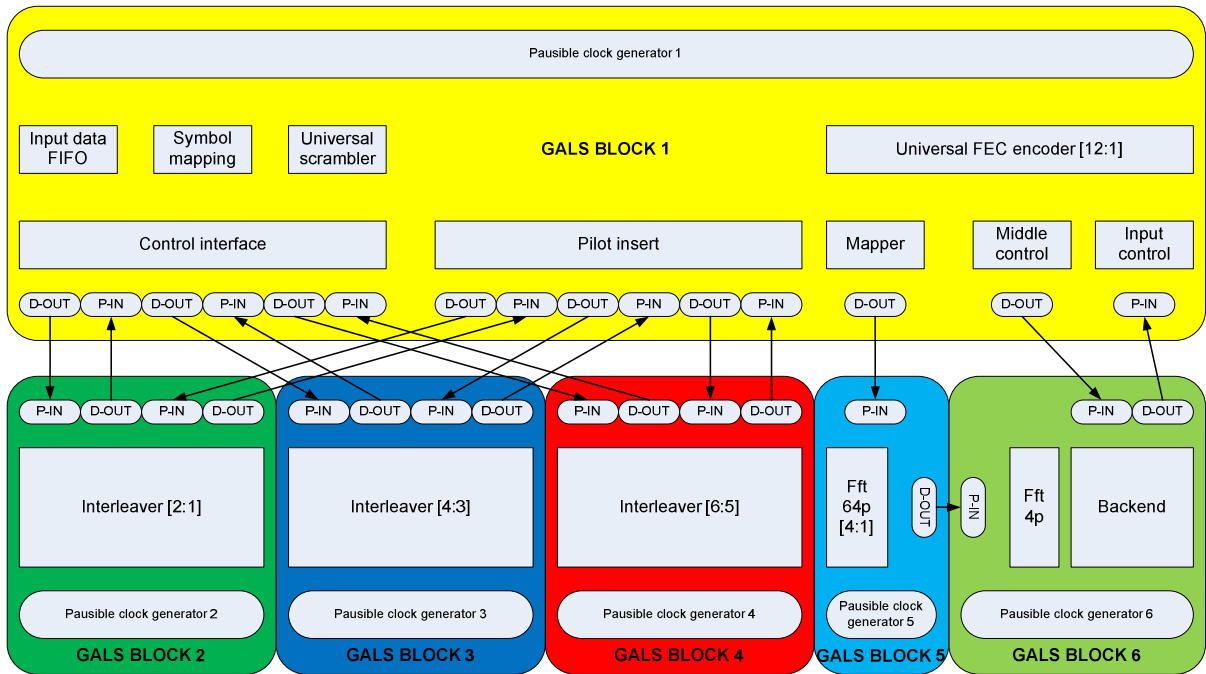


Figure 9: GALS Partitioning in the Moonrake TX design



Note that the Rx module shall not be implemented in the test ASIC, instead, the design efforts shall be concentrated on the comparative demonstration example that is based on the Tx module in synchronous and GALS versions.

3.2 BENEFITS AND DIFFICULTIES IN IMPLEMENTING NOC IN DESIGN

We are exploring the effectiveness of GALS approaches to deep sub-micron processes, and therefore, our approach will provide completely different additional merits to the GALS subject. The planned application in our case seems to be very challenging because of its ad-hoc architecture with very demanding processing requirements. There are several difficulties in designing such a large and complex design as proposed OFDM baseband processor:

- The desired throughput is high and difficult to achieve.
- The occupied area is very large even in such an advanced technology as 40nm.
- GALS partitioning of complex synchronous design is block- and function-dependent.
- Despite maturing of the GALS design flow on course of this project, this is the largest and most complex GALS design reported up to date and is very hard to predict all the possible difficulties in the backend and simulation phase of the designing process.

One of the key benefits of GALS systems that need to be demonstrated and explored is related to the EMI reduction that can be achieved in such systems, with respect to conventional synchronous systems. The following study summarizes the findings of the specific example of a 64-point pipelined GALS FFT processor based on pausable clocking scheme with phase and frequency modulation.

To minimize hardware consumptions, the 64-point FFT was divided and conquered by two cascaded 8-point FFT computation. Low-EMI FFT design is of particular importance for the OFDM communications systems, where large size FFT is required for high data throughput. Hence low-EMI FFT is necessary for substantial mitigation of the unintentional radiated emission from the OFDM systems.

It is obvious that the optimal performance in EMI reduction can be achieved by clock modulation if all the GALS blocks have similar contributions to supply currents. However, to accurately estimate the currents is usually difficult and time-consuming for complex digital systems. As a solution, instead of the current estimation, dynamic power analysis based on the netlist simulation waveforms was performed using Synopsys PrimeTime. To guarantee the analyzing accuracy, quite small timing interval, about 100ps here, was set. Based on the dynamic power analysis, the average power and current of each functional block were estimated, which is accurate enough for system partitioning.

The partitioning of 64-point pipelined FFT design is presented in Table 2. There are in total 4 GALS blocks deployed in the system which are average in terms of supply current. Figure 10 illustrates the architecture of pausable clocking based GALS FFT processor.

Table 2: System partitioning of GALS FFT



| | GALS B1 | GALS B2 | GALS B3 | GALS B4 |
|--------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Functions | BF 1 | BF 2/3 | CMULT | BF 4/5/6 |
| Area | 38556 μm^2 | 46119 μm^2 | 47459 μm^2 | 40547 μm^2 |
| Avg. power | 1.2mW | 1.5mW | 1.2mW | 1.7mW |
| Avg. current | 1mA | 1.25mA | 1mA | 1.4mA |

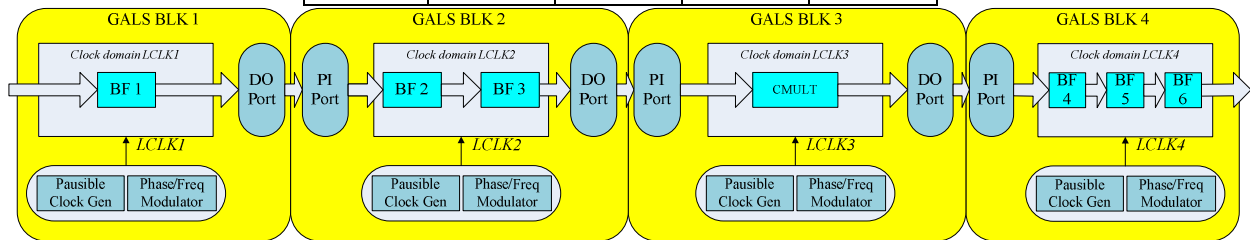


Figure 10: Architecture of 64-point pipelined GALS FFT processor

As a pipelined design, all the 4 GALS blocks were configured to run at the same frequency. To spread the switching activities of GALS blocks, a skew as long as a quarter of clock period is inserted in between each pair of cascaded GALS blocks. The triangular frequency modulator was applied in the experiment.

Direct simulations on the radiated emission require accurate models of parasitic inductance as well as capacitance of both the packaging structures and the board power traces. However, considering that the far-field radiated strength either increases linearly with the current frequency in common-mode or with square of the current frequency in differential-mode, hence current spectrum can be investigated instead in terms of EMI noise reduction. Figure 10 presents the amplitude spectrum of GALS FFT processor with above mentioned phase and frequency modulation by GalsEmulator. Significant reduction of EMI noise is shown over a large spectral region. Reduction of noise in the order of 20 dB (or higher) is expected.

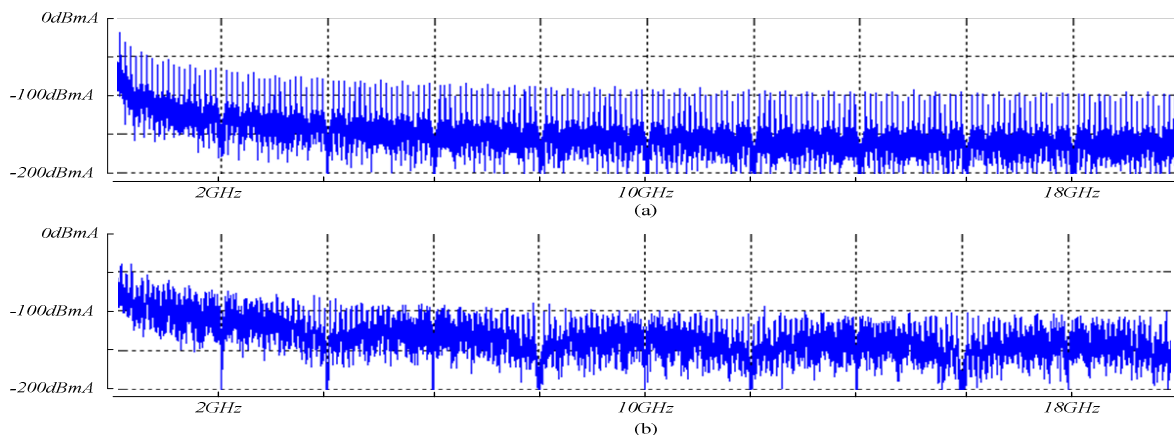


Figure 11: Amplitude spectrum of supply current (a) without clock modulation (b) with clock modulation ($f_c=100\text{MHz}$). EMI noise is significantly reduced



Hence, the experimental demonstration of the potential benefits of GALS architectures with respect to EMI reduction in complex systems shall be followed as one of the key objectives in the target design, due to the high expected benefits of the GALS approach. This benefit can be demonstrated independent of NoC solution.

In a complex design such as the baseband processor Tx block, the main challenge is to generate general purpose decoding accelerator that covers different techniques/modes/data rates in the system. This means that ideally, a fully reconfigurable system architecture with high data throughput would be desirable. However, it is noted that reconfigurability (such as provided by a NoC architecture) and high throughput usually generate conflicting demands that complicate the performance trade-off.

Due to the inherent difficulty of attacking a large and complex GALS design that is in the same time fully reconfigurable, the focus has to be on particular decoding technique, and limited set of the data rates and system should be optimized accordingly. In this case, point-to-point GALS emerges as the most suitable solution since:

- Point-to-point processor architecture is optimal for the specific design in question.
- Achieving planned throughput implies large NoC structures that should provide sufficient QoS.
- Additional overhead introduced by NoC architecture is rather large and implies the final chip area much larger than planned, increasing the necessary cost planned for chip fabrication.
- Additional power consumption due to NoC is not negligible even though it is minimized with GALSification.
- In the process of GALSification, optimal partition size for GALS modules is dictated by the functionality of the processor design and is suboptimal for typical blocks used in NoC, i.e. optimal GALS modules are smaller than typical cores used in NoC architectures.
- Even though GALSification of all NoC building blocks is done according to project agreement, integration of NoC on the large and complex system such as OFDM processor brings a lot of additional hazards. Moreover, optimal GALSification of switches is still an open problem.

Due all these drawbacks, and despite the potential benefits concerning reconfigurability, our suggestion is that in case of large and complex design such as OFDM processor as target hardware accelerator, implementation of NoC is *not* beneficial.

Nevertheless, it was decided to implement additional NoC test structures and components that were developed during the course of the project, as part of the test chip realization.



3.3 ASSESSING SYNCHRONIZATION INTERFACES FOR GALS BASED NETWORK-ON-CHIP DESIGN

Traditional globally synchronous clocking circuits have become increasingly difficult to design with growing chip size, clock rates, relative wire delays and parameter variations. Additionally, high speed global clocks consume a significant portion of system power and lack the flexibility to independently control the clock frequencies of submodules to achieve high energy efficiency. The globally asynchronous locally synchronous (GALS) clocking style separates processing blocks such that each block is clocked by an independent clock domain and is an effective strategy to address global clock design challenges.

This new paradigm heavily impacts the architecture of the chip-wide communication infrastructure.

In fact, synchronizers are typically inserted between two connected network building blocks belonging to different clock domains. In practice, they break the switch-to-switch or the network interface-to-switch connections depending on the decisions about clock domain partitioning. However, there is typically no co-optimization of the synchronizer with the following/preceding network-on-chip (NoC) sub-module, therefore a significant latency, area and power overhead materializes. This design practice is hereafter denoted as the loose coupling of synchronization interfaces with the NoC. In contrast, the objective of this testchip is to prove that the tight integration of the synchronizer into the NoC switch can result into a novel architecture block taking care of synchronization but also of other tasks such as switch input buffering and flow control.

The consequent reuse of buffering resources for different purposes in turn leads to large energy savings that make a GALS NoC affordable at almost the same area and power cost of its synchronous counterpart. Moreover, by moving the synchronizer inside the switch (or the network interface), the communication latency at the clock domain boundary reduces to the ideal synchronization latency, which simplifies flow control and reduces its buffering requirements.

In order to prove the practical viability of this approach, this testchip considers several synchronization scenarios which are relevant for GALS NoC design, implements both loosely coupled and tightly coupled synchronization interfaces and aims at a comparison between them. For the sake of comparison, baseline synchronous interfaces are considered as well. The final objective is twofold. On one hand, the testchip intends to prove that the developed synchronization interfaces can be safely used for GALS NoC design and their implementation is robust enough to on-chip noise sources and to parameter variations. On the other hand, the testchip paves the way for a post-silicon validation of the tightly coupled design philosophy for synchronization interfaces.

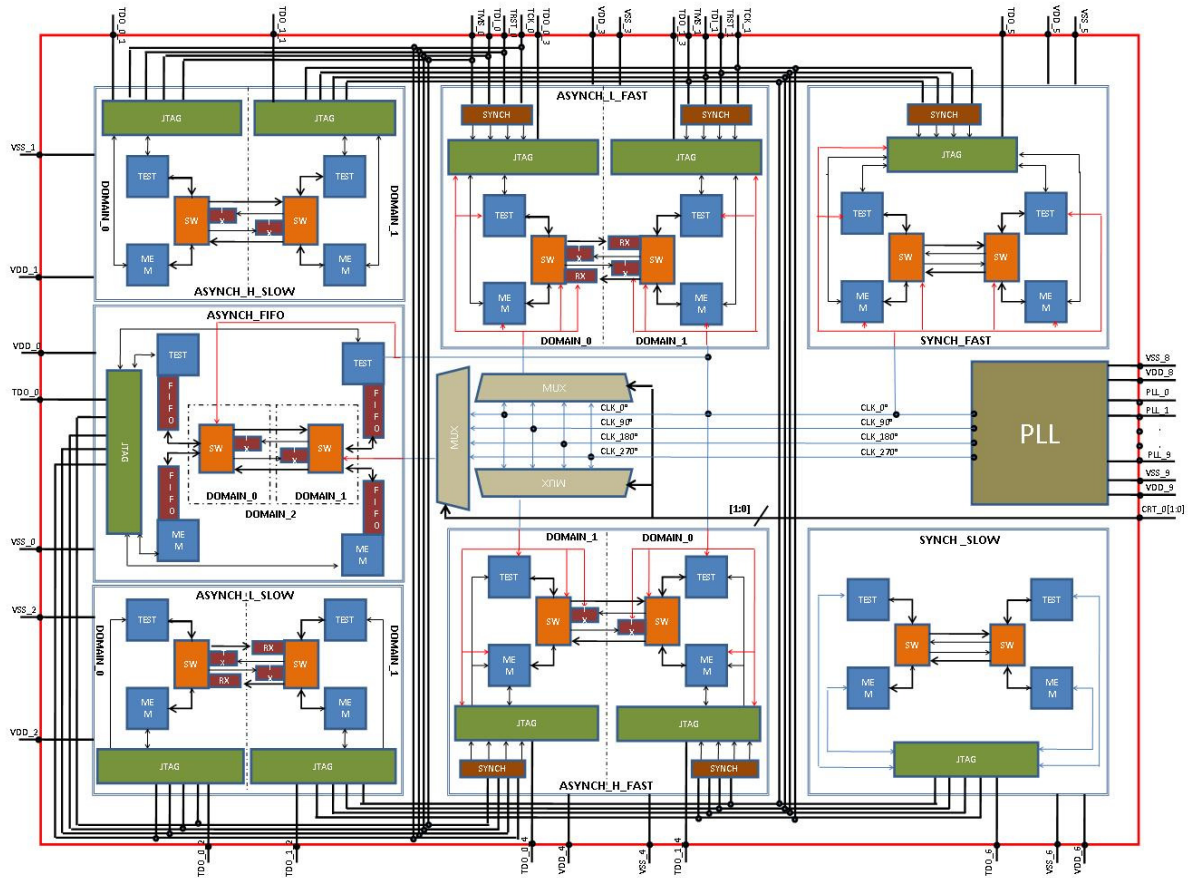
The proposed design contains seven NoC (sub-)systems. Each sub-system implements a 2-ary 1-mesh topology with 4 cores. Since our design is very modular, in every sub-system each network switch is connected to two IP cores (a processor core and a memory core) and each core is programmable and connected to a JTAG interface for I/O. Figure 12 shows the block diagram of the entire design. Our goal is to implement different sub-systems comparing them in terms of latency, area and power results. In order to be able to test the design at low and high frequencies, two versions of every synchronization scenario are implemented (denoted as *fast* and *slow* variants). In the *slow* variant the clock signals come from external pins; on the contrary, in the *fast* variant the clock comes out of the PLL. The PLL generates four clocks (at



250MHz) with different phases and it drives the fast platforms exploiting the multiplexers which select one of the four PLL clock phases.

The peculiarity of each sub-system can be described as follows:

- “**Synch_Slow**” demonstrates a fully synchronous NoC communication at low frequency. The platform is composed by 1 JTAG, 2 switches, 2 memories and 2 testers and it is clocked by a unique external clock.
- “**Synch_Fast**” implements a fully synchronous NoC communication at high frequency. The platform is composed by 1 JTAG, 2 switches, 2 memories, 2 testers and 1 synch. The synch implements a brute force synchronizer and it is used to synchronize the JTAG inputs with the fast and real platform clock which comes from the PLL.
- “**Asynch_L_Slow**” has two clock domains having the same low clock frequency but *unknown phase offset (mesochronous synchronization)*. This platform uses loosely coupled (i.e., not embedded into the switches) mesochronous synchronizers (RX and TX) placed in the bidirectional link next to the switches. The RX module synchronizes the data while the TX module synchronizes the flow control signal absorbing the phase offset between the domains. The platform is composed by 2 JTAG, 2 switches, 2 memories, 2 testers, 2 loose RX and 2 loose TX.
- “**Asynch_L_Fast**” has two clock domains having the same high clock frequency but *unknown phase offset*. The domain0 is driven by PLL clock with 0 skew while the domain1 is driven by the output of the multiplexer. This platform uses loosely coupled mesochronous synchronizers (RX and TX) placed into the bidirectional link next to the switches. The platform is composed by 2 JTAG, 2 switches, 2 memories, 2 testers, 2 loose RX, 2 loose TX and 2 synch.
- “**Asynch_H_Slow**” has two clock domains having the same low clock frequency but unknown phase offset. This platform has two tightly coupled mesochronous synchronizers integrated into the switch input stage and two TX synchronizers into the bidirectional link. The platform is composed by 2 JTAG, 2 switches, 2 memories, 2 testers, 2 tight RX and 2 loose TX.
- “**Asynch_H_Fast**” has two clock domains having the same high clock frequency but unknown phase offset. This platform has two tightly coupled mesochronous synchronizers integrated into the switch input stage and two TX synchronizers into the bidirectional link. The domain0 is driven by PLL clock with 0 skew while the domain1 is driven by the output of the multiplexer. The platform is composed by 2 JTAG, 2 switches, 2 memories, 2 testers, 2 tight RX, 2 loose TX and 2 synch.
- “**Asynch_Fifo**” has three clock domains: domain0 and domain1 have the same high clock frequency but unknown phase offset while domain3 has the low clock frequency. The cores and the switches have true independent clocks with distinct frequency and offset. Since the platform has different clock frequencies, two tightly coupled dual-clock FIFOs are integrated into the switch input stage and two additional loosely coupled dual-clock FIFOs are instantiated next to the cores in order to synchronize the information coming from the cores and the switches, respectively. Moreover the platform has two tightly coupled mesochronous synchronizers integrated into the switch input stage and two TX synchronizers into the bidirectional link. The domain0 is driven by PLL clock with 0 skew while the domain1 is driven by the output of the multiplexer. The platform is composed by 2 JTAG, 2 switches, 2 memories, 2 testers, 2 tight RX, 2 loose TX, 2 tight FIFO, 2 loose FIFO and 2 synch.





4 FINAL DISCUSSION AND CONCLUSION

Considering the specific application area of wireless telecommunications, and taking into account the various constraints on the overall system topology, it was decided to focus on a point-to-point GALS solution. In particular, the target design shall be utilized to demonstrate the low EMI properties of GALS architecture, as applied in the point-to-point topology. Hence, the target design, Moonrake Tx processor, has been chosen as a high-performance, point-to-point GALS system demonstrator.

There are two significant advantages of larger design point-to-point (OFDM processor), despite the potential benefits of GALS NoC implementation and potential impact that this kind of demonstration can have:

- Since there is only one demonstrator target design in the project, the safer approach is to use structures that are already tested and proven.
- Since the main goal of the whole project is to demonstrate potential benefits of GALS designs (especially in terms of EMI reduction), implementation of such a large GALS design that has never before been demonstrated will have the larger potential impact and significance in the research community (and industry) than smaller GALS NoC implementation that has already demonstrated implementations.

In order to utilize the design expertise that has been gained and to experimentally demonstrate the potential benefits, it was also decided to implement additional NoC test structures and components that were developed during the course of the project, as part of the test chip realization. The particular synchronization structures to be targeted in this context are described in Section 3.3.