



Deliverable – D25

Report on the crossbenchmarking results of fully synchronous vs GALS NoC implementations and GALS-oriented Interfaces for NoC design

| | |
|---|--|
| 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: | 30.10.2008. |
| Contractual Date of Delivery to the EC: | 1. Dec. 2010 |
| Actual Date of Delivery to the EC: | 1. Dec. 2010 |
| Author(s): | Daniele Ludovici (UNIBO), Alessandro Strano (UNIBO), Davide Bertozzi (UNIBO), Luca Benini (UNIBO), Christoph Heer (IFX), Milos Stanisavljevic (EPFL) |
| Participant(s): | UNIBO, IFX, EPFL |
| Work Package: | mainly WP6, but also WP5 |
| Security: | Public |
| Nature: | Report |
| Version: | 1 |
| Total number of pages: | 40 |

Abstract:

WP6 of the Galaxy project has developed not just a synchronizer-based GALS NoC concept, but rather a mature synchronization technology, consisting of layout-proven robust interfaces and suitable design technology enhancements. By leveraging this technology, this deliverable presents crossbenchmarking results between a fully synchronous NoC and a mesochronous NoC for use in GALS systems. In fact, the main objective is to use one (or both) of them together with dual-clock FIFOs at the network boundary to decouple network speed from IP core speed. The outcome of this comparison framework is key for industrial designers to establish a roadmap toward the relaxation of synchronization assumptions in MPSoCs in a cost-effective way. Moreover, this work aims at highlighting tedious architecture and design technology issues that need a further optimization effort to foster widespread adoption of network-based GALS systems in the embedded computing domain. Overall, we find that considering carefully engineered mesochronous NoC technology instead of synchronizer concept schemes or silicon-unaware synchronization interfaces is a distinctive feature of this work. Building on this, the paper is able to provide a realistic assessment and quantification of fundamental physical design concerns such as variability robustness or power efficiency of hierarchical clock tree synthesis.



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

Keyword list: GALS network-on-chip; dual-clock FIFO; mesochronous NoC; hierarchical clock tree synthesis; variability robustness



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

| <i>Function</i> | <i>Responsibility</i> | <i>Date</i> | <i>Signature</i> |
|---------------------|-----------------------------------|-------------|------------------|
| Written by: | Daniele Ludovici, Davide Bertozzi | 30/11/2010 | |
| Checked by: | | | |
| Approved by: | | | |

Reserved to EC

| | | | |
|---------------------|--|--|--|
| Approved by: | | | |
|---------------------|--|--|--|





GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

CHANGE RECORDS

| <i>ISSUE</i> | <i>DATE</i> | <i>§ : CHANGE RECORD</i> | <i>AUTHOR</i> |
|--------------|-------------|--------------------------|---------------|
| | | | |
| | | | |
| | | | |
| | | | |



BIBLIOGRAPHIC RECORD

| | |
|---|--|
| Project Number: | 214364 GALAXY |
| Project Title: | GALAXY |
| Deliverable Type: | Report |
| Deliverable Number: | D25 |
| Contractual Date of Delivery: | 1. Dec. 2010 |
| Actual Date of Delivery: | 1. Dec. 2010 |
| Title of Deliverable: | <i>Report on the crossbenchmarking results of fully synchronous vs GALS NoC implementations and GALS-oriented Interfaces for NoC design</i> |
| Work package contributing to the Deliverable: | Mainly WP6, but also WP5 |
| Authors: | Daniele Ludovici (UNIBO), Alessandro Strano (UNIBO), Luca Benini (UNIBO), Christoph Heer (IFX), Milos Stanisavljevic (EPFL), Davide Bertozzi (UNIBO) |
| Abstract | <p>WP6 of the Galaxy project has developed not just a synchronizer-based GALS NoC concept, but rather a mature synchronization technology, consisting of layout-proven robust interfaces and suitable design technology enhancements. By leveraging this technology, this deliverable presents crossbenchmarking results between a fully synchronous NoC and a mesochronous NoC for use in GALS systems. In fact, the main objective is to use one (or both) of them together with dual-clock FIFOs at the network boundary to decouple network speed from IP core speed. The outcome of this comparison framework is key for industrial designers to establish a roadmap toward the relaxation of synchronization assumptions in MPSoCs in a cost-effective way. Moreover, this work aims at highlighting tedious architecture and design technology issues that need a further optimization effort to foster widespread adoption of network-based GALS systems in the embedded computing domain. Overall, we find that considering carefully engineered mesochronous NoC technology instead of synchronizer concept schemes or silicon-unaware synchronization interfaces is a distinctive feature of this work. Building on this, the paper is able to provide a realistic assessment and quantification of fundamental physical design concerns such as variability robustness or power efficiency of hierarchical clock tree synthesis.</p> |
| Keywords | GALS network-on-chip; dual-clock FIFO; mesochronous NoC; hierarchical clock tree synthesis; variability robustness |
| Confidentiality Level | Public |



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

| | |
|------------------------|------------------------|
| Name of Client: | EC |
| Distribution List: | GALAXY, EC, internet |
| Authorised by: | |
| Issue: | 1 |
| Document ID: | D25 |
| Total Number of Pages: | 40 |
| Contact Details: | dbertozzi@ing.unife.it |



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

TABLE OF CONTENTS

| | |
|---|-----------|
| CHANGE RECORDS | 4 |
| BIBLIOGRAPHIC RECORD..... | 5 |
| TABLE OF CONTENTS | 7 |
| LIST OF FIGURES | 8 |
| LIST OF TABLES..... | 8 |
| 1 INTRODUCTION | 9 |
| 2 STATE-OF-THE-ART | 11 |
| 2.1 REFERENCE DOCUMENTS..... | 12 |
| 3 TARGET GALS ARCHITECTURE..... | 15 |
| 3.1 GENERAL OVERVIEW | 15 |
| 3.2 THE MOTIVIATION FOR MESOCHRONOUS NOCS..... | 17 |
| 3.3 HYBRID COUPLED MESOCHRONOUS SYNCHRONIZER | 19 |
| 3.4 PLATFORMS SYNTHESIS AND PLACE AND ROUTE..... | 21 |
| 4 POST-LAYOUT EXPERIMENTAL RESULTS | 24 |
| 4.1 AREA AND WIRING OVERHEAD | 24 |
| 4.2 SYSTEM-LEVEL POWER ANALYSIS..... | 24 |
| 4.3 CLOCK TREE POWER ANALYSIS..... | 27 |
| 5 VARIABILITY ROBUSTNESS | 30 |
| 6 PERFORMANCE CONSIDERATIONS | 34 |
| 7 PORTING TO THE 40NM TECHNOLOGY NODE..... | 35 |
| 8 CONCLUSIONS | 38 |



LIST OF FIGURES

| | |
|--|----|
| Figure 1: The NoC is split in multiple VFIs | 15 |
| Figure 2: The NoC is a global clock domain..... | 16 |
| Figure 3: Mesochronous NoC for GALS systems..... | 16 |
| Figure 4: Hierarchical Clock Tree Synthesis Process..... | 17 |
| Figure 5: area occupancy of different synchronization schemes with respect to the vanilla switch. | 18 |
| Figure 6: Hybrid Coupled Mesochronous Synchronizer..... | 20 |
| Figure 7: Post-place and Route screenshot of a 4x4 Platform..... | 23 |
| Figure 8: Area and wiring overhead normalized..... | 24 |
| Figure 9: Power consumption with low network load | 25 |
| Figure 10: Power consumption with high network load..... | 26 |
| Figure 11: Power consumption under a realistic multimedia application..... | 27 |
| Figure 12: Power of the top level clock tree as a function of required skew. | 28 |
| Figure 13: Skew tolerance with slack (enforced by the physical synthesis tool)..... | 30 |
| Figure 14: Skew tolerance with no slack..... | 31 |
| Figure 15: Variability-induced percentage deviations of maximum speeds of the designs under test with respect to nominal ones..... | 32 |
| Figure 16: Relative area ratios across technology nodes..... | 35 |
| Figure 17 Relative critical path delay ratios across technology nodes..... | 36 |
| Figure 18 Area breakdown for the loosely coupled switch across technology nodes..... | 36 |

LIST OF TABLES

| | |
|--|----|
| Table 1: Clock tree specification file..... | 22 |
| Table 2: Top clock tree power for a 64 cores system..... | 28 |



1 INTRODUCTION

Networks-on-chip (NoCs) are proving capable of easing the communication bottleneck arising in multi-core computing platforms [IH07],[VA08],[FL09],[WE07], thus overcoming the fundamental performance, power and physical design limitations of shared and multi-layer busses.

There is today little doubt on the fact that a high-performance and cost-effective NoC can only be designed in 45nm and beyond under a relaxed synchronization assumption [BO09],[FL09]. One effective method to address this issue is through the use of globally asynchronous and locally synchronous (GALS) architectures, where the chip is partitioned into multiple independent voltage and frequency domains. Each domain is clocked synchronously while inter-domain communication is achieved through specific interconnect techniques and circuits [KR07].

The methodology of inter-domain communication is a crucial design point for GALS architectures. One approach currently experimented in GALS NoC prototypes consists of using purely asynchronous clockless handshaking for transferring data words across clock domains [DV05],[BS05]. A few chip demonstrators prove the viability of this solution [CL07],[CB10],[TV10], but they have not achieved widespread adoption of asynchronous NoCs in the industrial arena yet.

In fact, asynchronous handshaking techniques are complex and use unconventional circuits (e.g., Muller C-elements) usually unavailable in industrial technology libraries. Moreover, asynchronous logic is not well supported by CAD tools. In this context, the most effective solution found so far for actual chip fabrication and industry-relevant designs consists of implementing routers and GALS interfaces as hard macros using ad-hoc design styles [TV10]. The hard macro methodology is however more a way of working around the lack of proper design and verification tools and thus guaranteeing performance during physical implementation rather than a way of optimizing area. In fact, this latter remains consistently larger than fully synchronous NoC counterparts (1,8x in [TV10]).

What is currently missing in the open literature as well as in the industrial prototyping experience is a mature GALS NoC architecture making use of source synchronous communication techniques. This method achieves high efficiency by obtaining an ideal throughput of one data word per source clock cycle with a design style which is more easily compatible with common standard cell design flows. In spite of a few early works taking this approach [PC08],[TT10], this GALS design style has not been consistently brought to maturity over time, therefore reducing source synchronous communication to a nice concept with only limited relevance for real-life designs. For instance, the area and latency overhead associated with the use of synchronizers has never been tackled in a systematic way, and even advanced research prototypes from industry like the Intel Polaris chip [VA08] live with such an overhead. As a consequence, source synchronous communication has never truly evolved from a concept to a mature technology.

The work carried out so far mainly in WP6 of the Galaxy project and reflected in the open literature by papers such as [LB09] [LS09] [SL10] [LS10] has tried to bridge this gap and to develop synchronizer-based GALS NoC technology. In particular, design techniques merging synchronizers with network building blocks (named the *tightly coupled design style*) have proved area, power and performance efficient with respect to loosely coupled solutions, where synchronizers are placed as external blocks to NoC switches. All previous work concerns



architecture design space exploration and quality metrics assessment of synchronizer-based communications at the level of the inter-switch link (see tasks 6.1, 6.2 and 6.4 of WP6).

The work described in this document builds on these milestones and moves a step forward by taking the network-level perspective. While the migration from fully synchronous parallel systems to GALS systems with voltage/frequency decoupling between IP cores is taken as a matter of fact in this work, there are significant GALS NoC architecture variants the designer can still choose from. The first one consists of placing NoC switches in the clock domains of the IP cores they are connected with. In contrast, an alternative solution consists of inferring the on-chip network as an independent clock domain, disjoint from those of the IP cores. In this scenario, dual-clock FIFOs need to be instantiated only at the network boundary, since the network is synchronized by a single and independent clock signal. The homogeneous performance of NoC switches, the fewer amount of dual-clock FIFOs required and the possibility to have an always on system interconnect fabric make this solution more attractive to this work. However, the feasibility and efficiency of this solution is now mainly on burden of the physical designer. In fact, he has again to deal with a large synchronous clock domain (the NoC itself) distributed throughout the entire chip.

A workaround for this problem consists of inferring the network as a set of mesochronous domains, instead of a global synchronous domain, yet retaining a globally synchronous perspective of the network itself. The granularity of a mesochronous domain can be as fine as a NoC switch, which is the case considered in this work. The communication between neighboring switches is then mesochronous as the top-level clock tree might not be equilibrated. This brings the additional advantage that mesochronous synchronizers are typically more lightweight than dual-clock FIFOs for use in switch-to-switch links.

This deliverable leverages mature mesochronous communication technology to perform a comprehensive crossbenchmarking of a mesochronous NoC with a fully synchronous NoC for use in a GALS system. Both networks share the same baseline MPSoC-oriented NoC architecture for the sake of fair comparison. The tightly coupled design principle is followed for mesochronous links, so that their unique optimization opportunities in the NoC domain are fully exploited. The work relies on actual implementations on a 65nm industrial technology library and reports preliminary synthesis results in Infineon 40nm technology that pave the way for testchip fabrication (documented in deliverable D27).

The outcome of this crossbenchmarking framework is key for industrial designers to establish a roadmap toward the relaxation of synchronization assumptions in MPSoCs in a cost-effective way. Moreover, this work aims at highlighting tedious architecture and design technology issues that need a further optimization effort to foster widespread adoption of network-based GALS systems in the embedded computing domain.

Overall, we find that considering carefully engineered mesochronous NoC technology instead of synchronizer concept schemes or silicon-unaware synchronization interfaces is a distinctive feature of this work, which the Galaxy project made possible. Building on this, the deliverable is able to provide a realistic assessment and quantification of fundamental physical design concerns such as variability robustness or power efficiency of hierarchical clock tree synthesis. The work provides useful guidelines for those industrial designers currently committed to the development of next generation NoC-based MPSoCs.



2 STATE-OF-THE-ART

Chip synchronization in the nanoscale regime is an urgent challenge orthogonal to all parallel computing platforms, even in the SoC domain [MS07]. Since clock tree design and tuning for variability robustness comes at a non-negligible cost [NK09],[TZ05],[KS08],[GS08]. The work described in this document investigates how to live with uncertainties by means of proper architectural support in the on-chip interconnect. The GALS paradigm has been frequently experimented by using asynchronous logic [DV05],[BS05]. A chip dedicated to flexible baseband processing for 3G/4G wireless telecommunication applications and making use of an asynchronous NoC is described in [CL07],[CB10]. However, currently the intricacy of asynchronous design and its poor CAD tool support makes the design of hard macros with ad hoc design techniques [MN06] the only viable solution for industrial exploitation [TV10]. This penalizes flexibility and is not able to cut down on area overhead of timing-robust asynchronous realizations. The practical viability of synchronizer-based GALS networks has been proven in [PC08], where the hierarchical clock tree synthesis technique for such systems is detailed. Since there is no parametric exploration there, it is not possible to validate a common opinion (for instance, reported in [JA10],[ML06],[VI08]) that large on-chip power consumption can be reduced by replacing the balanced clock tree with a GALS clocking scheme which only guarantees minimal clock skew within the local processing elements. This common sense claim has remained unproved so far, without any precise quantification.

A circuit switched source synchronous GALS link is described in [TT10], making use of long distance interconnect paths. It is then experimented on a 65nm reconfigurable NoC for an heterogeneous GALS many-core platform. However, there is no comparison whatsoever with alternative clocking styles and/or implementations.

A mesochronous link is integrated within a Multi Processor tiled architecture based on a Network-on-Chip communication backbone on a CMOS 65 nm technology in [VI08]. The work builds on a full-duplex link architecture illustrated in [MF07] and on integrated flow control [SV08]. The baseline mesochronous synchronizer is instead proposed in [ML06]. However, the synchronizer is still an external module to the NoC. The same drawback is exposed by a different synchronization architecture, detailed in [VA08].

Our previous work in WP6 and resulting in papers [LB09] and [LS09] poses the foundations of a new design style where synchronizers are merged with NoC building blocks. The approach was applied to a mesochronous synchronizer and to a dual-clock FIFO [SL10]. Timing constraints as a function of varying architecture and physical design parameters were analyzed in [LS10]. We are now in a position to perform the first crossbenchmarking framework between a fully synchronous NoC and a mesochronous NoC for use in GALS systems as we will describe throughout this deliverable.



2.1 REFERENCE DOCUMENTS

| Ref. | Document Title |
|--------|---|
| [HS10] | C. Hernandez, F. Silla, and J. Duato, <i>A Methodology for the Characterization of Process Variation in NoC Links</i> , Proceedings of the Design, Automation and Test in Europe (DATE), 2010. |
| [HR10] | C. Hernandez, A. Roca, F. Silla, J. Flich, and J. Duato, <i>Improving the Performance of GALS-based NoCs in the Presence of Process Variation</i> , Proceedings of the International Symposium on Networks-on-Chip (NOCS), 2010. |
| [MT07] | M. Mosin, R. Tamer, W. Xiang, A. Adnanm and M. Yehia, <i>Provisioning On-Chip Networks under Buffered RC Interconnect Delay Variations</i> , Proceedings of the International Symposium on Quality Electronic Design, 2007. |
| [GS08] | M.R. Guthaus, D. Sylvester, R. B. Brown, <i>Clock Tree Synthesis with Data-Path Sensitivity Matching</i> , Proceedings of the 2008 Asia and South Pacific Design Automation Conference, 2008. |
| [ITRS] | ITRS roadmap, online at http://www.itrs.net |
| [KS08] | V. Khandelwal, A. Srivastava, <i>Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation</i> , in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 27, Issue 4, April 2008. |
| [NK09] | K. Nagaraj, S. Kundu, <i>A Study on Placement of Post Silicon Clock Tuning Buffers for Mitigating Impact of Process Variation</i> , Proceedings of the Design, Automation and Test in Europe (DATE), 2009. |
| [TZ05] | J. Tsai, L. Zhang, C. Chen, <i>Statistical Timing Analysis Driven Post-Silicon-Tunable Clock-Tree Synthesis</i> , Proceedings of ICCAD, 2005. |
| [MN06] | A. J. Martin, M. Nystrom, <i>Asynchronous Techniques for System-on-Chip Design</i> , Proceedings of the IEEE, vol.94, no.6, pp.1089-1120, 2006. |
| [SA05] | S. Stergiou, F. Angiolini, S. Carta, L. Raffo, D. Bertozzi, G. De Micheli, <i>xpipes Lite: a Synthesis Oriented Design Library for Networks on Chips</i> , Proceedings of the Design Automation and Test in Europe Conference, pp. 1188-1193, 2005. |
| [KL10] | M. R. Kakoei, I. Loi, L. Benini, <i>A New Physical Routing Approach for Robust Bundled Signaling on NoC Links</i> , Proceedings of the 20th Great Lakes Symposium on VLSI, pp.3-8, 2010. |
| [LS10] | D. Ludovici, A. Strano, G. N. Gaydadjiev, L. Benini, D. Bertozzi, <i>Design Space Exploration of a Mesochronous Link for Cost-Effective and Flexible GALS NOCs</i> , Proceedings of Design, Automation and Test in Europe (DATE), pp. 679-684, 2010. |



| | |
|--------|---|
| [LS09] | D. Ludovici, A. Strano, D. Bertozzi, <i>Architecture design principles for the integration of synchronization interfaces into Network-on-Chip switches</i> , Proceedings of the 2nd International Workshop on Network on Chip Architectures, pp. 31--36, 2009. |
| [LB09] | D. Ludovici, A. Strano, D. Bertozzi, L. Benini, G. N. Gaydadjiev, <i>Comparing tightly and loosely coupled mesochronous synchronizers in a NoC switch architecture</i> , Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip, pp.244-249, 2009. |
| [SL10] | A. Strano, D. Ludovici, D. Bertozzi, <i>A Library of Dual-Clock FIFOs for Cost-Effective and Flexible MPSoCs Design</i> , Proceedings of the International Conference on Embedded Computer Systems: Architectures, MOdeling and Simulation (SAMOS), 2010. |
| [JA10] | T. N. K. Jain, <i>Asynchronous Bypass Channels: Improving Performance for Multi-Synchronous NoCs</i> , NOCS 2010, pp. 51-58, 2010. |
| [SV08] | S. Saponara, F. Vitullo, R. Locatelli, P. Teninge, M. Coppola, L. Fanucci, <i>LIME: a Low-Latency and Low-Complexity On-Chip Mesochronous Link with Integrated Flow Control</i> , Proceedings of the Euromicro Conference on Digital System Design (DSD), pp. 32-35, 2008. |
| [MS07] | D. Mangano, A. Scandurra, C. Pistrutto, <i>Relieving Physical Issues in New NoC-based SoCs</i> , Proceedings of the International Conference on Nano-Networks, 2007. |
| [MF07] | D. Mangano, G. Falconeri, C. Pistrutto, A. Scandurra, <i>Effective Full-Duplex Mesochronous Link Architecture for Network-on-Chip Data-Link Layer</i> , Proceedings of the Euromicro Conference on Digital System Design (DSD), pp. 519-526, 2007. |
| [VI08] | F. Vitullo et al., <i>Low-Complexity Link Microarchitecture for Mesochronous Communication in Networks-on-Chip</i> , IEEE Trans. on Computers, Vol.57, issue 9, pp.1196-1201, 2008. |
| [CMP] | Circuits Multi-Projects, Multi-Project Circuits, http://cmp.imag.fr |
| [ML06] | D. Mangano, R. Locatelli, A. Scandurra, C. Pistrutto, M. Coppola, L. Fanucci, F. Vitullo, D. Zandri, <i>Skew Insensitive Physical Links for Network on Chip</i> , 1st International Conference on Nano-Networks (NANO-NET 2006), Proceedings of the 1st International Conference on Nano-Networks (NANO-NET), 2006. |
| [TT10] | A. T. Tran, D. N. Truong, B. Baas, <i>A Reconfigurable Source-Synchronous On-Chip Network for GALS Many-Core Platforms</i> , IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol.29, no.6, 2010. |
| [PC08] | I. M. Panades, F. Clermidy, P. Vivet, A. Greiner, <i>Physical Implementation of the DSPIN Network-on-Chip in the FAUST Architecture</i> , Proceedings of the International Symposium on Networks-on-Chip (NOCS), pp. 139-148, 2008. |



| | |
|--------|---|
| [WE07] | D. Wentzlaff et al., <i>On-Chip Interconnection Architecture of the Tile Processor</i> , IEEE Micro, vol.27, no.5, pp. 15-31, 2007. |
| [IH07] | D. A. Iltzky, J. D. Hoffman, A. Chun and B. P. Esparza, <i>Architecture of the Scalable Communications Core's Network on Chip</i> , IEEE Micro, vol.27, Issue 5, pp.62 - 74, 2007. |
| [CL07] | F. Clermidy, R. Lemaire, X. Popon, D. Ktenas, Y. Thonnart, <i>An Open and Reconfigurable Platform for 4G Telecommunication: Concepts and Application</i> , Proceedings of the Euromicro Conference on Digital System Design (DSD), pp. 62-74, 2009. |
| [CB10] | F. Clermidy, C. Bernard, R. Lemaire, J. Martin, I. Miro-Panades, Y. Thonnart, P. Vivet, N. Wehn, <i>A 477mW NoC-based Digital Baseband for MIMO 4G SDR</i> , Proceedings of the Internation Solid State Circuits (ISSC), pp. 278-279, 2010. |
| [TV10] | Y. Thonnart, P. Vivet, F. Clermidy, <i>A Fully-Asynchronous Low-Power Framework for GALS NoC Integration</i> , Proceedings of the Design, Automation and Test in Europe (DATE), pp. 33-38, 2010. |
| [KR07] | M. Krstic et al., <i>Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook</i> , IEEE Design and Test of Computers, vol. 24, no. 5, pp. 430-441, Sept. 2007. |
| [VA08] | S. Vangal et al., <i>An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS</i> , IEEE Journal of Solid-State Circuits, Vol.43, Issue 1, pp.29-41,2008. |
| [FL09] | E. Flamand, <i>Strategic Directions Towards Multicore Application Specific Computing</i> , Proceedings of the Design, Automation and Test in Europe (DATE), pp. 1266, 2009. |
| [BO09] | S. Borkar, <i>Design Perspectives on 22nm CMOS and Beyond</i> , Proceedings of the Design Automation Conference (DAC), 2009. |
| [DV05] | R. Dobkin, V. Vishnyakov, E. Friedman, R. Ginosar, <i>An Asynchronous Router for Multiple Service Levels Networks on Chip</i> , Proceedings of ASYNC, pp. 44-53, 2005. |
| [BS05] | T. Bjerregaard, J. Sparso, <i>A Router Architecture for Connection-Oriented Service Guarantees in the MANGO Clockless Network-on-Chip</i> , Proceedings of the Design, Automation and Test in Europe (DATE), pp. 1226-1231, 2005. |



3 TARGET GALS ARCHITECTURE

This section describes the platforms that will be compared throughout the remainder of this document.

3.1 GENERAL OVERVIEW

A GALS-based design style fits nicely with the concept of voltage and frequency islands (VFIs), which has been introduced to achieve fine-grain system-level power management and is currently driving the transition of most MPSoCs to GALS systems.

In these systems, if network components belong to the core's VFIs as in Figure 1, then performance of communication flows would be determined by the slowest domain crossed on the way to destination. Also, in case a VFI is shut down, global connectivity is jeopardized.

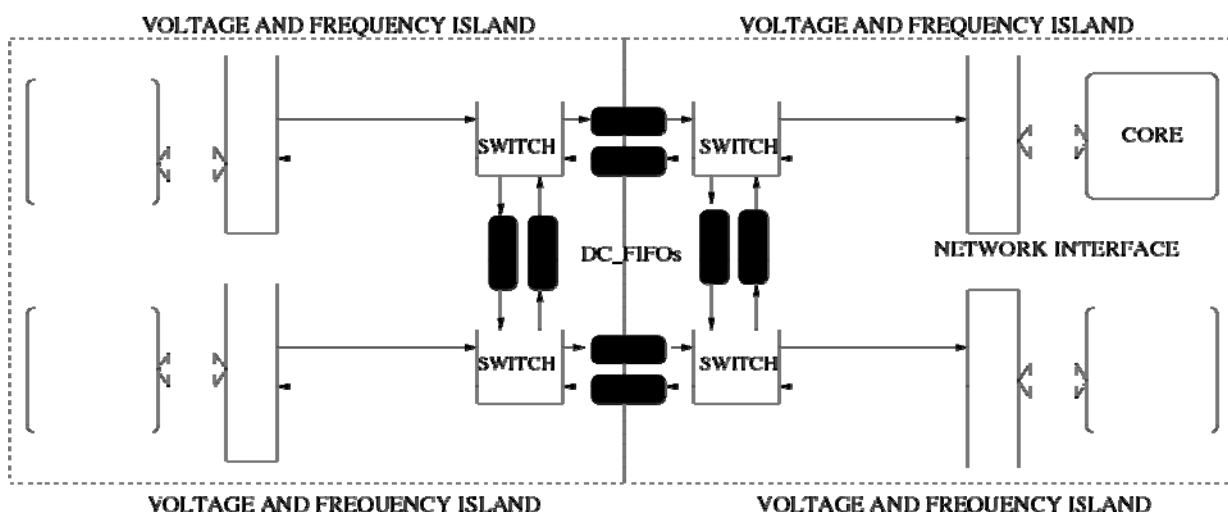


Figure 1: The NoC is split into multiple VFIs

An alternative solution is illustrated Figure 2, where the NoC lies in its independent VFI. This way, performance of the whole switching fabric would be homogeneous, with only boundary effects to take care of. Also, the network would be loosely coupled with the cores' VFIs, and each core/cluster of cores could be shutdown without any impact on global network connectivity.

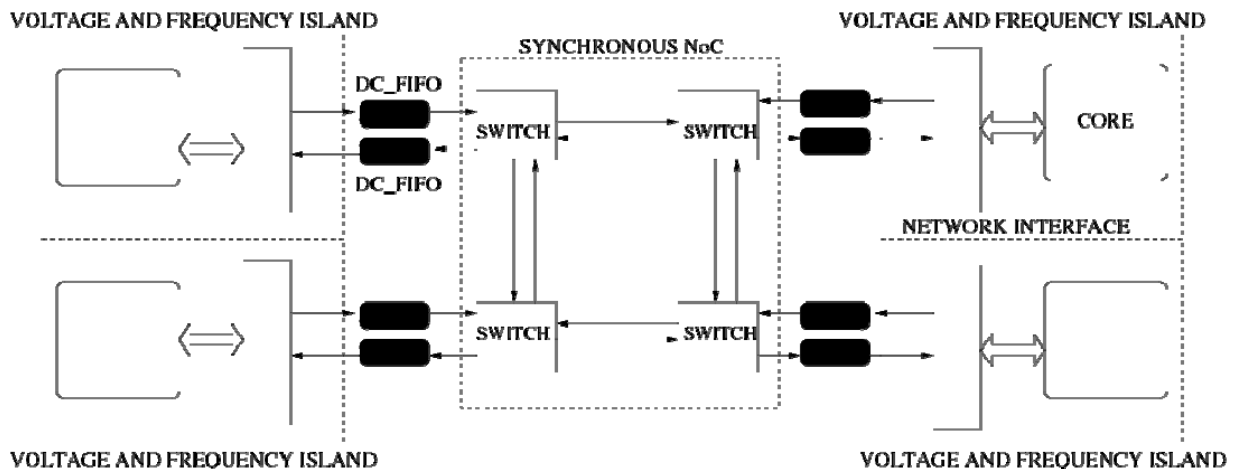


Figure 2: The NoC is a global clock domain.

The main issue with an independent NoC VFI consists of the feasibility of its clock tree. The reverse scaling of interconnect delays and the growing role of process variations are some of the root causes for this. Even though inferring a global clock tree for the entire network will still be feasible for some time, it will probably come at a significant power cost. Moreover, it is unclear when the difficulty of tightly and globally enforcing the skew constraint will truly become a roadblock.

However, a workaround for this problem does exist, as illustrated in Figure 3. The network could be inferred as a collection of mesochronous domains, instead of a global synchronous domain, yet retaining a globally synchronous perspective of the network itself. There are several methods to do this. One simple way is to go through a hierarchical clock tree synthesis process. Pictorially illustrated in Figure 4.

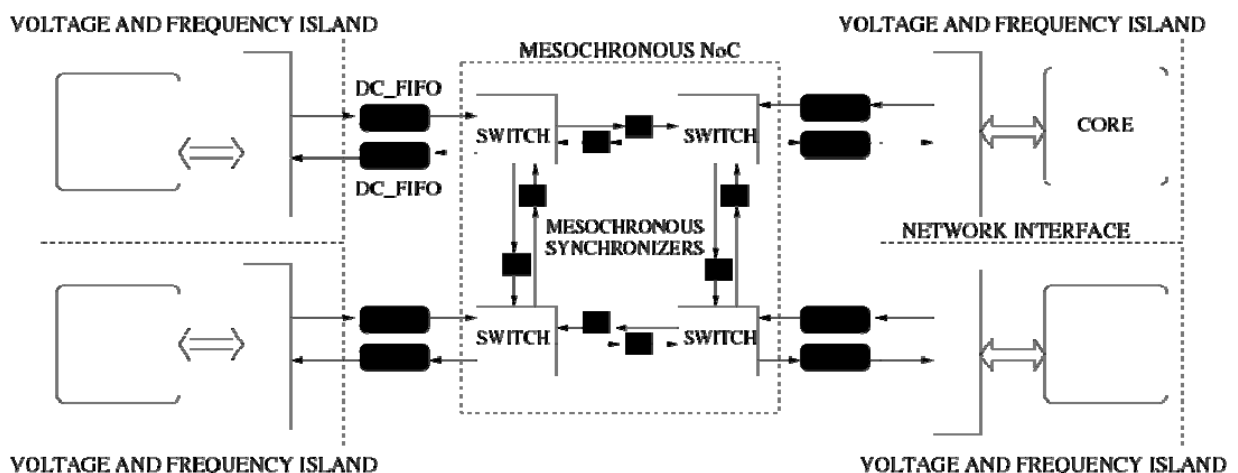


Figure 3: Mesochronous NoC for GALS systems.



In practice, a local clock tree is synthesized for each mesochronous domain, where the skew constraint is enforced to be as tight as in traditional synchronous designs. Then, a top-level clock tree is synthesized, connecting the leaf trees with the centralized clock source, with a

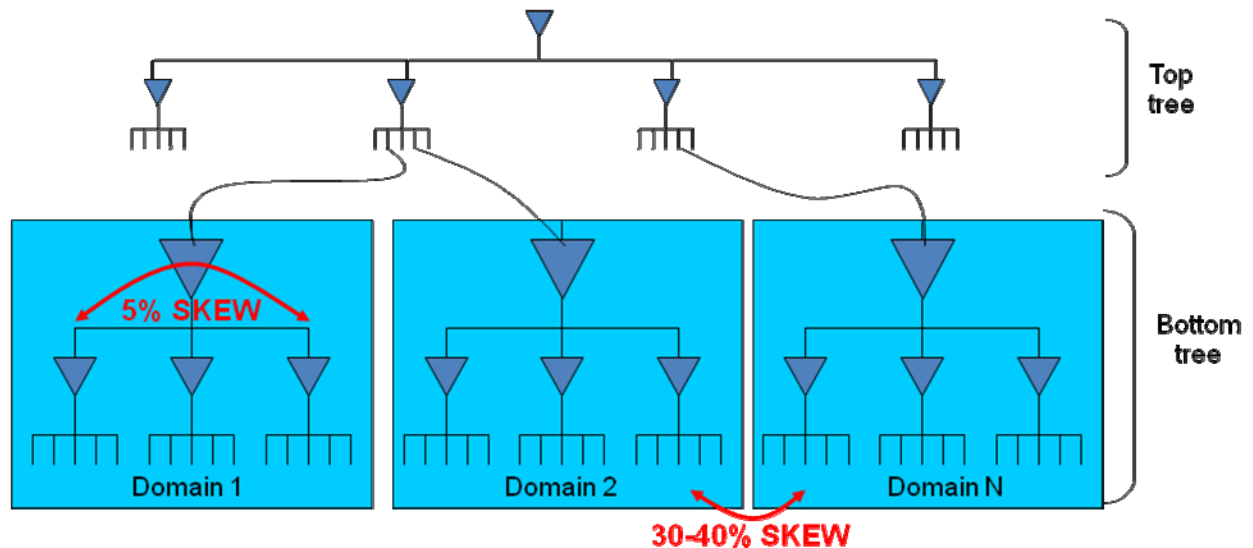


Figure 4: Hierarchical Clock Tree Synthesis Process.

very loose clock skew constraint. This way, many repeaters and buffers, which are used to keep signals in phase, can be removed, reducing power and thermal dissipation of the top-level clock tree. The granularity of a mesochronous domain can be as fine as a NoC switch block. The communication between neighboring switches is then mesochronous as the clock tree is not equilibrated, while the communications between switch and IP cores are fully asynchronous because they belong to different clock domains. Bi-synchronous FIFOs are therefore used to connect the network switches to the network interfaces of the cores, as showed in Figure 3.

3.2 THE MOTIVATION FOR MESOCHRONOUS NoCs

Even assuming that a fully synchronous NoC like in Figure 2 will be inadequate for forward-looking GALS systems, then the question is whether the best candidate is represented by the dual-clock FIFO-intensive architecture of Figure 1 or by the mesochronous NoC in Figure 3. We have already discussed above about the fact the mesochronous NoC provides a performance-homogeneous interconnect fabric and is more easy to handle in case power management strategies are implemented (the network could be always on at an affordable power cost, thus providing connectivity to those cores that are not shut down).

However, a synchronization paradigm based on a mesochronous NoC comes with additional advantages. First, it makes a conscious use of area/power-hungry dual-clock FIFOs, which end up being instantiated only at network boundaries. Instead, more compact mesochronous synchronizers are used inside the network, thus minimizing the cost for GALS technology.

In order to give a more quantitative estimation of such choice, let us compare a mesochronous interface with 3 slot buffers (thus providing $\pm 100\%$ skew tolerance) developed in task 6.1 with its analogous dual-clock FIFO developed in task 6.2 out of the baseline xpipes NoC components library. Among the baseline dual-clock FIFO options, the only interface that is



able to work without a performance decrement (full throughput condition) in a mesochronous scenario is that with 5 slots (see deliverable D13 and published results in [SB11]). The reason is that, other solutions with smaller buffer slots, are not able to interface two clock domains working at the same frequency while retaining a full throughput condition.

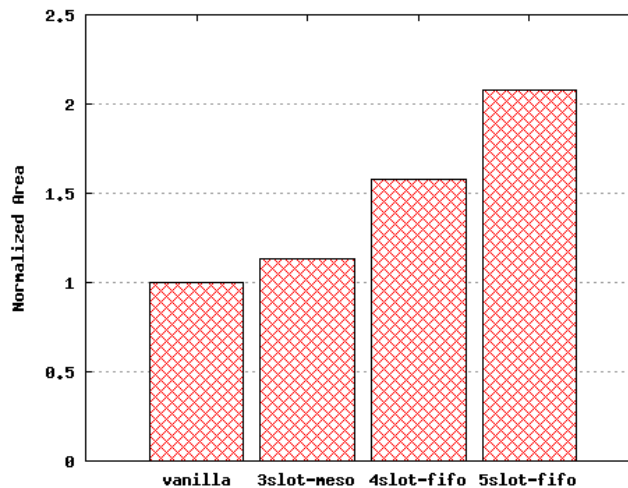


Figure 5: area occupancy of NoC switches with different synchronization schemes with respect to the vanilla fully synchronous switch.

However, D13 and [SB11] also prove a customized architecture can be engineered out of the general-purpose dual-clock FIFO that is capable of working in a synchronous regime with unknown phase offset with only a 4-slot buffer. Summarizing, in the library of dual-clock FIFOs, there exist two possible solutions to be deployed in a mesochronous scenario: (i) a baseline 5-slot and (ii) a specialized 4-slot FIFO synchronizer.

As depicted in Figure 5, the most specialized dual-clock FIFO is almost 40% more area expensive (in absolute term) with respect to the mesochronous solution. Obviously, the reason stems from the fact that the dual-clock FIFO has been natively conceived for a multi-frequency domain whereas the mesochronous interface is designed ad-hoc for the scenario under investigation (same clock frequency, different phase) and requires less control logic. Therefore, the main conclusion from this experiment is that a native (carefully engineered) mesochronous synchronizer is always more area-efficient than a customized dual-clock FIFO for the mesochronous regime (unless the customization is such that the dual-clock FIFO is changed into the mesochronous synchronizer itself!).

Finally, from the implementation viewpoint, unlike fully asynchronous interconnect fabrics, the synchronizer-based source-synchronous GALS architecture illustrated in Figure 3 is within reach of current mainstream design toolflows with just incremental effort. As proved by activities in WP 5 of the Galaxy project, some scripting effort within commercial synthesis frameworks enables these latter to meet the physical requirements of source-synchronous designs. Galaxy's results and other complementary efforts are reported in [PC08],[KL10].

From a latency viewpoint, the mesochronous architecture is again the most effective one. The focus here is on synchronization latency, since the stall/go mechanism implemented in our synchronizer ensures that a stall-to-go transition of the flow control signal can be immediately



propagated to the next stage. Hence, there are no wasted cycles at flow resumption. When the tightly coupled mesochronous synchronizer is composed by 3 buffer slots, there is no further synchronization latency introduced in the network switch, i.e., only 1 execution cycle is required to cross the switch in a null skew scenario. The reason is that the tightly coupled solution seamlessly replaces the input buffer of the network switch thus providing a fast, reliable and robust mechanism for data synchronization. Similarly, the 4 buffer slots mesochronous synchronizer variant does not require additional clock cycles to perform the synchronization. In fact, although the latency of the backward flow control signal is incremented by 1 clock cycle, the switch input buffer still latches the incoming data as soon as they change in the input and data is actually sampled at the next clock cycle by the output buffer. On the contrary, the 5 buffer slots mesochronous synchronizer variant requires a further clock cycle to alleviate the timing constraints. As a result, a switch composed by a mesochronous synchronizer needs, in the worst case, 2 clock cycles (plus the skew) to be traversed. This is an interesting result if compared with the 4 clock cycles required by a switch co-designed with the 5-slot dual-clock FIFO.

The ultimate result is that mesochronous NoCs are the reference solution for ultra-low cost synchronizer-based GALS systems. Yet, many state-of-the-art MPSoC platforms feature a fully synchronous interconnect fabric or this latter is a subset of a larger interconnect infrastructure making intensive use of dual-clock FIFOs. For this reason, it is important to assess the implementation cost for migrating from a fully synchronous NoC to a mesochronous NoC in order to pave the way for industrial exploitation of the new synchronization paradigm or at least to enable its inclusion in industrial roadmaps. The rest of this document pursues this objective and compares the architectures in Figure 2 and Figure 3 by means of physical synthesis runs. The xpipesLite NoC architecture [SA05] is used as baseline experimental setting to implement both GALS platforms, thus aiming at an architecture-homogeneous comparison.

Since flow control considerations are at the core of each realistic NoC architecture evaluation framework, the reader should recall that the flow control protocol used by xpipesLite is stall/go: a forward signal flags data availability (valid), while a backward signal flags a destination buffer full (stall) or empty (go) condition.

Next section will detail the architecture of the hybrid coupled synchronizer utilized to implement the switch of the GALS NoC based on mesochronous communication, which is used hereafter in the comparison with the fully synchronous NoC. Please notice that we do not use an entirely coupled synchronizer with the NoC (a tightly coupled architecture), but rather an hybrid coupled one (i.e., the data-path synchronizer is merged with the switch input buffer, while the control path one, much smaller in size, is not) since the work in WP6 has demonstrated its better robustness to layout constraints and its better timing margins.

3.3 HYBRID COUPLED MESOCHRONOUS SYNCHRONIZER

Usually, mesochronous synchronizers are just placed in front of the downstream switch (*the loosely coupled design style*). This has implications on the size of the switch input buffer as well, which should cover the round trip latency to sustain maximum throughput. Given the large buffering and latency overhead of this approach, we proposed in deliverable D6 and in [LB09] to bring the data-path synchronizer deeper into the downstream switch, as illustrated in **Figure 6**.

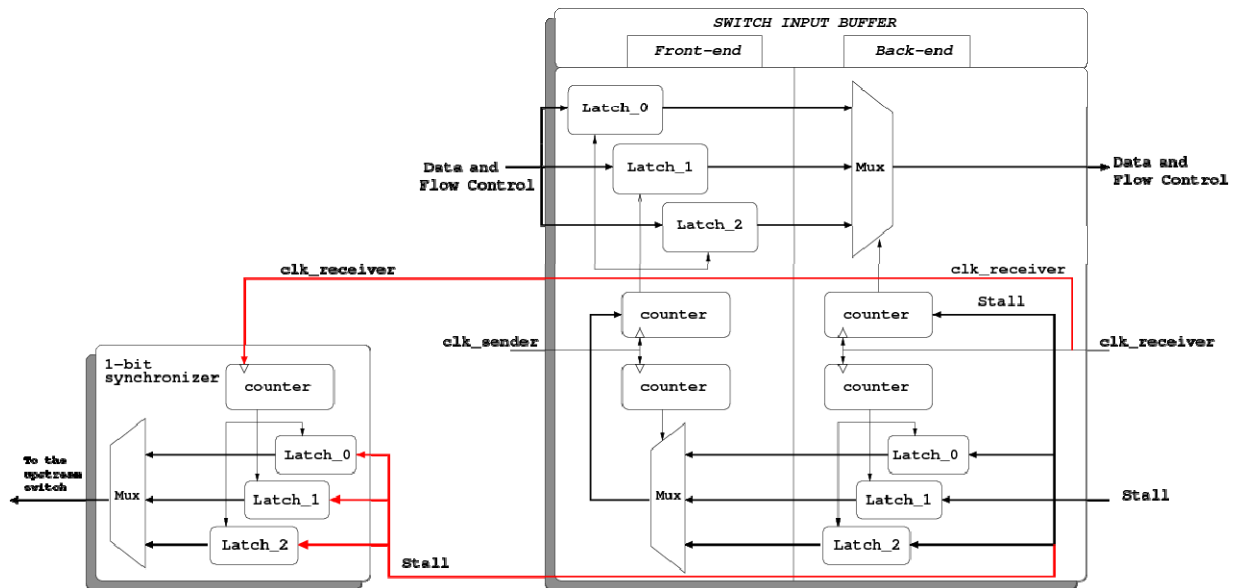


Figure 6: Hybrid Coupled Mesochronous Synchronizer.

The reference synchronizer architecture receives as its inputs a bundle of NoC wires representing a regular NoC link, carrying data and/or flow control commands, and a copy of the clock signal of the sender used as a strobe signal for them. The circuit is composed by a front-end and a back-end. The front-end is driven by the incoming clock signal, and strobes the incoming data and flow control wires onto a set of parallel latches in a rotating fashion, based on a counter. The back-end of the circuit leverages the local clock, and samples data from one of the latches in the front-end thanks to multiplexing logic which is also based on a counter. The rationale is to temporarily store incoming information in one of the front-end latches, using the incoming clock wire to avoid any timing problem related to the clock phase offset. Once the information stored in the latch is stable, it can be read, processed and sampled by the target clock domain.

In the architecture of **Figure 6**, the synchronizer output now directly feeds the switch arbitration logic and its internal crossbar, thus materializing the coupling concept of the mesochronous synchronizer with the switch architecture.

The ultimate consequence is that the mesochronous synchronizer becomes the actual switch input stage, with its latching banks serving both for performance-oriented buffering and synchronization. A side benefit is that the latency of the synchronization stage in front of the switch is removed, since now the synchronizer and the switch input buffer coincide. The buffering overhead in the switch input buffer because of flow control is also removed accordingly. The main change required for the correct operation of the new architecture is to bring the stall/go flow control signal to the front-end and back-end counters of the synchronizer, in order to freeze their operation in case of a stall. While this signal is already in synch with the back-end counter, it should be synchronized with the transmitter clock before feeding the front-end counter. The backward-propagating stall/go is then directly synchronized with the transmitter clock available in the front-end by means of a similar but smaller (1-bit) synchronizer. For this architecture solution, only 3 latching banks are needed in the synchronizer front-end, since link latency has been minimized. In practice, only 1 slot more than the input buffer in the fully synchronous switch. A loosely coupled approach would require a 4 slot input buffer and a 3 latch banks synchronizer.



As regards the control path, a 1-bit synchronizer is replicated in front of the upstream switch. This synchronizer is not merged with the downstream buffer, since this would give rise to overly tight timing constraints [LS10]. In contrast, integrating only the datapath synchronizer is denoted as the *hybrid coupling* and gives more guarantees for timing closure, and is the approach taken hereafter.

Next section will describe the physical synthesis techniques for the two GALS platforms under investigation. The former leverages a fully synchronous NoC while the latter utilizes a switch equipped with the abovementioned hybrid coupled mesochronous synchronizer. Interestingly, the hybrid coupling optimization enables to consider not an abstract synchronization concept nor a nice circuit scheme affected by layout uncertainties, but rather a mature GALS technology with practical relevance and suitable for chip fabrication. We find this is a mature achievement of the Galaxy project.

3.4 PLATFORMS SYNTHESIS AND PLACE AND ROUTE

Both the synchronous and the mesochronous platforms have been designed to be seamlessly integrated into an industrial design flow using commercial tools for physical synthesis. From this viewpoints, the design technology enhancements for physical design of GALS systems developed in WP5 are an integral part of this framework and deliverable.

Only standard cells are used and no full custom components. Synopsys Physical Compiler is used for placement-aware logic-synthesis, while Cadence SoCEncounter is used for place&route. The reference topology of our experiment is a 4x4 mesh network where each switch is connected to either a core or a memory (of size 1.5mm). As far as the physical synthesis is concerned, the same bottom-up methodology has been utilized for both platforms. Specifically, each network switch has been placed and routed in isolation with a target frequency of 500MHz. The clock tree of each switch has been synthesized with a tight skew constraint of 5% of the target clock period. Once the local clock tree is characterized with its input delay, skew and input capacitance, a *macromodel* is built in order to be used in the next design step. Furthermore, in order to implement a hierarchical clock tree synthesis, a buffer has been inserted to the input clock pin of each switch block. Once the switches have been placed and routed, they are imported as macro blocks in the main network design along with their libraries detailing both timing and physical characteristics. The next step consists of performing a top-level clock tree synthesis by leveraging the switch macromodels previously extracted. In fact, this model can be used to characterized the bottom clock tree given that these local clock trees will not be modified by the place&route tool. Therefore, in order to preserve the clock tree local to the switches, a **PreservePin** tag must be used in the CTS specification file (see the excerpt of code in Table 1).

```
MacroModel port switch_3x3_6_5/clk 414.7ps 357.2ps 423ps 365.2ps 0fF
MacroModel port switch_4x4_6_9/clk 420.1ps 366.8ps 436.1ps 388.8ps 0fF
MacroModel port switch_5x5_6_10/clk 483.4ps 458ps 440.6ps 418.4ps 0fF
...
....
AutoCTSRootPin  clk
Period          2ns
```



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

```
MaxDelay    1ns
MinDelay    0ns
SinkMaxTran 300ps
BufMaxTran  500ps
MaxSkew     100ps
NoGating    rising
MaxDepth    1024
RouteType   CK1
DetailReport YES
RouteClkNet NO
PostOpt     YES
OptAddBuffer YES

Buffer      HS65_LL_BFX18  HS65_LL_BFX27  HS65_LL_BFX35  HS65_LL_BFX44
HS65_LL_BFX53  HS65_LL_BFX71  HS65_LL_BFX9   HS65_LL_IVX18  HS65_LL_IVX27
HS65_LL_IVX35 HS65_LL_IVX44  HS65_LL_IVX53  HS65_LL_IVX71  HS65_LL_IVX9

PreservePin

+ xpswitch_0/clk
+ xpswitch_1/clk
+ xpswitch_2/clk
+ xpswitch_3/clk
+ xpswitch_4/clk
+ xpswitch_5/clk
+ xpswitch_6/clk
+ xpswitch_7/clk
+ xpswitch_8/clk
+ xpswitch_9/clk
+ xpswitch_10/clk
+ xpswitch_11/clk
+ xpswitch_12/clk
+ xpswitch_13/clk
+ xpswitch_14/clk
+ xpswitch_15/clk

End
```

TABLE 1: Clock tree specification file.

Please notice that the hierarchical CTS has been used both for the synchronous and the mesochronous platforms, since this is a standard methodology for parallel hardware platforms. The only difference is the skew constraint in the top level clock tree, which can be loosened for



the mesochronous design while should be tightly enforced for the synchronous one with respect to the clock trees which are local to switches.

Final step of our hierarchical methodology consists of routing the switch-to-switch links and performing parasitics extraction for accurate static-timing analysis and power estimation.

Timing closure for both the synchronous and mesochronous NoC has been achieved at 500 MHz by performing exactly the same physical synthesis steps (although with different parameters at given steps) for the two designs under test. **Figure 7** reports the GALS system implementing an hybrid coupled mesochronous network after place and route.

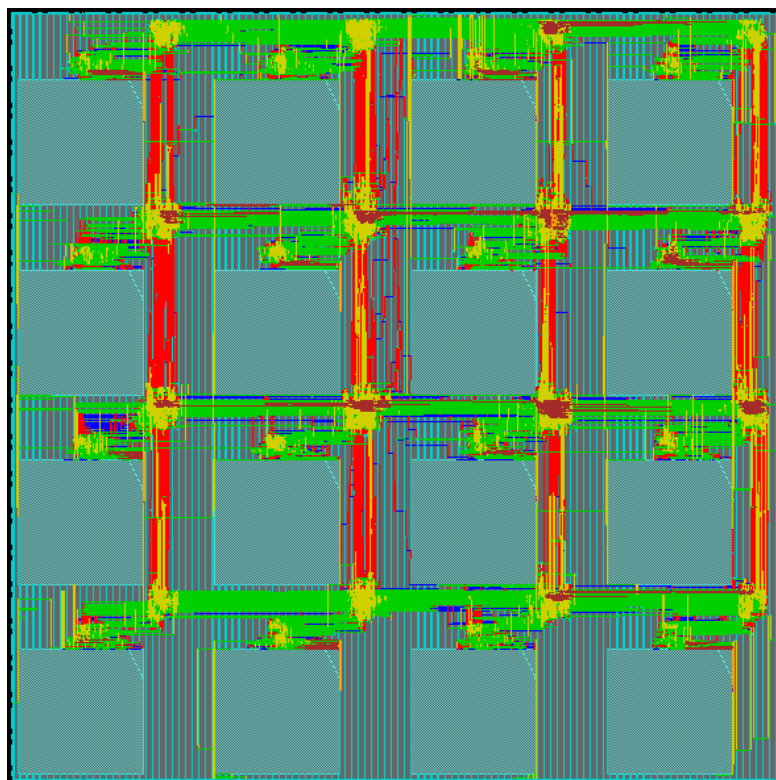


Figure 7: Post-place and Route screenshot of a 4x4 Platform.



4 POST-LAYOUT EXPERIMENTAL RESULTS

4.1 AREA AND WIRING OVERHEAD

Figure 8 reports post-place&route area and wiring statistics for the architectures under analysis. From an area viewpoint, both systems exhibit the same footprint. More in detail, our baseline architecture (i.e., the fully synchronous mesh) features a 2-slots input and 6-slots output buffers. On the other hand, its mesochronous counterpart has 3-slots input buffer and exactly the same amount of output buffering. Nonetheless, the area overhead is identical.

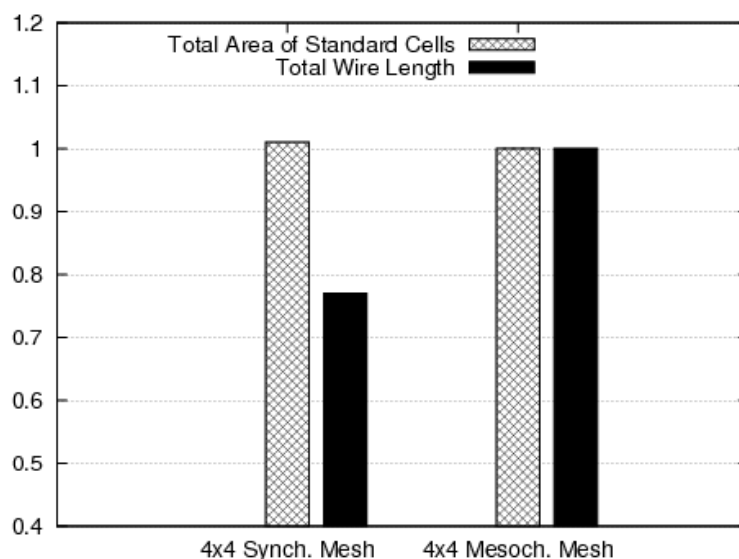


Figure 8: Area and wiring overhead normalized with respect to the mesochronous NoC.

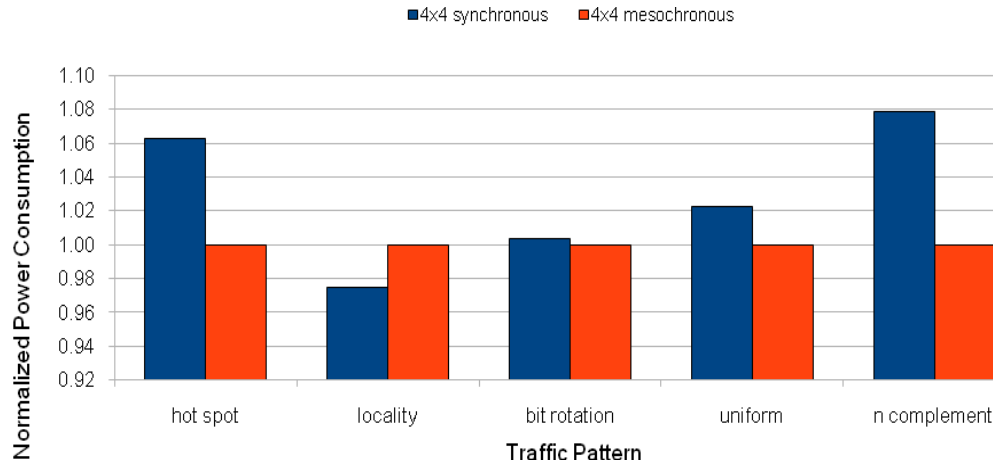
This is due to the fact that synchronization mechanisms, tightly coupled in the input buffer, are implemented through latch banks, which require typically a smaller area footprint compared to the flip-flops adopted in the input buffer of the baseline architecture. The ultimate result is an equal area occupation in both platforms. From the wiring point of view, a 23% net saving is achieved by the fully synchronous platform. The reason lies in the fact that the mesochronous platform features an additional clock wire per output port utilized as strobe signal for data synchronization and a further external single bit synchronizer for backward flow control synchronization instantiated in each of the 48 switch-to-switch channels of the network. Last but not least, the slightly more complex network topology (due to the external although small control-path synchronizers) contributes to a more complex structure of the clock tree.

4.2 SYSTEM-LEVEL POWER ANALYSIS

By leveraging post-layout netlists and back-annotated switching activity, we were able to achieve very accurate power figures. In fact, cycle-accurate simulations have been carried out



with uniform random traffic as well as with all the network switches in idle conditions. A value-change-dump file (VCD) has been annotated from the simulations and consequently utilized to carry out a very accurate power estimation with Synopsys PrimeTimePX.



**Figure 9: Power consumption with high network load
(normalized with respect to the mesochronous network)**

Figure 9 reports power consumption of both fully synchronous and mesochronous networks under several synthetic traffic patterns when the network is heavily loaded (i.e, small idle time between any two consecutive transactions). In agreement with the requirements of task 6.3 of WP6 of the Galaxy project, the traffic patterns have been modeled according to the Micro-Benchmark Specification 1.0 in the context of the OCP-IP standardization activity. When stimulating the networks with these traffic patterns, the mesochronous Network-on-Chip exhibits a smaller power consumption with respect to the fully synchronous one. The reason lies in the inherent architectural difference between the input buffer of the mesochronous switch and of the synchronous one. In this latter, both flip-flop banks are triggered at each cycle by the clock signal, although typically only one bank samples actual input data (the other one, although fed by the clock, is enabled only when a stall signal is received and a backup of the data should be created). The triggering signal is the clock signal switching the input capacitance of the flip flops at the operating frequency of the network.

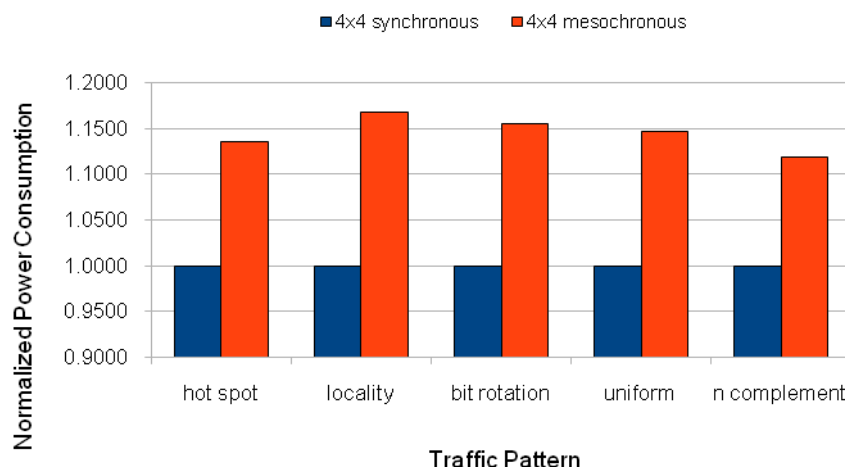
Conversely, latch banks of the mesochronous input buffer are triggered by an enable signal driven by a counter. Since the counter logic enables only a single latch bank at a time, the ultimate register power consumption of the mesochronous input buffer is smaller than its synchronous counterpart. Here the non-sampling banks do not even get the triggering signal. The selective triggering signal switches at a frequency which is one third of the network operating speed, since the 3 latch banks are used in a rotating fashion.

On the other hand, when the networks under investigation are stimulated under light load traffic patterns (large inter-packet idle time, see Figure 10), the overall power consumption plays in favor of the synchronous solution, since the larger standby power of the mesochronous NoC comes into play. The rationale is that, in the presence of network idleness, the synchronous input buffer is designed in such a way that data sampling happens if and only if data is valid because of proper assertion of the enable signal. In practice, if data is



not valid, then the enable input of sampling flip flops is disabled and only some switching power associated with the clock signal input of the flip flops is accounted for.

Conversely, in the mesochronous design latch banks and counters always work regardless of the validity of the received signals since they are driven by the source-synchronous clock signal transmitted together with data on source synchronous links. This latter signal keeps switching all the time, thus causing the receiving interface to burn power regardless of the validity of input data. This adds up to the switching power of the source synchronous clock signal itself. This calls for a further optimization of the mesochronous architecture in order to implement a smart clock gating strategy that is left for future work. Just to give an insight, the reason why clock gating was not implemented in this mesochronous architecture is twofold. First, 3 latch banks do not suffice to store all the data in flight when also clock gating is implemented. In order to remove such limitation, an increment of the number of latch banks to 4 units and a bit of gating logic would suffice in implementing such power saving technique. Second (and more challenging concern), by gating the source synchronous clock the relative alignment between write and read pointers in the synchronizer front-end and back-end would get lost, thus making flow resumption challenging. Such second-order optimizations are left for future work. The objective of the current activity and deliverable is to develop the first generation of synchronizer-based GALS NoC technology and also to identify those optimization targets that will lead to the second generation of such synchronization interfaces. For the time being, it is useful to consider that under high traffic load, the mesochronous NoC proves power efficient. It should also be observed that by applying a clock gating technique in the synthesis flow of the fully synchronous NoC, the power gap could be easily bridged, however the key take-away does not change significantly: under high-traffic load, the migration from a fully synchronous NoC to a mesochronous NoC for GALS systems does not imply any major power overhead. When the network experiences significant idleness, then the larger standby power of the mesochronous NoC makes the difference and a 15% overhead in the worst case should be expected. Let us now investigate how the above considerations combine to explain power figures in the presence of a real-life application.



**Figure 10: Power consumption with light network load.
(normalized with respect to the synchronous network)**



Our experimental setup was extended by mapping a Full-HD Video Playback application with 60 Frames/second in true color onto our systems and simulating its operation in order to capture the overall system power consumption (reported in **Figure 11**). The traffic pattern of the application has been specified in terms of a core graph representation with annotated average communication requirements. Such representation was achieved by means of the specification from industrial partners of the consortium.

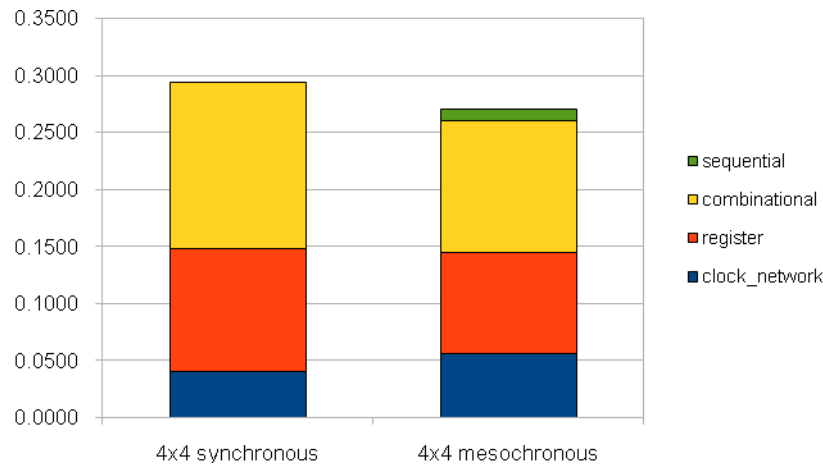


Figure 11: Power consumption in W under a realistic multimedia application.

Such Full-HD Video application heavily stresses the capability of both systems. However, the mesochronous NoC turns out to be more power effective with respect to its synchronous counterpart. As it can be seen in the power breakdown reported in **Figure 11**, the power consumption of the clock network is more expensive with respect to that of the synchronous platform. The reason is that, the mesochronous architecture utilizes a further clock signal (per channel) in order to strobe data at the receiver end. In terms of registers power consumption, the fully synchronous architecture is more power hungry due to the utilization of flip-flops whereas the mesochronous synchronizer is implemented leveraging latches. There is a small contribution of sequential cells in the mesochronous system which are basically those cells clocked by a signal that is not in the clock network, i.e., the counters at the front-end interface of the synchronizer. Regarding the combinational power consumption, the mesochronous architecture has been heavily optimized and thus requires less combinational logic with respect to the fully synchronous input buffer.

4.3 CLOCK TREE POWER ANALYSIS

With a mesochronous NoC, an interesting opportunity pointed by [JA10],[ML06],[VI08] is to exploit hierarchical clock tree synthesis to reduce power of the top level clock tree. The tuning knob to materialize power savings is the relaxation of the skew constraint, so that less buffers are instantiated in the top-level tree. While this is often reported in papers as a common sense belief without any experimental evidence (or at least single design points are characterized, without a true spanning of the skew constraint and measurement of the associated power savings), this document aims at a step further in the direction of assessing the efficiency of this mechanism and of quantifying such efficiency with a parametric experimental framework.



For the first time, we experimented this on the 4x4 mesochronous mesh. Given the relatively small system size, we constrained the top level tree to be placed and routed outside IP core area, which captures the challenging requirements of many real-life MPSoC designs

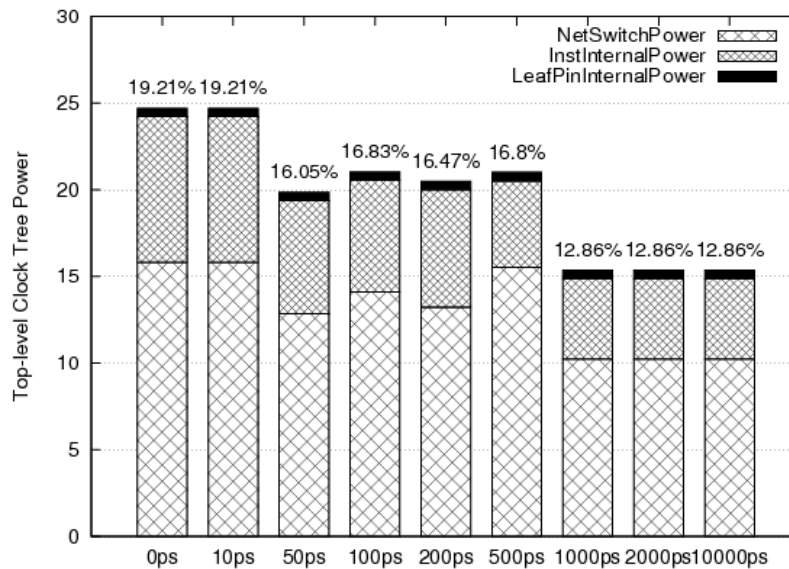


Figure 12: Power of the top level clock tree as a function of required skew.

Power of the top-level clock tree is reported in **Figure 12** as a function of the required clock skew, ranging from 0 to 10000 ps. Transition time constraints are set to be very tight for the CTS tool (Cadence SoC Encounter). The percentage on top of the bars indicates the impact of the top level tree on the total clock tree power. We can see that power of the top level tree can be decreased by up to 40%, from roughly 25mW to 15mW. Also, the impact of the top level tree on total clock tree power can be as large as 20% and can be reduced to 12% with the hierarchical clock tree synthesis.

Figure 12 may be misleading. In fact, the required skew does not keep up with the actually enforced skew by the CTS tool. In fact, when 2000 ps or more were required, the achieved clock skew saturates at about 600ps. Power savings could be even more than those reported if only the CTS tool were able to infer larger skews while saving clock tree area. Unfortunately, the CTS has not been natively conceived for this, but rather for the opposite: minimizing the skew. This result allows to identify a further optimization target for the second generation of synchronizer-based GALS NoCs. CTS tools should be able not only to meet tight skew constraints at the cost of increased clock tree complexity, but also to decrease such complexity as a function of the skew constraint relaxation. Further power savings will be measured for the top level clock tree subject to this optimization.

| Required Skew | Actual Skew | Top Tree Power | % of total clock tree |
|---------------|-------------|----------------|-----------------------|
| 10ps | 320ps | 64.817mW | 13.93% |
| 1000ps | 973ps | 61.307mW | 13.17% |

TABLE 2: Top clock tree power for a 64 cores system.



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

However, there is another effect that becomes apparent when the same experiment is performed on an 8x8 2D mesochronous mesh with 64 cores. Results are reported in Table 2. The CTS tool was not able to meet the required upper bound on the skew of the top level clock tree. When 10 ps were required, the actual skew resulted 320 ps. The clear message here is that as the system size becomes large and (not showed here) feature sizes shrink, it will become impossible to meet the desired skew constraint in the top level clock tree. This result is a precise quantification of the common belief that globally enforcing the skew constraints will not be soon feasible. Clearly, this calls for a skew absorbing mechanism in the NoC architecture, since in the CTS process the onset of such a skew is unavoidable.



5 VARIABILITY ROBUSTNESS

In this section, in order to prove robustness to timing uncertainties of the platforms under test, we performed the following experiment. We synthesized, placed and routed two synchronous vs two mesochronous 5x5 switches placed 1.5mm far apart, like in the global networks. The smaller test case enables to focus on the variability robustness of the architectures while being able to precisely identify where exactly such robustness originates in the switch microarchitecture.

Each switch has its own input pin for the bottom level local clock tree. We manually injected positive vs. negative skew between these two clock pins, thus analyzing the skew tolerance of the two designs to uncertainties in the top-level clock tree. The motivation lies in the fact that the two architectures differ only in the switch interfaces, therefore variability effects affecting internal switch gates and/or nets are likely to have the same impact on the designs under test.

So, let us focus on uncertainties affecting the top-level tree. The skew tolerance of the two designs is reported in **Figure 13**. The synchronous design has some margins both for positive and negative skews, before violating the hold time and the setup time constraints respectively. However, the margins of the mesochronous design are much larger and approach 90% of the clock period. The positive skew tolerance is larger since the interconnect delay consumes part of the negative skew margins of the mesochronous link.

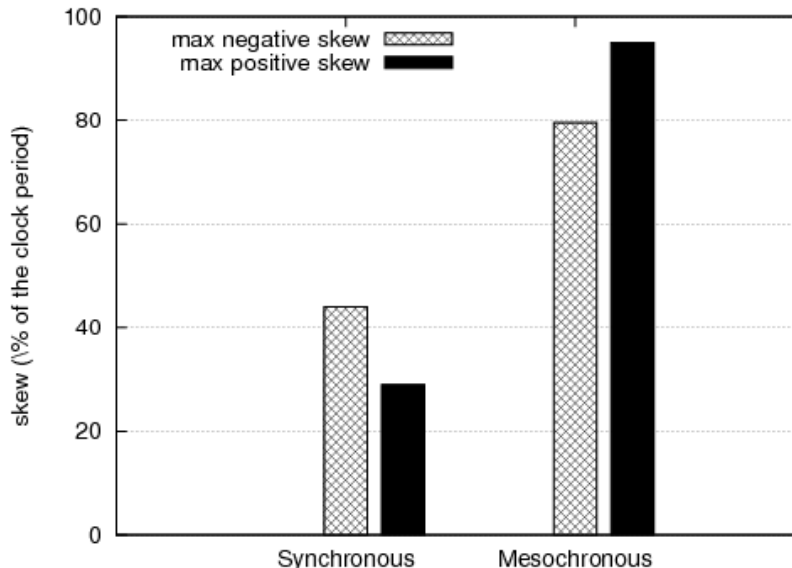


Figure 13: Skew tolerance with slack (enforced by the physical synthesis tool).

The above experiment however suggests that the designs have some slack with respect to the target speed of 500 MHz they were synthesized for. Therefore, we repeated the experiment by having the two synthesized designs run at their maximum operating speed, and evaluating skew tolerance in that operating point. In practice, slack was nullified and the true skew tolerance of the designs can be experimented.

The new results are reported in **Figure 14**. Now, the synchronous design has clearly null negative skew, since the negative skew directly impacts the critical path. This latter goes from



the output buffer of the upstream switch to the input buffer of the downstream switch going through the switch-to-switch link. A certain tolerance to the positive skew is however always there. In contrast, the mesochronous design features an excellent robustness to both positive and negative skew, meaning that such robustness is intrinsic of the architecture. The reason is that as the skew becomes increasingly relevant, a different critical path than the one constraining the operating speed shows up and causes failure. The skew does not directly affect the frequency-limiting path delay, but makes a non-critical path become critical as it becomes larger. The zero-skew difference between the delay of the two paths is the inherent skew tolerance of the mesochronous design.

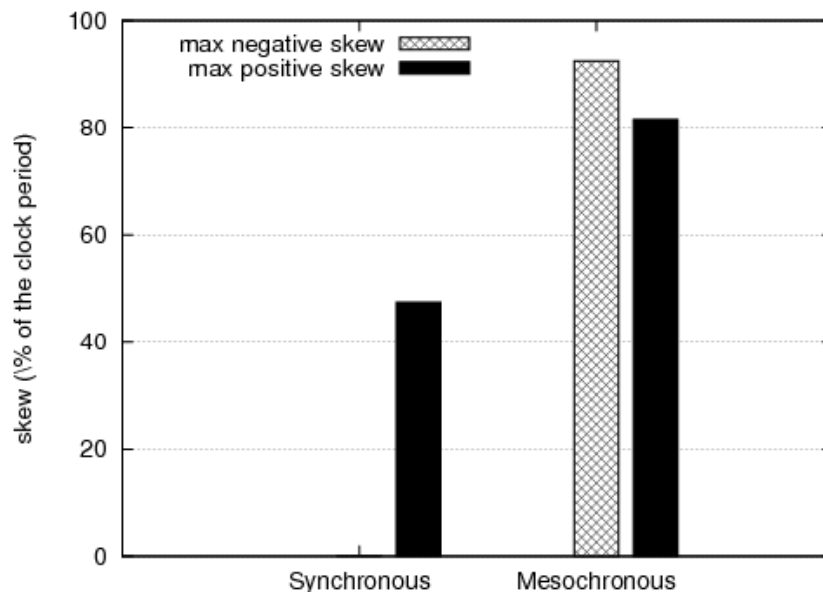


Figure 14: Skew tolerance with no slack.

The indications of these experiments have a number of implications:

- for a given target frequency of the NoC (dictated by system considerations), a mesochronous NoC can guarantee timing closure at that speed even though the CTS tool is not able to constrain the skew of the top-level clock tree (like in the 64-core system). In contrast, the synchronous NoC is directly exposed to such top level clock tree skew.
- for a given target frequency, the mesochronous NoC will certainly prove more robust to process variations affecting the top-level clock tree. In fact, the ultimate effect of variability will be an unexpected skew deviation with respect to the CTS tool statistics, a deviation that the mesochronous NoC can better absorb.

The above findings on process variation tolerance are now validated by means of a real injection experiment of process variations in the top level clock tree of a 4x4 2D mesh. Recently, a methodology for characterizing variability in NoC links was proposed [HS10]. The detailed variability model used in that work was later extended to NoC routers in order to analyze how process variation simultaneously affects both components of the network [HR10]. This new model takes into account, at the same time, systematic and random front-end variations due to, respectively, defects in the photolithographic process and deviations in the threshold voltage due to Random Dopant Fluctuations (RDF). The main use of this model is to



generate many instances of a given chip and statistically analyze how process variation affects that particular design.

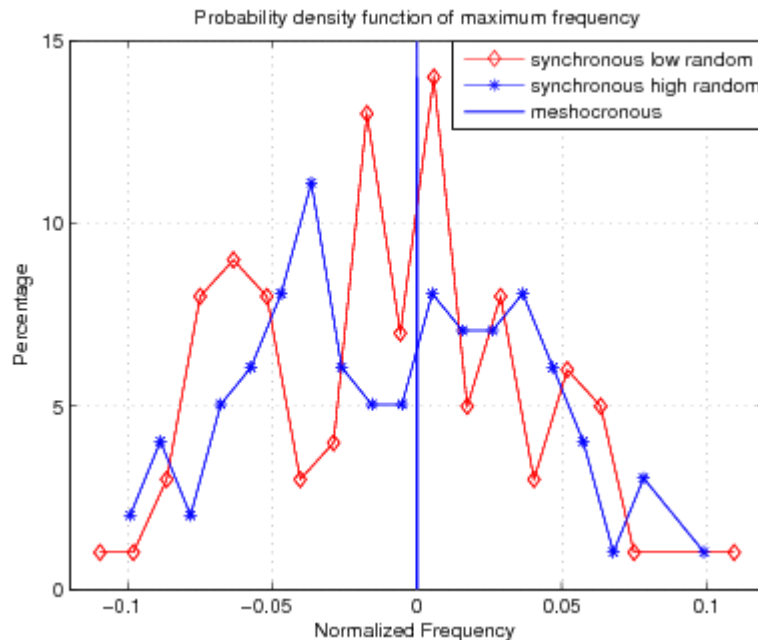


Figure 15: Variability-induced percentage deviations of maximum speeds of the designs under test with respect to nominal ones.

This model has been enhanced to consider back-end variability due to resistance variations introduced by the chemical metal planarization process (CMP) in wire dimensions. Concretely, the effect of metal thickness variations in wire resistance has been introduced. According to the predictions of the ITRS, expected variations of the metal thickness will be lower than 10% (3sigma).

Note that in agreement with other studies, the effect of capacitance variations in NoC links is not considered [HS10],[MT07]. For those wires of a link routed in the same layer the same variation to the metal thickness has been applied. This behavior has been considered to satisfy the strong spatial correlation present in the variations introduced by the CMP process.

Figure 15 shows the probability density function (pdf) of the maximum achievable frequency of both synchronous and mesochronous designs when variations are injected to the nominal design. The variations injected are both systematic and random. Concretely, the variability sources considered are: transistors channel length $3\sigma_{Leff}=12\%$, threshold voltage $3\sigma_{Vth}=\{33\%,58\%\}$, and the metal thickness $3\sigma_t=10\%$. These values reflect expected variations for a 45nm technology node [ITRS].

Results confirm that for the synchronous design the variability injected in the clock tree has a considerable impact on the maximum achievable frequency. Concretely, we have measured a standard deviation of the maximum frequency equal to 6.8% and 6.2% for the cases of low and high random variations, respectively. Note that differences in the pdf for high and low random variation are minimum. This can be explained by the fact that random variations do not have a significant impact in the top level clock tree, thanks in part to the large size of the clock buffers. On the contrary, the mesochronous design is clearly able to absorb the variations introduced in the clock tree, since it is able to preserve the nominal frequency in all



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

cases. Obviously, since the clock skew directly impacts the critical path of the synchronous design, it is sometimes possible that this latter works at a higher speed of the mesochronous one (depending on the sign of the skew), however this argument is not able to counter the conclusion: the mesochronous NoC proves more robust to process variations in the top-level clock tree.



6 PERFORMANCE CONSIDERATIONS

As already pointed out when presenting the synchronizer architecture, our proposed hybrid coupling mesochronous interface is perfectly integrated with the existing switch architecture. The ultimate purpose of such new input buffer is threefold: buffering, flow control and data synchronization. Therefore, comparing the fully synchronous and the new mesochronous switch, the only notable difference resides in the input buffer architecture. From a latency viewpoint, let us first analyze what happens in a typical synchronous scenario. In the xpipesLite architecture, traversal of the fully synchronous switch takes a single cycle. Therefore, a switch--to--switch data transmission requires 1 clock cycle to cross the link and a further clock cycle for the switch traversal. In order to cover the same path, the GALS switch requires from 1 to 3 clock cycles depending on the skew condition (that could be either negative or positive). As we already proved, both systems are able to tolerate a certain amount of skew. Nonetheless, when the amount of skew is such that the mesochronous switch spends an additional clock cycle for data synchronization, the fully synchronous switch has already failed. The ultimate conclusion is that when both systems are able to work with the same amount of skew, the average cycle latency is exactly the same. When the skew is such that a further cycle has to be devoted to data synchronization, only the mesochronous architecture can keep working without any failure in that operating condition.



7 PORTING TO THE 40NM TECHNOLOGY NODE

The building blocks of the developed synchronizer-based GALS NoCs were ported onto the 40nm silicon technology provided by Infineon. In particular, a fully synchronous switch was synthesized with the new technology, and so were a loosely coupled switch (i.e., with mesochronous synchronizers in front of each input port) and a tightly coupled switch (i.e., with mesochronous synchronizers merged with the switch input buffers). The objective of this porting experiment was to validate the capability of the tightly coupled architectures to preserve area savings with respect to loosely coupled ones and to preserve the marginal overhead when migrating from a synchronous switch to a tightly coupled mesochronous one. The reader should observe that what really matters here is the capability to preserve relative advantages across technology nodes, not the comparison between actual 65nm and 40nm area and performance numbers, since this comparison would not be technology library homogeneous.

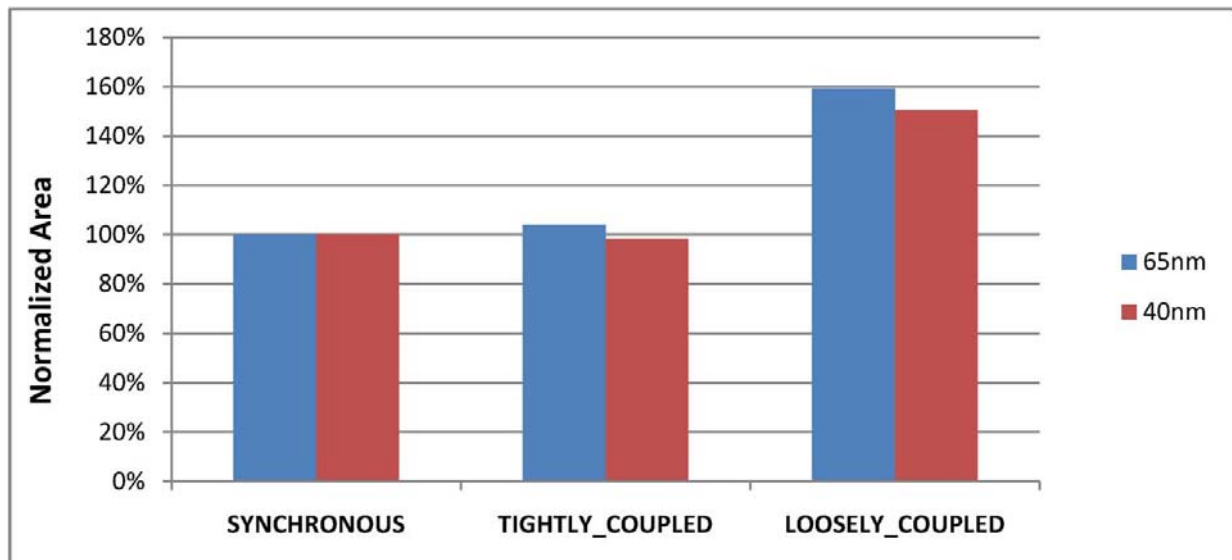


Figure 16 Relative area ratios across technology nodes

Figure 16 shows relative area ratios between the fully synchronous, the tightly coupled and the loosely coupled mesochronous switches and how they vary across technology nodes. The tightly coupled architecture proves even more area-efficient in 40nm, however this is an effect of the different set of standard cells included in the technology libraries and of the way they are handled by the synthesis tool. The loosely coupled architecture seems less expensive in 40nm, however its overhead is still prohibitive (around 50%). Overall, when accounting for the non-homogeneity of the technology libraries, the conclusion is that the tightly coupled architecture preserves its area advantages with respect to the loosely coupled one, and still features comparable complexity with respect to a fully synchronous switch.

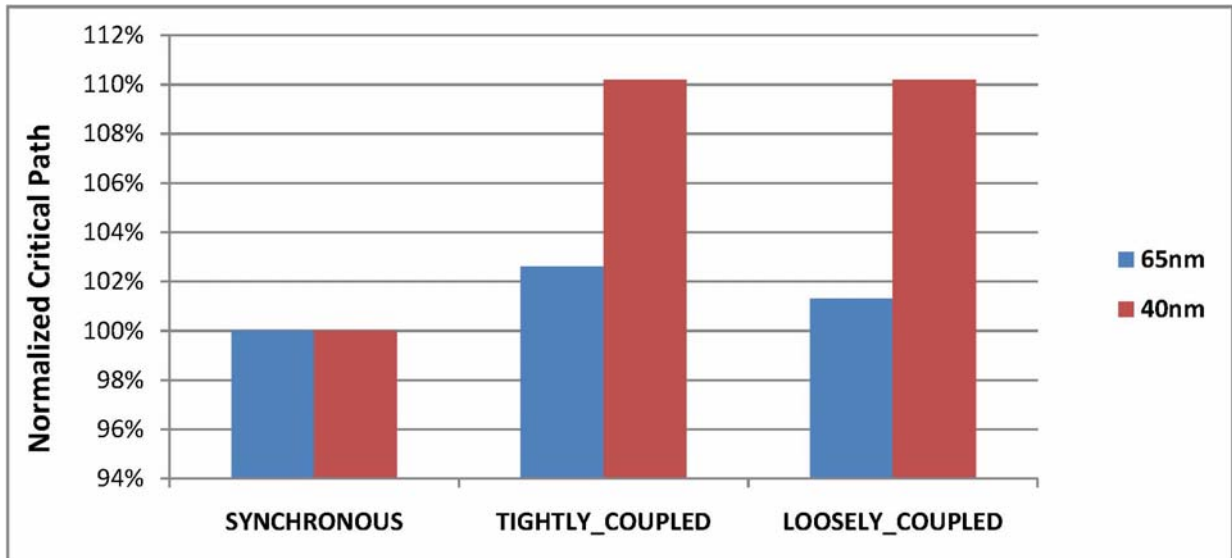


Figure 17 Relative critical path delay ratios across technology nodes

Figure 17 shows relative ratios of critical path delays across technology nodes. Here, a slightly larger degradation of the critical path when moving from the synchronous to the tightly coupled and loosely coupled architectures can be observed (from about 1 or 2% to 10%). However, since the critical path stays the same across the technology libraries, we do not view this as an effect of the architecture but rather as an effect of the lower number of cells in the 40nm technology library and of the consequent difficulty of the synthesis tool to optimize the path delay.

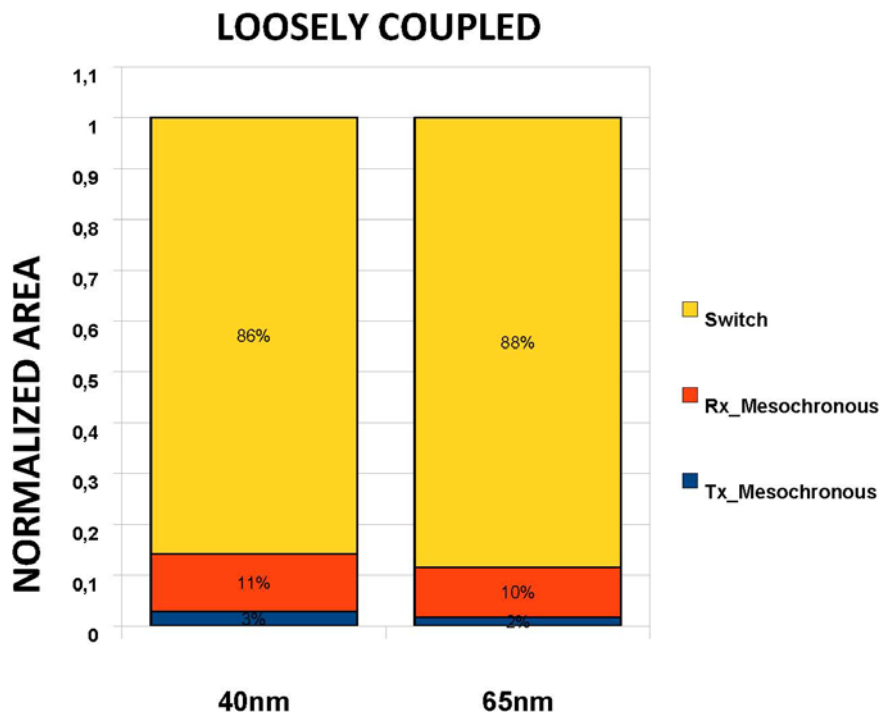


Figure 18 Area breakdown for the loosely coupled switch across technology nodes



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

Figure 18 shows the area breakdown of the loosely coupled switch architecture. It can be observed that the relative contribution of switch sub-blocks to total area stays almost the same across technology nodes. The larger contribution of the data-path synchronizer with respect to the 1-bit control path synchronizer is clearly visible.

Overall, we conclude that relative area and critical path ratios between NoC switch variants do not change a lot as an effect of the scaled technology node, but rather as an effect of the different features (and different targets) of the technology libraries used in the experiments. Hence, the developed synchronization technology proves suitable for implementation in aggressively scaled technology nodes.



8 CONCLUSIONS

Evolution of MPSoCs to GALS systems is an ongoing process, driven by the immediate need to decouple voltage and frequency of IP cores from each other for power management. However, when a GALS NoC is implemented as an independent clock domain with dual-clock FIFOs at the boundaries, then physical designers have again to deal with a global chip wide clock tree (i.e., the one of the network itself). By capitalizing on mature mesochronous technology, this document compares a mesochronous NoC and a fully synchronous NoC (both for use in a GALS system, i.e., they both implement dual-clock FIFOs at the boundary) in a systematic way. The lesson learned from this experimental work can be summarized as follows:

- Mesochronous synchronizers are more lightweight than dual-clock FIFOs from an area and power viewpoint. Moreover, they feature a lower synchronization latency. Even customizing the dual-clock FIFO for a mesochronous regime does not materialize a more cost-effective block than the native mesochronous synchronizer. **Therefore, we view mesochronous NoCs as the reference solutions for ultra-low cost GALS MPSoCs.** This comes with additional benefits: the network can be used as an always on interconnect fabric to better support power management strategies, and the network ends up having an homogeneous speed thus avoiding to reduce network performance to the speed of the slowest switching fabric.
- **A fully synchronous NoC can be evolved to a mesochronous NoC at no area, latency and dynamic power overhead because of the hybrid coupling design style.** This implementation technique is the most important contribution of WP6 of the Galaxy project to improve state-of-the-art MPSoC platforms in the direction of relaxing synchronization assumptions in a cost-effective way. Although a tight coupling design style could be even more aggressive to reduce synchronization overhead, we found the hybrid coupling technique (i.e., merging the data path synchronizer with the switch input buffer, while leaving the control path synchronizer as an external module to the switch) more robust to layout constraints. In practice, it can support longer inter-switch links for a given target frequency. With respect to the loosely coupled solution (i.e., both data path and control path synchronizers placed as external blocks in front of the switches, with consequent oversizing of switch buffering to cover round trip latency), the hybrid architecture features a significantly shorter area footprint while closely tracking its design margins.
- **During network activity, the mesochronous NoC proves more power-efficient because of the inherent clock gating** implemented at its tightly coupled synchronizers in the switch input buffer. A clock gating technique applied to the synchronous NoC may probably bridge the gap. In any case, the mesochronous NoC does not feature any significant overhead from a dynamic power viewpoint. These considerations hold as long as the network does not experience significant idleness. Another difference between the two networks is the use of latches in the mesochronous NoC for the synchronization task. On one hand, such latches lead to some area and power savings for the mesochronous NoC. On the other hand, they pose a precise challenge associated with the capability of testing them.
- **In standby mode of operation, a 20% higher standby power is incurred by a mesochronous NoC** with respect to a fully synchronous one because of the transmitted and continuously switching clock signals in source synchronous links. Therefore, at



runtime, the higher the network idleness the higher the impact of the standby power overhead for the mesochronous NoC.

- **Hierarchical clock tree synthesis in a mesochronous NoC can potentially reduce total power of the top level clock tree at the cost of progressively loosening skew constraints in the same tree. However, power savings that are achieved in this way are not significant yet**, for a number of concurrent reasons. First, there is a gap between required maximum skew and the obtained one, since the CTS tool has been conceived for minimizing skew, and not for increasing it. Therefore, to take full advantage of this effect, CTS tools should be customized accordingly. On the other hand, when tight skew constraints are required under challenging physical and timing constraints, the CTS tool is not able to meet the target and therefore clock tree power does not increase a lot. In practice, this means that with current CAD tools it will become rapidly impossible to enforce tight skew constraints in the top level clock tree. Under these operating conditions, it is important to have an underlying architecture with inherent skew robustness. In our experiments, mesochronous NoCs proved capable of meeting this requirement.
- As technology keeps scaling to the nanoscale regime, process variations become increasingly important. In this document, we assess their effects when affecting the top level clock tree, thus experimenting the variability robustness of switch interfaces. The mesochronous NoC exhibits an inherent robustness to such delay uncertainties, since the critical path that constraints the operating speed is not directly impacted by the skew of the top level clock tree. This key feature of the mesochronous NoC in turn has important implications in everyday life's design practice:
 - for a given target frequency of the NoC (dictated by system considerations), a mesochronous NoC can guarantee timing closure at that speed even though the CTS tool is not able to constrain the skew of the top-level clock tree (like in the 64-core system). In contrast, the synchronous NoC is directly exposed to such top level clock tree skew.
 - for a given target frequency, the mesochronous NoC will certainly prove more robust to process variations affecting the top-level clock tree. In fact, the ultimate effect of variability will be an unexpected skew deviation with respect to the CTS tool statistics, a deviation that the mesochronous NoC can better absorb.
- The area savings provided by mesochronous NoCs with respect to fully synchronous NoCs in the 65nm technology node have been proven to be retained also in the more aggressive 40nm technology node. The validation concerned the relative savings between architecture variants across technology nodes. In particular, the relative area footprint advantages of (more or less) tightly coupled synchronization schemes over loosely coupled schemes were confirmed, and so was the marginal overhead with respect to a fully synchronous switching fabric. Also, the lack of a relevant degradation of the critical path when going mesochronous was demonstrated also in the 40nm technology node. The sensitivity of final quality metrics proved more significant with respect to the features of the technology libraries used for the implementation, while many indicators confirmed the insensitivity of the benefits of the developed technology to a scaled technology node. This paves the way for the chip fabrication planned in the Galaxy project and for GALS NoC technology exploitation as geometries shrink deeper into the nanoscale regime.

The Galaxy project aimed at the development of the first generation of synchronizer-based GALS NoC technology, going significantly beyond comparison frameworks based on synchronizer concept schemes and interfaces with questionable timing margins. This deliverable has demonstrated that the goal of developing mature GALS NoC technology in a



GALAXY

GALS InterfAce for CompleX Digital
SYstem Integration

Confid. Level: Public
Date : 30/11/2010
Issue: 1

cost-effective way has been achieved. However, the project also identified key optimization targets that might be the stepping stone into the second generation of mature synchronizer-based GALS NoCs. Such targets have been briefly summarized hereafter:

- Development of clock gating techniques for source synchronous links to cut down on standby power.
- Enhancement of mainstream CTS tools with extensions capable of decreasing clock tree complexity as the skew constraint is relaxed, in order to increase power efficiency of a hierarchical clock tree synthesis.

This deliverable should be seen as an enabler for the testchip fabrication task envisioned in the Galaxy project and due in deliverable D, where test structures embodying the developed synchronizer-based GALS technology will be laid out on the 40 nm silicon technology provided by Infineon. The testchip will be the final means of validating the developed technology.