A novel Asynchronous Interface with Pausible Clock for Partitioned Synchronous Modules

Duarte L. Oliveira, Tiago Curtinhas, Lester A. Faria
Divisão de Engenharia Eletrônica
Instituto Tecnológico de Aeronáutica – ITA
SJC/SP – Brazil
e-mail duarte@ita.br, thiagohd@ita.br, lester@ita.br,

Leonardo Romano
Departamento de Engenharia Elétrica
Centro Universitário da FEI
SBC/SP – Brazil
e-mail leoroma@uol.com.br

Abstract— **Contemporary digital systems must necessarily be based on the "System-on-Chip – SoC" concept. An interesting style for SoC design is GALS (Globally Asynchronous, Locally Synchronous) paradigm. Currently, the major drawback in the design of a GALS system shows to be the asynchronous interface (asynchronous wrapper – AW), especially when the GALS system is applied to a multi-point topology. The AW interfaces found in literature are always based on controller ports. They are responsible for data communication between locally synchronous modules, where to each point of data communication there is an input or output port. The increasing number of port leads to complex AWs and to a high increase in area. This paper proposes a new asynchronous GALS interface focused on multi-point GALS. For a case study considering a multi-point data communication system, the proposed interface achieved an average reduction in area (products + literals) of 85% and 74%, when compared to two different AWs found in literature.**

Keywords— multi-point GALS; BM specification; port; logic asynchronous

I. INTRODUCTION

Contemporary digital systems must be based on the "System-on-Chip - SOC" concept. The main reason is to satisfy a ruthless demand for high performance, reusability and low power requirements [1,2]. SOC circuits are composed by functional modules that may be "Intellectual Property (IP)" from different companies. These IP modules are pre-designed, checked and tested to achieve high performances. They allow reducing cost and, especially, designing time. If SOC circuits are implemented in VLSI with a global clock signal, not only the performance and power (clock skew and distribution network) are affected, but also the timing analysis becomes more complex [3,4].

A solution that significantly reduces the problems related to the global clock is the Globally Asynchronous Locally Synchronous (GALS) methodology, proposed initially by Chapiro [5]. A GALS system consists of functional synchronous modules, which present individual clocks with non-related frequencies. They communicate to each other asynchronously. This paradigm is a combination of synchronous and asynchronous paradigms, where each one of the modules is synchronous. To handle the asynchronous communication between modules, an interface circuit must be included around each synchronous module, creating an asynchronous wrapper (see Fig. 1) [6]. Techan et al. [7] shows different styles of asynchronous interfaces focusing on GALS systems. This interface may be composed, for example, by a local clock, a FIFO, or an asynchronous communication controller (inputs ports, output ports). Many different asynchronous ports designed for GALS can be found in literature [8-15]. Among them, the use of "communication ports" show to be very interesting, once they allow removing the typical handshake of synchronous modules. Therefore, the synchronous modules can be designed through the standard and well-known techniques of synchronous design.

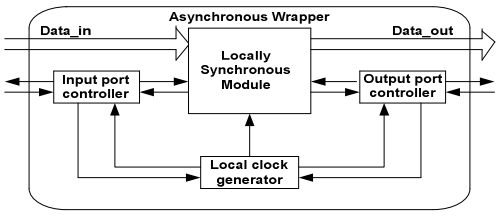

Fig. 1. Asynchronous wrapper of [6].

Based on the previous features, the synchronous IP modules become an optimum choice for design. Furthermore, GALS systems minimize the main drawbacks associated with global clock, reducing the problem of timing analysis and strongly encouraging the reusability of IPs [16]. GALS systems based on the wrapper interfaces, as seen in Fig. 1, are efficient in "point to point" architecture, as shown in Fig.2.

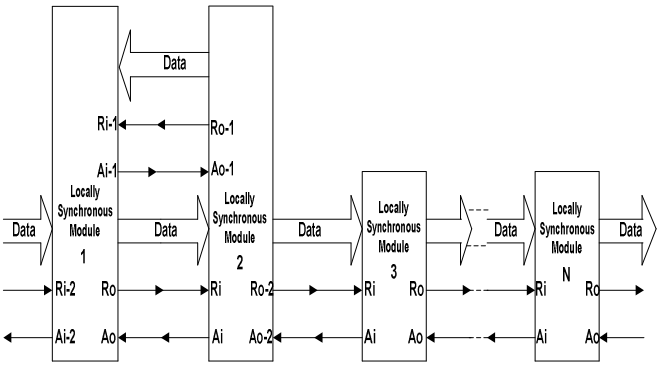

Fig. 2. GALS with point-to-point bidirectional data communication.

In many applications, system partitioning leads to a multi-point data communication between functional modules (see Fig. 3). Furthermore, functional modules operate sequentially, what means that only one module operates at a time. AW

978-1-4799-8332-2/15/$31.00 ©2015 IEEE

interfaces, as seen in Fig. 1, can be generalized to a multi-point architecture. In the architecture shown in Fig. 2 the wrapper of the functional module 1, presents two input ports and one output port. Altogether, there are eight ports in the first three modules. For GALS architectures with multi-point topology, the AW interface becomes very complex, dramatically increasing the area, once every point of communication relates to an input or to an output port [10].

This paper proposes a novel asynchronous interface with a locally-generated clock to GALS style (see Fig. 4). This interface allows multi-point GALS and, as advantage, it uses a single control point for all data communications. The proposed interface allows managing the activation and deactivation of sub-modules. A functional module can be decomposed into several sub-modules. For a case study, the proposed interface presented an average area reduction of 85% and 74% (products + literal) when compared with the asynchronous wrappers found in the literature [9] and [13], which were generalized to multi-point by adding input and output ports.

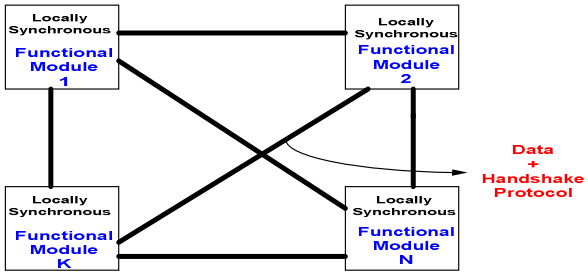

Fig. 3. GALS with multi-point data communication.

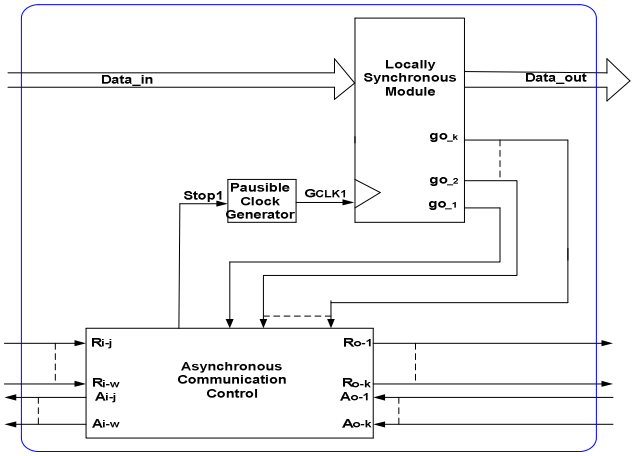

Fig. 4. Proposed asynchronous interface.

II. AN INTERFACE FOR GALS MULTI-POINT

A functional module can be described by a network of K controllers, where K ≥ 1, and W data-paths, being and K≥W and W≥0. Signals "go_1,…" are generated by the functional module when there is data transfer or a change of the sub-module. The proposed asynchronous interface is composed by three blocks (see Fig. 4): the "processing block" that is a functional module; the "clock block" that is locally generated [14], but may also be externally generated using gated-clock [15]; and the "data communication control block" that is responsible for the interaction between functional modules. In the generated multi-point architecture, while a functional module operates, the others are paused. The proposed asynchronous interface allows the partitioning of a synchronous module into synchronous sub-modules. Figure 5 shows the proposed architecture when a module is decomposed, for example, into two sub-modules, where each sub-module presents a pausible clock. The signal "go_1, go_2" is used to indicate which sub-module will be activated and to which functional module data will be sent.

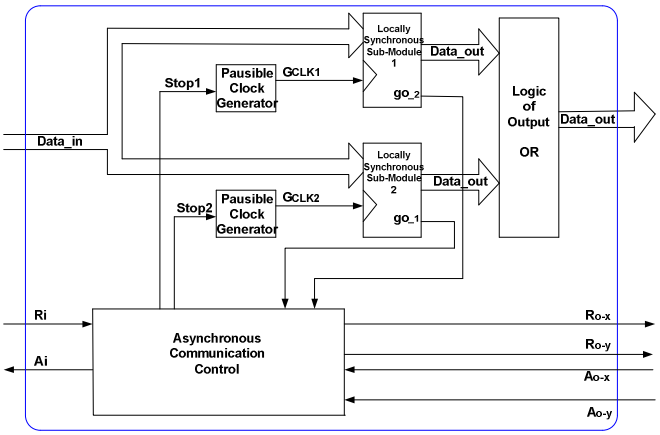

Fig. 5. Proposed asynchronous interface with sub-modules.

III. SYNTHESIS OF COMMUNICATION CONTROL

For each functional module of the digital system (implemented in multi-point GALS architecture with pausible clock) there is an asynchronous communication control. The synthesis of the communication control presents five steps:

1. Consider a digital system partitioned into N functional modules. If there is a functional module Mj, that contains sub-modules, go to step 2, otherwise go to step 3.
2. Find the interactions between the Mj sub-modules and set the signals "stop" and "go" in the data communication control.
3. Find the data interactions (communication) between functional modules using the "four phases handshake" protocol and set the signals "request" and "acknowledge" for each communication control.
4. Describe each one of the data communication control using burst-mode specification [17].
5. Synthesize each data communication control, for example, using 3D Tool [17].

IV. CASE STUDY

In order to demonstrate the synthesis method for data communication controls in a multipoint GALS system, we used a hypothetical digital system decomposed into three functional modules, interacting to each other as shown in Fig. 6. The three modules do not contain sub-modules, so each functional module has only a single pausible clock. Fig. 7

shows step 3, where the signals "request" (Ri and Ro) and "acknowledge" (Ai and Ao) were inserted following the data flow of the three modules. Step 4 generates the specifications of the three communication control following the sequence of the data flow of each module.

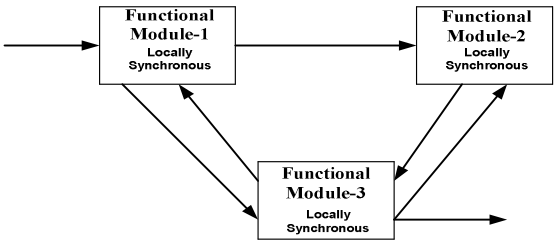

Fig. 6. Decomposition of functional modules.

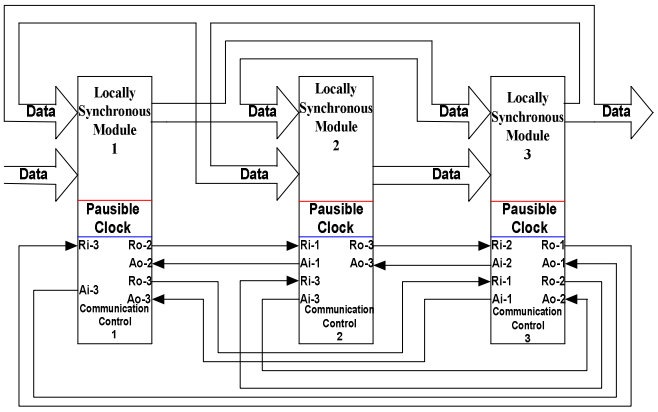

Fig. 7. Multi-point GALS architecture for case of study.

Fig. 8, 9 and 10 show respectively the burst-mode specification of the controls number 1, 2 and 3. They operate in the "Four-Phases handshake" protocol. For control number 1 the input signals are: go_{-2}, go_{-3}, Ao_{-2}, Ao_{-3}, Ri_{-3} and the output signals are Ro_{-3}, Ro_{-2}, $stop_1$, Ai_{-3}. It contains 7 states and 8 state transitions. For control number two, the input signals are: Ri_{-1}, Ri_{-3}, go_{-3}, Ao_{-3} and the output signals are: Ai_{-1}, Ai_{-3}, Ro_{-3}, $stop_2$. It presents 6 states and 7 state transitions. Considering control number 3, the input signals are: Ri_{-1}, Ri_{-2}, Ao_{-1}, Ao_{-2}, go_{-1}, go_{-2} and the output signals are: Ai_{-2}, Ai_{-2}, Ro_{-1}, Ro_{-2}, $stop_3$. It presents 8 states and 10 state transitions. The three controls were synthesized using 3D tool [17] and implemented in Huffman architecture with feed-back output [17].

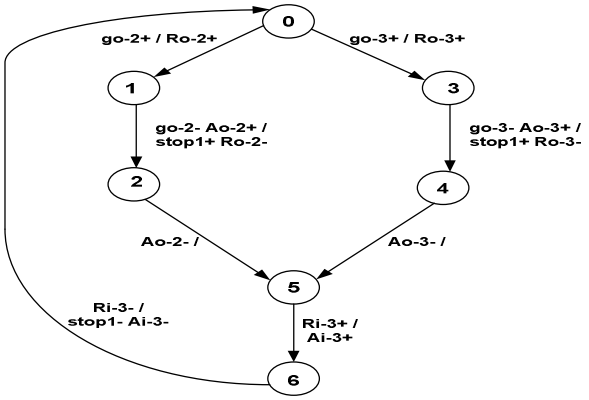

Fig. 8. Burst-mode specification: communication control of number 1.

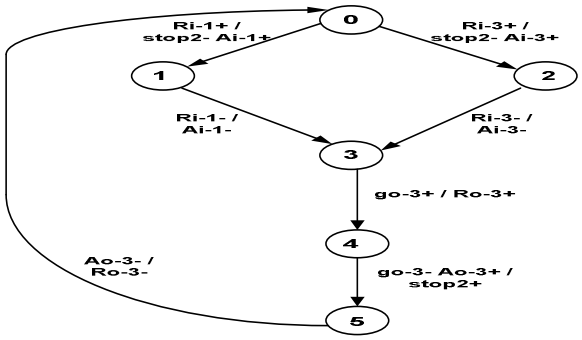

Fig. 9. Burst-mode specification: communication port of number 2.

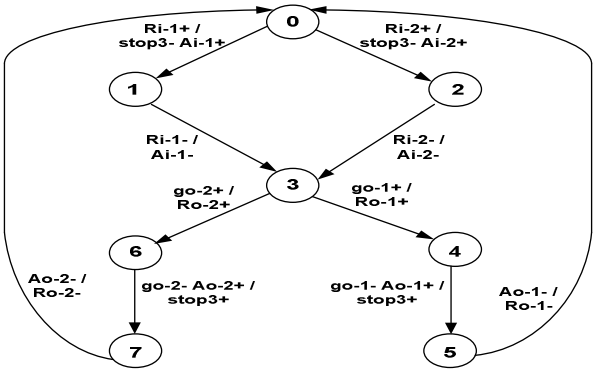

Fig. 10. Burst-mode specification: communication control of number 3.

V. SIMULATION & RESULTS

The simulation of the data communication control for the multipoint GALS architecture of Fig. 7 was performed by ALTERA tool, QUARTUS II software, version 9.1, Cyclone III family, and EP3C16F484C6 device [18]. Fig. 11 shows the simulation of multi-point GALS, where data communications controls comply with the specifications described in Fig. 8, 9 and 10. Therefore, the circuits are free of hazards.

In this section we compare AW interfaces areas of our proposed interface with the ones from [9] and [13], which were generalized to support multi-point data. The comparison is performed for the case study. Table I presents results (products and literals – sum-of-products) of the proposed data communication control, as well as the number of ports and the total number of products and literals of these ports (data communication between the three functional modules of the GALS system seen in Fig. 8). Table II presents the results of latency time of the controls for the case study.

Table I shows that the proposed data communication control achieved an average reduction of 86% in the number of products and 83% in the number of literals when compared with [9]. When compared with [13] it showed an average reduction of 78% in the number of products and 70% in the number of literals. The AWs interfaces from [9] and [13] needed 5 input ports and 5 output ports to perform the data communication of the case study.

Table II shows that the proposed data communication control achieved in the times of latency an average penalty on input port of 18% and an average reduction on output port of 5% when compared to [9]. Compared to [13] the proposed

data communication control achieved in the times of latency an average penalty in the input port of 2% and an average reduction in output port of 14%.

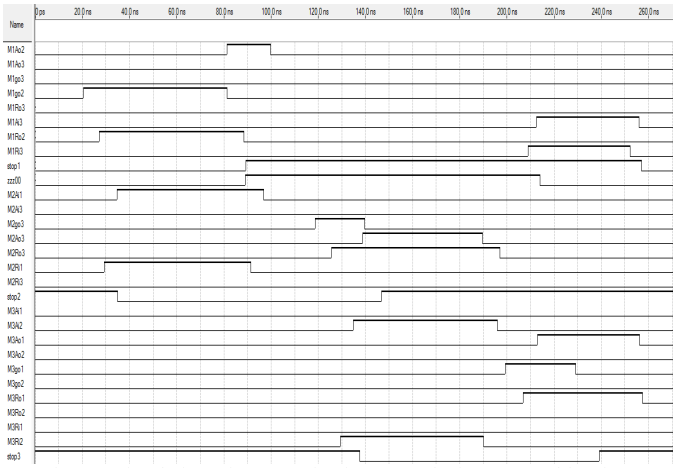

Fig. 11. Simulation of communication control of GALS multi-point.

TABLE I. RESULTS OF AREA ON CASE OF STUDY

Case of study	Asynchronous wrapper of [9]			Asynchronous wrapper of [13]			Asynchronous interface proposed	
	Nro. Ports	Number of Products	Number of Literals	Nro. Ports	Number of Products	Number of Literals	Number of Products	Number of Literals
Locally Synchronous Module-1	3	34	94	3	21	48	6	17
Locally Synchronous Module-2	3	29	64	3	18	42	3	11
Locally Synchronous Module-3	4	42	112	4	26	60	5	17

TABLE I.I. RESULTS OF LATENCY ON CASE OF STUDY

Case of study	Asynchronous wrapper of [9]		Asynchronous wrapper of [13]		Asynchronous interface proposed
	Time of Latency Port_input	Time of Latency Port_output	Time of Latency Port_input	Time of Latency Port_output	Time of Latency
Locally Synchronous Module-1	**30,1ns**	37.5ns	34.9ns	41.7ns	**30.2ns**
Locally Syncrhonous Module-2	**30,1ns**	**37.5ns**	34.9ns	41.7ns	37.8ns
Locally Synchronous Module-3	**30,1ns**	**37.5ns**	34.9ns	41.7ns	38.8ns

Table III shows the area results for point-to-point bi-directional GALS for the first three modules, considering the sequential operation (see Fig. 2). The proposed data communication control achieved an average reduction of 87% in the number of products and 86% in the number of literals when compared with [9]. Compared with [13] it achieved an average reduction of 78% in the number of products and 56% in the number of literals.

TABLE III. RESULTS OF AREA OF FIGURE 2

GALS of Fig. 2	Asynchronous wrapper of [9]			Asynchronous wrapper of [13]			Asynchronous interface proposed	
	Nro. Ports	Number of Products	Number of Literals	Nro. Ports	Number of Products	Number of Literals	Number of Products	Number of Literals
Locally Synchronous Module-1	3	29	74	3	18	42	**3**	**10**
Locally Synchronous Module-2	3	34	94	3	21	48	**5**	**13**
Locally Synchronous Module-3	2	21	56	2	13	30	**3**	**8**

VI. CONCLUSION

GALS systems showed to be an interesting design style for SoCs, but typical problems concerning to asynchronous interfaces, especially those for GALS systems with multi-point topology, shows to be a major drawback. Concerning to this situation, it was proposed a new architecture to asynchronous interface that showed to be able to overcome the previously discussed drawbacks, showing to be a good option for those designers who need to implement GALS systems in multi-point architecture. For future works, we will apply it on real multi-point GALS cases

REFERENCES

[1] C. Piguet, "Low-Power CMOS Circuits Technology, Logic Design and CAD Tools," Taylor & Francis Group, 2006.

[2] H. Hsieh, F. Balarin, and L. Lavagno , A. Sangiovanni-Vincentelli, "Synchronous approach to the Functional Equivalence of Embedded System Implementations," IEEE Trans. On CAD of Int. Circuits and Systems, vol.20, no.8, pp.1016-1033, August 2001.

[3] G. DE Micheli, "An Outlook on Design Technologies for Future Integrated Systems," IEEE Trans. on CAD of Integrated Circuits and Systems, vol28, no.6, pp. 777-789, June 2009.

[4] K. D. Muller-Glaser, G. Frick, E. Sax, and M. Kuhl, "Multiparadigm Modeling in Embedded Systems Design", IEEE Trans. on Control Systems Technology, vol. 12, no. 2, March 2004.

[5] D. M. Chapiro, *Globally-Asynchronous Locally-Synchronous Systems*, PhD thesis, Stanford University, October 1984.

[6] D. S. Bormann and P. Y. K., "Asynchronous Wrappers for Heterogeneous Systems," Proc. Int. Conf. Computer Design (ICCD), pp.307-314, October 1997.

[7] P. Techan, M. Greenstreet, and G. Lemieux, "A Survey and Taxonomy of GALS Design Styles," IEEE Design & Test of Computers, vol. 24, pp.418-428, September-October 2007.

[8] A. J. Martin and M. Nystrom, "Asynchronous Techniques for System-on-Chip Design*,"* Proc. of the IEEE, vol.94, no. 6, pp.1089-1120, June 2006.

[9] J. Muttersbach, "*Globally-Asynchronous Locally-Synchronous Architectures for VLSI Systems*," Ph.D. Thesis, ETH, Zurich, 2001.

[10] A. Reddy Ravi, "*Globally-Asynchronous, Locally-Synchronous Wrapper Configurations for Point-to-Point and Multi-Point Data Communication,*" Master thesis, University of Central Florida, 2004.

[11] J. Carlsson, "*Studies on Asynchronous Communication Ports for GALS Systems,"* Lic. Doctorate Thesis, Linkoping Studies in Science and Technology, Linkoping University, Sweden, June 2005.

[12] E. Amini, M. Najibi and H. Pedram, "Globally Asynchronous Locally Synchronous Wrapper Circuit based on Clock Gating," Proc. Emerging VLSI Technologies and Architectures, pp.193-199, 2006.

[13] J. Pontes, R. Soares, E. Carvalho, F. Moraes, and N. Calazans, "SCAFFI: an Intrachip FPGA asynchronous interface based on hard macros," 25th Int. Conf. on Computer Design, pp.541-546, 2007.

[14] D. L. Oliveira, L. A. Faria, and E. Lussari, "Design of an Improved and Robust Asynchronous Wrapper (AW) for FPGA Applications," Journal of Integrated Circuits and Systems, vol. 8, nro.1, pp.54-63, 2013.

[15] D. L. Oliveira, E. Lussari, S. S. Sato, and L. A. Faria, "An Asynchronous Interface with Robust Control for Globally-Asynchronous Locally-Synchronous Systems," Journal of Aerospace Technology and Management, vol. 5, nro. 1, pp.91-102, 2013.

[16] W. Hardt, M. Visarius, and B. Kleinjohann, "Architecture Level Optimization for Asynchronous IPs", Proc. 13th Annual IEEE Int. Conf. ASIC/SOC, pp.158-162, 2000.

[17] K. Y. Yun and D. L. Dill, "Automatic Synthesis of Extended Burst-Mode Circuits: Part I (Specification and Hazard-Free Implementation) and Part II (Automatic Synthesis)," IEEE Trans. on CAD of Integrated Circuit and Systems, vol. 18:2, pp. 101-132, February 1999.

[18] ALTERA Corporation-www.altera.com, 2014.