Synchronously Triggered GALS Design Templates Leveraging QDI Asynchronous Interfaces

W. Gul[&1], S.R. Hasan*[2], O. Hasan[&3], F. K. Lodhi[&4], F, Awwad[$5]

* Department of Electrical and Computer Engineering, Tennessee Technological University Cookeville, TN, USA.
[&]Sch. of Elect. Engg. and Comp. Sc., National University of Sciences and Technology (NUST), Islamabad, Pakistan
[$]College of Engineering, United Arab Emirates University, Al-Ain, UAE
Email: {[1]waqas.gul, [3]osman.hasan, [4]faiq.khalid}@seecs.edu.pk, 2shasan@tntech.edu, [5]f_awwad@uaeu.ac.ae

Abstract – **Single clock distribution over a large high performance chip can be very challenging. This led to evolution of globally asynchronous and locally Synchronous (GALS) systems in modern deep sub-micron (DSM) technology. In GALS mostly bundled data protocols which are based on handshake mechanism, are used for data transfer. But these protocols rely on timing assumptions between handshake signals and data values that causes timing closure problems, which poses strict constraints in system-on-chip (SoC) design. This work leverages quasi delay insensitive (QDI) designs to propose GALS design templates. This will facilitate the use of GALS systems in a conventional digital design flow with minimal intervention to interfacing modules. Modifications for two different quasi delay insensitive (QDI) asynchronous designs have been suggested, implemented and verified by using the proposed templates. Power, energy and latency have been compared for two different interfaces.**

Keywords— GALS; delay-insensitive; synchronization; multiple clock domains; system-on-chip

I. INTRODUCTION

The integrated circuit designs in modern nanometer technologies have become communication-centric [1]. To reduce time-to-market in modern technologies, it is desirable to integrate several intellectual property(IP) modules in a system-on-chip (SoC). Due to different clocking requirements of IPs, SoCs are usually divided into multiple clock domains (MCDs). This approach saves time by not redesigning of IP modules. A mechanism that can allow safe inter-module communication is needed to overcome clock timing constraints with minimal design interventions in the IP modules. IP modules in MCDs of a SoC are mutually asynchronous. Proposed inter-module communication techniques should be able to overcome the timing anomaly. Consequently, this leads to a paradigm shift from the globally synchronous design to globally asynchronous locally synchronous (GALS) designs [2]. GALS consists of asynchronous wrappers surrounding the synchronous modules to perform inter-module communication. GALS designs have been popular to the researchers due to their potential low-power consumptions as well [3].

Several different interfacing mechanisms to communicate between modules in a GALS system have been proposed [4]. A summary of the mechanisms of GALS interfaces can be found in the literature [5] [6]. These mechanisms can be broadly categorized into three different types: pausible clocking, self-timed FIFO based on conventional synchronizers and boundary synchronization based GALS techniques.

The choice of inter-module communication strategy is context dependent and literature has provided guidelines for their suitable scenarios [5] [6]. This work introduces the concept of hardware templates to reduce the IP design intervention which leads to an improved pausible clocking based GALS system using quasi-delay insensitive (QDI) asynchronous wrappers. The protocols for asynchronous wrappers can be divided into two main categories: bundled-data and quasi-delay insensitive. In bundled data protocols, request and acknowledgement signals are bundled with data signals. It shows improvement compared to the C-element and standard cell based designs [7][8]. Circuits based on bundled data protocol are prone to timing closure problem, which arises due to the interdependent timing of data and control signals [9]. The second type of protocol, QDI protocol is so far mostly focused on asynchronous circuits only. Quasi delay insensitive (QDI) protocol based interfaces are free from the timing mismatch problem, as request signal(s) is (are) embedded in the data. However, the hardware complexity increases as we move into more sophisticated QDI data encoding mechanisms. Recently designed QDI interfaces are promising in terms of performance and energy improvement for GALS [10].

This paper leverages the benefits of QDI asynchronous interfaces for GALS designs, with clock pausing triggered by synchronous modules, avoiding potential metastability problems. Hardware design template is proposed for 1-of-N data encoded GALS systems. The proposed template can be used with any asynchronous interface of the same class with nominal modifications in the asynchronous interface. This paper elaborates the flexibility of the design templates by implementing two different interfaces. Full sequence of signal transitions is provided with the required modifications in the asynchronous interfaces. These templates can have a number of exciting applications such as multi-processor SoCs (MPSoCs) [11]. Performance, power and complexity of all the used asynchronous interfaces are also presented.

This paper is organized as follows: Section II provides an overview of QDI asynchronous interfaces. Section III describes the mechanics of our novel GALS template. Section IV presents the proof of concept simulation results. Section V provides a general discussion on the context and evolution of this work. Finally, Section VI concludes this work.

II. OVERVIEW OF QDI ASYNCHRONOUS INTERFACE

QDI data codes are simple codes which do not contain any other codes in themselves so these codes can be received without any ambiguity [12]. Researchers have proposed a lot of other encoding schemes but 1-of-N is extensively used in on-chip communications, due to low hardware complexity.

One-of-N data encoding scheme is generally termed as one hot encoding in which only one bit is high, at a time, out of N

978-1-4799-5341-7/16/$31.00 ©2016 IEEE

bits. Table I shows 1-of-4 encoding scheme. One-of-N encoding scheme requires a $\log_2(N)$-to-N encoder for encoding at the sender end, and similarly, N-to-$\log_2(N)$ decoder for the decoding at the receiver side. Here, N is the number of wires in the system and $\log_2(N)$ is the number of bits per transaction. For 1-of-N encoding the number of wires increases exponentially (powers of 2) with the number of bits transmitted per data transaction.

TABLE I. 1-of-4 ENCODING SCEHEME

Two-bit value	X[0]	X[1]	X[2]	X[3]
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

III. Proposed GALS Hardware Templates for QDI Asynchronous Interface

A. *1-of-N Data Encoded QDI GALS Design Template (RTZ Signaling Schemes)*

Fig. 1 shows the architecture of the proposed GALS hardware template for 1-of-N data encoded QDI asynchronous interfaces, using return to zero (RTZ) signaling scheme. This design can be broadly divided into three sections: sender end, receiver end, and asynchronous switching interface, as shown in Fig. 1. The sending/receiving-end hardware consists of a synchronous sending/receiving module (SSM/SRM), which generates/receives N different 1-of-N encoded signals, labeled as RS0/RR0 to RS(N-1)/RR(N-1). The sending/receiving end further comprises of mutually exclusive elements, labeled ME_S/ME_R, coupled with a ring oscillator through the OR1/OR2 gate. ME_S/ME_R composed of two cross coupled NAND gates based upon two concurrent inputs and a filter which eliminates glitches at output, if one input occurs first then corresponding output also occurs first and vice versa. Req_Gen block is a simple TSPC based D -flip flop, based upon clock and ready signal as input with an asynchronous reset. The switching interface has two modes: idle and active. During idle mode it does not pass the incoming signals. During active mode, the incoming signals become available, to be latched by the SRM.

In Fig. 1, CLK1and CLK2 signals are clock signals for sender and receiver end, respectively. Whereas, interfacing mechanism is completely independent of these two signals. The asynchronous interface keeps the delay insensitive nature as the artificial delays for generating the Reset1 and Reset 2 signals does not affect the asynchronous operation of the interface block. To further understand Fig. 1, following sequence of operations is provided.

1) Sequence of Operations: This sub-section explains the initial conditions and sequence of operations of the proposed template shown in Fig. 1. The template for RTZ and single track handshaking schemes are identical, except for the pulser circuit to generate the Reset1 pulse, which is not required for single track handshaking. The sequence of operations leading to one transaction is as follows;

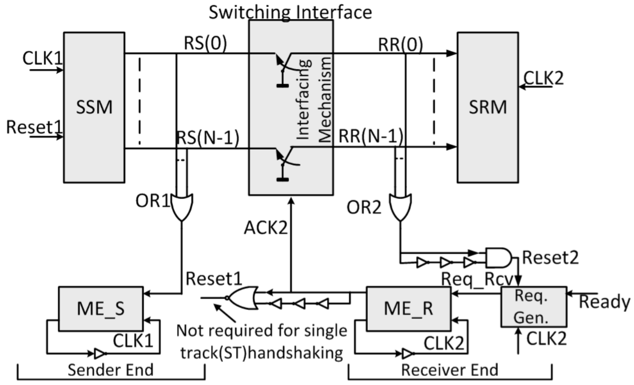

Fig. 1. Proposed GALS template for 1-of-N data encoded QDI asynchronous interfaces, using RTZ signaling schemes

(a) Initially, it is assumed that all the output signals of SSM (RS0 to RS(N-1)) and input signals (barring the CLK2 signal) to SRM (RR0 to RR(N-1)) are at logic 0. This is consistent with the RTZ signaling scheme.

(b) According to 1-of-N encoding, only one of the N signals is asserted to logic 1 at a time.

(c) OR1 senses this signal and requests ME_S to stop generating CLK1. CLK1 restarts only when the Reset1 pulse resets the SSM, and hence the de-assertion of OR1 signal releases the CLK1 signal.

(d) At the receiver end, the Ready signal is latched at Req_Rcv, which then stops the clock signal CLK2 and generates ACK2 signal.

(e) The ACK2 signal enables the switching interface, when one of the RS0 to RS(N-1) signal is asserted, the corresponding RR0 to RR(N-1) signal also gets asserted. *(f)* Reception of RR(x) signal, where x = 0 to N-1, OR2 resets the Req. Gen. through a pulser circuit, and Req_Rcv falls to logic 0.

(g) This releases CLK2 and de-assert ACK2 signal.

(h) The termination cycle begins at the switching interface, and at the sender end, the termination cycle starts with the de-assertion of the ACK2 signal. For the RTZ scheme, the pulser circuit sends an acknowledge pulse to the sender, shown as Reset1, which de-activates RS(x) signals. Concurrently, the ACK2 signal triggers the switching interface to de-assert RR(x), completing the RTZ signaling scheme.

2) Switching Interface: It is stated in the Section I that the modifications required to utilize the interfaces in some application specific context are minimal. To support this claim, this sub-section describes the required modifications to asynchronous interfaces to utilize template of Fig. 1. Modified GAsP [15] asynchronous interface is shown in Fig. 2.

One copy of this block links RS(x) to corresponding RR(x) and acts as a switch in the central block of Fig. 1. The modifications to the interface are the additional AND gate and delay element, shown as shaded in Fig. 2. The sequence of operations of this interface is as follows: initially, the signals RR(x), RS(x) and ACK2 are at logic 0, making Q5 and Q1 on. Node D is pre-charged through Q3. When the sender signal RS(x) becomes high, the AND gate waits for the receiver to be ready to receive the data. In Fig. 1, a logic 1 at the ACK2 signal indicates that receiver is ready to receive, and hence AND gate turns on Q2 and inv1 turns off Q5. Q2 and Q1 discharge D, which in turn make RR(x) high through Q4. Both Q3 and Q4 are turned OFF, as Q3 charges D again. RR(x) remains high, until

ACK2 discharges RR(x) again and makes the interface ready for the next transaction. The delay line at the bottom of Fig. 2 is required to allow the SRM, in Fig. 1, enough time after the resumption of CLK2 (Fig. 1), to latch the RR(x). Since this requires only local delay adjustments at the receiver end, therefore, this is deterministic in nature, and hence, sanctity of delay insensitivity is maintained.

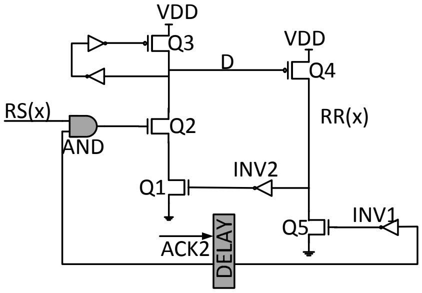

Fig. 2. Modified GAsP implementation (inspired from [15])

B. *1-of-N Data Encoded QDI GALS Design Template (ST Signaling Schemes)*

The template for ST signaling scheme is identical to Fig. 1 with few exceptions, and this is elaborated as follows: ST signaling scheme, by design, does not require the additional acknowledge wire to traverse from the receiver end to the sender end [4]. Hence, this modification is identified using an arrow in Fig. 1, where it states that the signal is not required for ST Handshaking. Fig. 3 shows the modified single track full buffer (STFB) [13] implementation, which is used as the asynchronous interface mechanism in the design template of Fig. 1. In the ST signaling scheme, a pulser circuit within the sender end senses the termination of the handshaking scheme (e.g. through the ST handshake interface signal assertion), which are again local to the sender end. This pulser circuit generates the Reset1 signal, which controls the mechanism of negating request line, hence avoids acknowledge line. For further details on STFB please refer to [13].

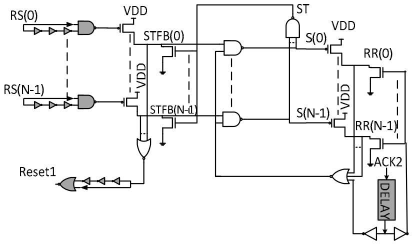

Fig. 3. Modified STFB implementation for single track (ST) handshaking

Switching Interface: For the ST handshaking interface template, we used the single-track full buffer (STFB) [13] interfacing scheme. The sequence of operations is provided here to understand the adaptation needed, which requires very little modification to the interface. The sequence is as follows: initially, in Fig. 3, it is assumed that RS(x), ACK2, STFB(x) and RR(x) are at logic level 0. Consequently, the NOR gate output and S(x) are at logic 1 and ST is at logic 0. Therefore, all the NMOS transistors connected with STFB(x) and RR(x) are OFF. When any of the RS(x) is set to logic high, the pulser circuit generates a pulse, and subsequently the corresponding STFB(x) is charged. Once the receiver is ready, ACK2 becomes high, then the corresponding NAND gate turns the respective PMOS on, and the respective S(x) output becomes low. This in turn makes that particular RR(x) high, which switches the NOR output back to low, and hence any further transitions in RS(x) are blocked. In Fig. 1 it can be seen that as soon as RR(x) is asserted, the Reset 2 signal restarts CLK2, and hence ACK2 is negated that passes through a delay to allow the SRM signal to latch the RR(x) signal. Subsequently, the negation of ACK2 discharges the RR(x), and thus, the interface is reinstated to its initial state. All the above explained modifications in STFB are shaded in Fig. 3.

IV. SIMULATION RESULTS

This section describes proof of concept simulations that are performed to explore the characteristics and performance of the proposed design template. All simulations are performed using IHP 90nm CMOS process technology (at 27° C). Fig. 4 and 5 shows the simulated waveforms for 1-of-N data encoded QDI GALS template (Fig.1) with RTZ and ST signaling schemes, respectively. RTZ scheme used modified GAsP (Fig. 2), whereas ST scheme used modified STFB (Fig. 3). After verifying the functional correctness of each design, we further analyzed these simulations for different metrics. Fig. 4 and 5 provide simulation results to measure latency in modified GAsP (Fig. 2) and modified STFB (Fig. 3), respectively.

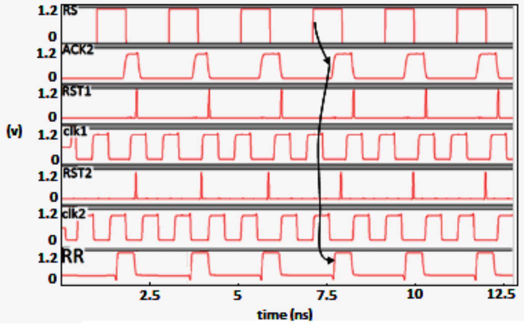

Fig. 4. Simulation results for 1-of-N data encoded QDI GALS template with RTZ scheme, using modified GAsP interface

The signals involved in measuring the latency and their causal relationship is illustrated with arrows in Fig. 4 and 5. The latency in these figures is defined as time taken from signal RS, to the data bit received at the corresponding RR register. It is observed that for this set of simulations, GALS system worked without pausing the clock for frequency up to 1.2 GHz. Total latency introduced by modified GAsP interface is approximately 750 psec, and 710 psec with the modified STFB interface. Proposed template is also simulated for other set of frequencies i.e. above 1.2 GHz but clock is paused during the data transfer from SSM to SRM.

Table II provides frequency, latency, power and energy values for each simulated case. The first column of Table II shows the case when the number of bits in a data transaction is 4, and the second column shows a 32 bits data transaction. The letter ‘X’ in the power columns indicates additional power required for encoding and decoding. The 8X in the 32 bit column of the row for power metric indicates that at least eight times the power is required for two 4-to-16 decoders and two 16-to-4 encoders, compared to their respective 2-to-4 and 4-to-2 counterparts. The symbol Y, in the latency for 32-bit case, is the additional delay needed for two 1-of-16 Data encoders. This delay is due to the additional Fan-out requirements (OR1 and OR2 as in Fig. 1). Similarly, the symbol Z in these tables is the additional energy consumption associated with these interfaces.

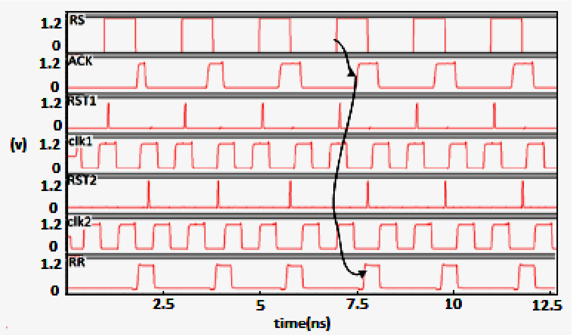

Fig. 5. Simulation results for 1-of-N data encoded QDI GALS template with ST scheme, using modified STFB interface

Table II. Simulation results of the GALS template for with both the interfaces using 1-of-N encoding

Measurement	4-bits (GAsP)	4-bits (STFB)	32-bits (GAsP)	32-bits (STFB)
Clock Frequency (GHz.) (without pausing)	1.19	1.14	1.19	1.14
Latency (ns)	.75	.71	.75+Y	.71+Y
Power (mW)	2.75+X	2.45+X	22+8X	19.6+8X
Energy(mJ)	14.17	4.83	14.17+Z	4.83+Z

It is observed that modified STFB interface consumes least power as compared to the modified GAsP. Modified STFB is also efficient in latency and energy requirement as compared to the modified GAsP. However, maximum clock frequency with no pause is a lower for the modified STFB.

V. DISCUSSION

The first major objective achieved in this work is to successfully prove that QDI asynchronous interfaces can be utilized efficiently in GALS design. Secondly, we have characterized two different interfaces based on complexity and performance so that a designer may make an informed decision on choosing a particular interface. The approach developed in this paper also provides a flexible interfacing mechanism for MPSoC based ASIP. Our technique can be combined with the configurable platform technique [14] to automate the generation of interfaces in the platform. Comparison between two interfaces i.e. modified GAsP and modified STFB performance is made. Another comparison is performed to the different templates under the same constraints and reported elsewhere [16][17].

VI. CONCLUSION

GALS design templates were introduced, to leverage from the robust QDI asynchronous interfaces. Two different asynchronous interfaces, utilizing RTZ and ST signaling schemes are analyzed. Robustness of the proposed design template is observed using several different frequency combinations. Our template based approach reduces the design complexity of IP based GALS design and the results provided on several metrics allow designers to make an informed decision to choose the GALS interface that fits best to the design need.

REFERENCES

[1] Semiconductor Industry Association, “International technology roadmap for semiconductors (ITRS).”, 2015.

[2] D. Chapiro, “Globally Asynchronous Locally Synchronous Systems”, Ph.D. thesis, Stanford University, October 1984.

[3] K. S. Chong, K. L. Chang, B.-H. Gwee and J. S. Chang,“Synchronous-Logic and Globally-Asynchronous-Locally-Synchronous (GALS) Acoustic Digital Signal Processors”, In IEEE journal of solid state circuits, Volume: 47, Issue No: 3, March 2012, pp. 769-780.

[4] M. D. Clercq and R. Negulescu, “1.1-GDI/s Transmission between Pausible Clock Domains”, In Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS‘02), Volume: 2, Phoenix-Scottsdale AZ, May 2002, pp. 768-771.

[5] S. R. Hasan, N. Bélanger, Y. Savaria, and M. O. Ahmad, "All digital skew tolerant synchronous interfacing methods for high-performance point-to-point communications in deep sub-micron SoCs", Integration, the VLSI journal, Volume: 44, Issue no: 1, 2011, pp. 22-38.

[6] S. R. Hasan, "Inter-module Interfacing techniques for SoCs with multiple clock domains to address challenges in modern deep sub-micron technologies." PhD diss., Concordia University, 2009. Available on line at:http://spectrum.library.concordia.ca/976728/1/NR67368.pdf

[7] X. Fan1, M. Krstić1, E. Grass1, B. Sanders2, and C.Heer,“Exploring Pausible Clocking Based GALS Design for 40-nm System Integration”, In Proceedings of Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, IEEE, March 2012, pp. 1118-1121.

[8] J.M. Chabloz and A.Hemani, “Low-Latency Maximal-Throughput Communication Interfaces for Rationally Related Clock Domains”, In Proceedings of IEEE transaction on Very Large Scale Integration (VLSI) systems (Volume:PP , Issue: 99), IEEE, April 2013, pp. 641-654.

[9] A. Upadhyay, S. R. Hasan, and M. Nekili, “A Novel Asynchronous Wrapper Using 1-of-4 Data Encoding And Single Track Handshaking”, In Proceedings of 2nd IEEE North-East Workshop on Circuit and Systems (NEWCAS‘04), IEEE, June 2004, pp. 205-208.

[10] Moreira, M.T.Guazzelli and N. L. V. Calazans, “Return-to-one protocol for reducing static power in C-elements of QDI circuits employing m-of-n codes”, In Proceedings of 25th Symposium on Integrated Circuits and Systems Design (SBCCI), Brasilia, IEEE, Aug-Sept 2012, pp. 1-6.

[11] S.Höppner, H.Eisenreich, S.Henker, D.Walter, G. Ellguth and R.Schüffny, “A Compact Clock Generator for Heterogeneous GALS MPSoCs in 65-nm CMOS Technology”, In Proceedings of IEEE transaction on Very Large Scale Integration (VLSI) systems, Volume: 21, Issue no: 3, March 2013, pp. 566-570.

[12] T. Verhoeff, “Delay-insensitive codes--an overview”, Computing Science Notes, Department of Mathematics and Computing Science Eindhoven University of Technology The Netherland , January 1987.

[13] M. Ferretti, P.A. Beerel, “Single-track asynchronous pipeline templates using 1-of-N encoding”, In Proceedings of Design Automation and Test in Europe Conference and Exhibition, (DATE‘02) , IEEE, 2002, pp. 1008-1015.

[14] D. Ayachi, Y. Savaria, C. Thibeault, “A configurable platform for MPSoCs based on application specific instruction set processors”, In Proceedings of 7th IEEE North-East Workshop on Circuit and Systems (NEWCAS-TAISA‘09), Toulouse, IEEE, June-July 2009, pp. 1-4.

[15] Huang, Shuang, Chun-Jie Zhou, Shuang-Hua Yang, and Yuan-Qing Qin. "Cyber-physical system security for networked industrial processes."*International Journal of Automation and Computing* 12, no. 6 (2015): 567-578.

[16] W. Gul, S.R. Hasan and O. Hasan, “Clock Domain Crossing (CDC) for inter logic layer Communication in 3-D IC”, In Proceedings of 12th NEW Circuit and System (NEWCAS), IEEE, Trios Rivieres QC, Canada, June 2014, pp. 476-479.

[17] W. Gul, S.R. Hasan and O. Hasan, “Yeild Aware inter logic layer communication in 3-D IC: Some Early Design Stage Recommendations”, In Proceeding of 57th IEEE Internation Symposium on Circuit and Systems (MWSCAS), College Station TX, USA, August 2014, pp. 222-225.