

Comparing GALS Architectures and Communicational Protocols

S. O. Bykov

dept. of Computer Engineering
Vladimir State University
Vladimir, Russia
e-mail: sobykov@gmail.com

S.G. Mosin

dept. of Computer Engineering
Vladimir State University
Vladimir, Russia
e-mail: smosin@vpti.vladimir.ru

Abstract—This paper describes the existing GALS (Globally Asynchronous Locally Synchronous) architectures and communicational protocols with respect to their applications for meeting concrete requirements. So it should help in decision-making during designing asynchronous systems.

Keywords—GALS systems, pausable clocks, handshake protocols.

I. INTRODUCTION

The problem of distributing the global clock in a chip with minimal clock skew is getting difficult to solve due to the increasing complexity of digital circuits. Additionally the integration of complex systems on chip (SoC) requires a multitude of clock frequencies to be integrated on a common die. A fundamentally different synchronization strategy is used in asynchronous design methodologies. So far as synchronous digital design is well understood and the design methodology and flow are established, it is more effective to combine this two strategies and get advantages of both of them. This idea has been realized in the GALS (Globally Asynchronous Locally Synchronous) approach.

A GALS system consists of complex digital blocks operating synchronously. Those blocks are usually developed using standard synchronous CAD tools and design flow. However, the operation of the blocks is not mutually synchronized—that why the term *locally synchronous* is used. The locally synchronous blocks communicate with one another asynchronously; on the block level (globally), the system is asynchronous. A common approach is to add an asynchronous wrapper, which provides an interface from the synchronous environment to the asynchronous one (and vice versa), to every locally synchronous block.

There are three main strategies for implementing GALS systems: pausable clocking, FIFO-based approach, and boundary synchronization [1]. All of them have their own advantages and should be applied in special cases.

The most popular communication protocol is handshake protocol [2]. But also there are some other protocols, for example, protocols based on clock transfers from sender to

receiver, and choice of protocol to be used should be done for each concrete design. There are three main requirements for GALS systems: throughput, area consumption, and power consumption. Meeting all of them is often impossible, so one should choose the most important factor and use it for making decisions during design process. This paper presents the solution which might help to meet each requirement.

II. THROUGHPUT

For systems, processing big streams of information and therefore requiring good throughput, the optimal choice is a FIFO-based solution. This approach uses asynchronous FIFO buffers between locally synchronous blocks to hide the synchronization problem. A SoC architecture that uses distinct clock domains connected through bisynchronous FIFO buffers is commonly called a GALS system. Such systems can tolerate very large interconnect delays and are also robust with regard to metastability - a state that doesn't settle into a stable '0' or '1' logic level within the time required for proper operation. Designers can use this method to interconnect asynchronous and synchronous systems and also to construct synchronous-synchronous and asynchronous-asynchronous interfaces. Figure 1 diagrams a typical FIFO interface.

The advantage of FIFO synchronizers is that they don't affect the locally synchronous module's operation, therefore the FIFO-based approach allows achieving high throughput. However, with very wide interconnect data buses, FIFO structures can be costly in terms of silicon area. Also, they require specialized complex cells to generate the empty/full flags used for flow control [1]. Another disadvantage of FIFO-based solution is high power consumption. That is why this architecture cannot be effective in mobile systems.

In such kind of systems a standard protocol for working with FIFO is generally used: sender writes data to FIFO and receiver reads it. But there are some systems, for example the DSP platform described in [3], where data transmission is performed through communication network. Such solutions use communication protocol based on clock transfers from sender to receiver. Sender sends clock signal with data and these data are written to FIFO, clocked on write side by sent

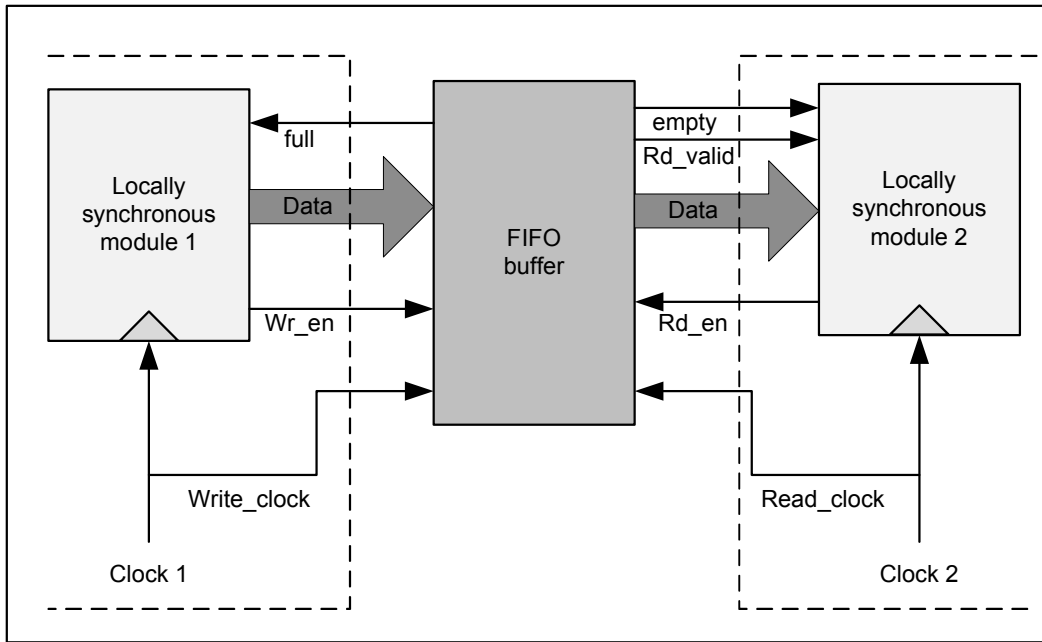


Figure 1. Typical FIFO-based GALS system

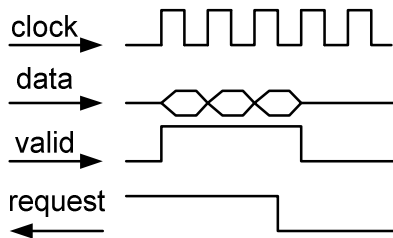


Figure 2. Timing diagram for protocol based on clock transfers

clock. Receiver reads these data from read side of FIFO. Timing illustration for this protocol is presented on Figure 2.

III. AREA OVERHEAD AND POWER CONSUMPTION

Area overhead and power consumption can be considered together, because power consumption is generally related to area if no special techniques are used. The most effective solution for these requirements is pausable clocking. This approach is described in [4]. Figure 3 illustrates the general structure of such system. The basic idea of this approach is transferring data between wrappers when both the data transmitter and data receiver clocks are stopped. This elegantly solves the problem of synchronization between the two clock domains [1]. But throughput of this solution strongly depends on data transfers rate. If the rate is high, frequent clock pauses will practically stop work of

synchronous block. This problem is solved in boundary synchronization approach, where data synchronization at the borders of the locally synchronous island performs without affecting the inner operation of locally synchronous blocks and without relying on FIFO buffers. This method can achieve very reliable data transfer between locally synchronous blocks. On the other hand, such solutions generally increase latency and reduce data throughput, resulting in limited applicability for high-speed systems [1].

Pausible clocking and boundary synchronization approach require programmable ring oscillators. This is an inexpensive solution that allows full control of the local clock. However, it has significant drawbacks. Ring oscillators are impractical for industrial use. They need careful calibration because they are very sensitive to process, voltage, and temperature variations. Moreover, embedded ring oscillators consume additional power through continuous switching of the chained inverters.

Solutions, based on pausable clocking, use standard handshake protocol and its different modifications. Existing asynchronous handshake protocols are bundled data protocol, dual-rail data protocol, 1/N data protocol and single-track data protocol [5]. Timing illustration for bundled data protocol is presented on Figure 4. Req and Ack signals need to be changed two times in one transmission cycle, so it is much slower. It allows reusing existing synchronous units, and it can be implemented in a small area, but fail to conquer electromagnetic interference (EMI) [5]. The advantage of dual-rail data protocol is that the data-

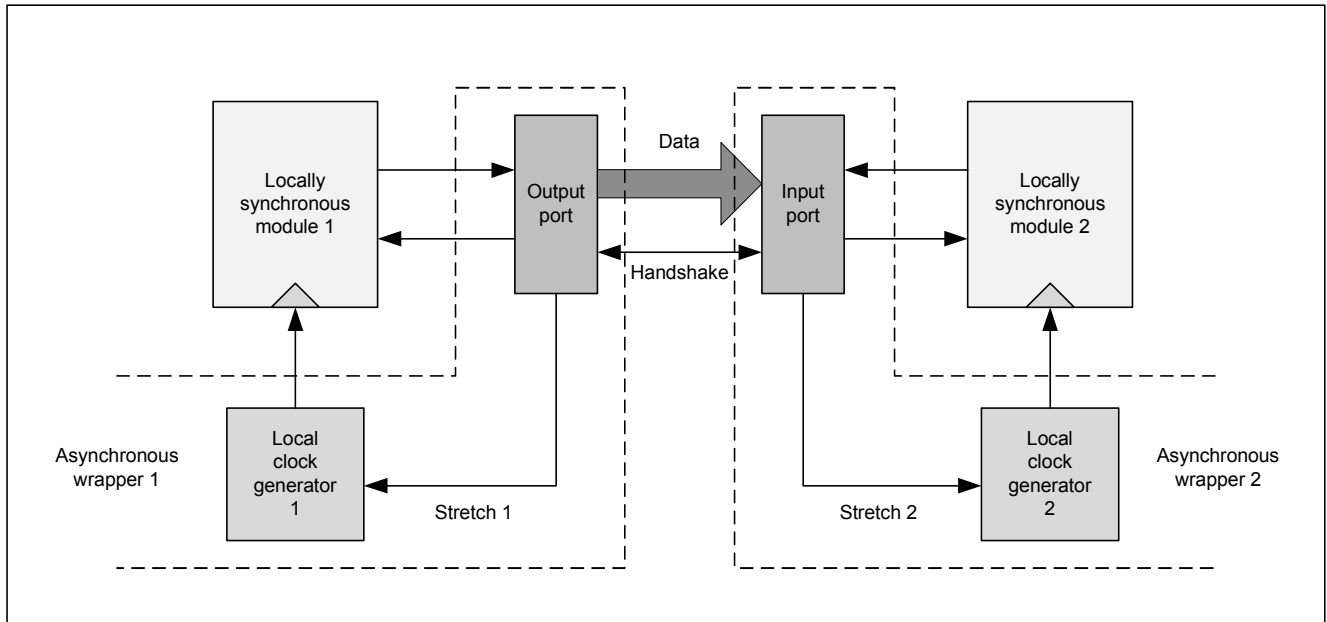


Figure 3. GALS system with pausable clocking

validation information can be carried on data themselves, that there is no need to use Req signal to denote data validation, thus avoiding the delaymatching efforts brought by the complex clocking relationship between req signals and ack signals. Dual-rail data protocol has a better anti-EMI capability due to the fact that two lines represent one data. However, the protocol implementation requires extra chip area (nearly twice as large as the bundle data protocol does) [5].

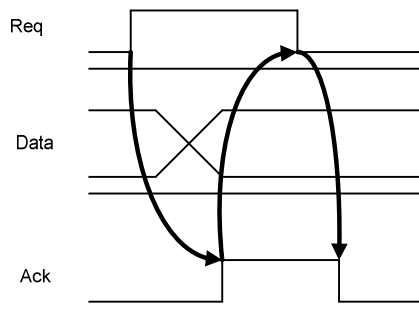


Figure 4. Timing diagram for bundled data protocol

IV. APPLICATION OF GALS SOLUTIONS

GALS systems can be used for video coding/decoding. MPEG-4 decoder is example of system, which requires a high throughput. All systems, worked with video, must

process big information streams. So, it is necessary to use FIFO-based solution for such designs, because other architectures cannot provide needed throughput.

V. CONCLUSION

GALS systems are roughly developing direction in a modern science. This paper presented some methods for design such systems. In some cases it may be not optimum, but generally these solutions can be used for various types of designs.

REFERENCES

- [1] Miloš Krstić, Eckhard Grass, Frank K. Gürkaynak, Pascal Vivet, "Globally Asynchronous, Locally Synchronous Circuits: Overview and Outlook", IEEE Design & Test, v.24 n.5, pp.430-441, September 2007, doi:10.1109/MDT.2007.164
- [2] Joep L. W. Kessels, Ad M. G. Peeters, Paul Wielage, Suk-Jin Kim, "Clock Synchronization through Handshake Signalling", Eighth International Symposium on Asynchronous Circuits and Systems (ASYNC'02), IEEE Computer Society Press, April 2002, doi: 10.1109/ASYNC.2002.10002
- [3] Anh Tran, Dean Truong and Bevan Baas, "A GALS Many-Core Heterogeneous DSP Platform with Source-Synchronous On-Chip Interconnection Network," ACM/IEEE International Symposium on Networks on Chip (NOCS), San Diego, CA, USA, May 2009, pp. 214-223.
- [4] K.Y.Yun, R.P.Donohue "Pausible Clocking: A First Step Toward Heterogenous Systems", Proc. International Conference on Computer Design (ICCD), IEEE Computer Society Press. 1996, pp. 118-123.
- [5] Xuguang Guan, Duan Zhou, Dan Wang, Yintang Yang, Zhangming Zhu "A Novel GALS Single-Track Protocol Asynchronous Communication Circuits", Pacific-Asia Conference on Circuits, Communications and System' 2009(PACCS'09), May 2009, pp. 269 - 272.