Asynchronous Circuit Design & Test – A Tutorial

Ney L. V. Calazans, Alexandre M. Amory, Matheus T. Moreira, Felipe A. Kuentzer

Pontifical Catholic University of Rio Grande do Sul (PUCRS) Faculty of Computer Science (FACIN) – Porto Alegre, Brazil
{ney.calazans, alexander.amory}@pucrs.br {matheus.moreira, felipe.kuentzer}@acad.pucrs.br

***Abstract*— Silicon technologies advances brought the possibility of integrating billions of transistors in a VLSI die. However, as these transistors become smaller due to fabrication method enhancements, some aspects that were negligible in older technologies emerge as difficulties for the design in current and future technology nodes. In this context, fully synchronous digital circuits become harder to build and test. For example, clock distribution may become an overwhelming design problem. Also, variability issues may require margins that render synchronous implementations prohibitive or very expensive. Asynchronous or clock-less design paradigms recently attracted new interest from research communities, due to their improved ability to cope with issues like design timing closure and variability. This tutorial approaches the most relevant traditional and modern methods for the design and test of asynchronous digital circuits. The emphasis is on the infrastructure required to support the design of such circuits, conditioned to a chosen design method.**

I. INTRODUCTION

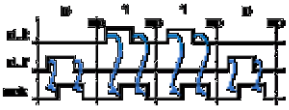
The synchronous paradigm prevails in the digital circuit design arena. This is due to the high abstractions levels achieved when considering time as a discrete variable controlled by a *clock* signal, the main characteristic of the paradigm. However, as silicon technologies evolve into deep-submicron nodes, synchronous design becomes more constrained [1]. In fact, the ITRS [2] in its 2011 edition already suggested that further advances in VLSI systems need a model shift. By 2020, it expected that 40% of the intrachip communication use asynchronous mechanisms. Thus, asynchronous techniques need to gain relevance and attention.

After several decades of evolution, electronic design automation tools (EDA) are mostly adapted to follow synchronous assumptions only. Consider the high-level tools used to capture and validate digital circuit descriptions. Tasks as simple as the logic simulation of a set of gates with one or more combinational feedback wires are often not possible! This is also true for the automatic synthesis of circuits from hardware description language (HDL) models in, say Verilog or VHDL. The absence of explicit synchronization points based on temporal barriers controlled by a special signal can easily prevent to correct translation of HDL models to hardware. This is reflected at the HDL level by the imposition of using special language constructs, which in fact create a synthesizable language subset accepted by synthesis tools that can only produce synchronous circuits. At the opposite end of the design spectrum, physical synthesis, the situation is similar. Those that already tried to construct asynchronous circuits on reconfigurable devices (as described in e. g. reference [3]) know how hard it is to obtain useful results. The same applies to the design of asynchronous circuits in the form of integrated circuits (ICs) or IC modules in the form of Intellectual Property (IP) Cores. This tutorial is an overview of asynchronous circuit design and test vast domain, coupled with some details of recently proposed developments for enabling these technologies to produce competitive digital circuit designs.

II. REASONING FOR THE PROPOSAL AND AUTHORS

Undergraduate as well as graduate curricula in Computer and Electrical Engineering abound with courses on digital design, but the number of such courses that approach the design of non-synchronous digital circuits is, to say the least, rare. However, the gains in design optimization that can be achieved with asynchronous design techniques is large, if not for any kind of circuit, for many distinct applications domains. For example, high speed memory cells and memory controllers are most often asynchronous circuits [4]. Besides, every synchronous system interacts with the asynchronous world through synchronization interfaces, which are typically asynchronous circuits [5]. Las but not least, the recent push towards the Internet of Things (IoT) will require billions of new, low end devices that operate under very low supply voltages, in near- or subthreshold transistor operation regions. This is a feat hard to adapt to synchronous design power overheads but naturally handled with asynchronous design [6]. A tutorial on asynchronous circuit design and test can thus be instrumental to fill such gaps in the mentioned curricula. Experienced designers and researchers can also benefit from the concepts covered in the tutorial, since these is hardly dealt with during usual digital circuit design activities.

The authors of this proposal have worked with several aspects of asynchronous design and test along the last ten years. From early works on asynchronous interfaces [3] and asynchronous circuit prototyping in FPGAs [7], their research evolved to provide design infrastructures for asynchronous design in the form of IC standard cell sets [8] [9], to propose new methods and templates to design asynchronous circuits [10] [11], and to suggest new design flows and electronic design automation (EDA) tools supporting asynchronous design [12] [13], among other contributions. The GAPH also cooperates with other research groups, to propose new design methods and to improve the testability of such methods as demonstrate e. g. references [14], [15] and [16]. The proponents of the tutorial have a long-term experience in asynchronous circuits design and test, and can thus provide the audience with a taste of their deep knowledge on the subject. The next four Sections give a very brief technical overview of the themes the tutorial is expected to cover.



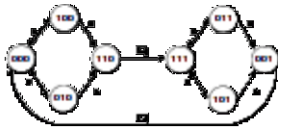
(a)

Value	d.t	d.f
Spacer	0	0
Logic 0	0	1
Logic 1	1	0
Invalid	1	1

(b)

A	B	Q_i
0	0	0
0	1	Q_{i-1}
1	0	Q_{i-1}
1	1	1

(c)



(d)

Figure 1 – (a) dual-rail (DR) transmission; (b) DR encoding; (c) a C-element truth table; (d) C-element state diagram (values inside states are in order ABQ_i).

III. Asynchronous Circuit Models and Templates

This part of the tutorial addresses some of the main concepts behind asynchronous circuits. Advantages and inconveniences of these circuits compared to synchronous ones, models that need to be employed in asynchronous circuit design and modeling, and special components required for their implementation are some of the topics to cover. To exemplify, consider some basic definitions and concepts. A digital circuit is *synchronous* if its operation is controlled using a single clock signal. Otherwise it is called *asynchronous*. Asynchronous circuits use explicit handshaking among their components to synchronize, communicate and operate [1]. Asynchronous design methods are called here *design templates* or simply *templates*. Templates are usually grouped into families. Most practical templates in use today are either in the *quasi-delay-insensitive* (QDI) family or in the *bundled-data* (BD) family. These will be defined and explored along the tutorial. Figure 1 shows some basics of a QDI template. Figure 1(a) is an example of data transmission using two wires to represent a bit of data, with the interpretation in Figure 1(b). Figure 1(c) and Figure 1(d) in turn show the behavior of an asynchronous component (or asynchronous gate) that can be used in a QDI template, the C-element.

IV. Cells, Cell Libraries and Cell Design Flow

Asynchronous design often requires special components like C-elements, NCL gates, etc [1]. No commercial standard cell library provides these today. This part of the tutorial will show the need for these components, and study how such components can be implemented efficiently, providing an overview of the ASCEnD flow [8] [9], proposed by the authors to solve the problem of developing enhanced cell libraries with the flow illustrated in Figure 2.

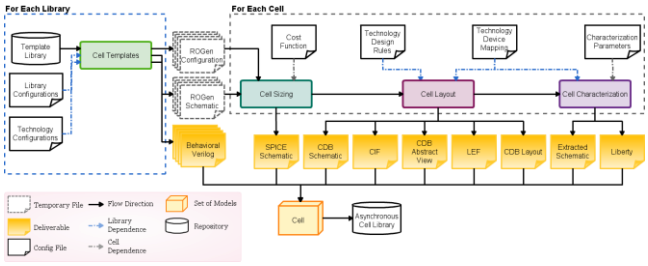


Figure 2 – Overview of the ASCEnD asynchronous library design flow.

V. The SDDS-NCL and the BLADE Templates

The number of distinct asynchronous design templates available in the literature is quite large (several dozen templates can be identified). To illustrate the design of different classes of asynchronous circuits the tutorial will explore two recently proposed templates, the SDDS-NCL template [10] [11], proposed by the GAPH research group and the BLADE template proposed in the University of Southern California with the participation of members of the GAPH group [14] [16].

VI. The Test of Asynchronous Circuits

IC test is a requirement for viable products. Without testing, the manufacturer could deliver defective chips for customers, causing economical and trust losses. From the test perspective, the basic difference between asynchronous and synchronous is the self-timed computation versus lock-step computation. Manufacturing for asynchronous or synchronous circuits is the same, which means that defects are similar in both designs [17]. The lack of universal testing methodologies, for asynchronous templates hinders the use of asynchronous designs in commercial products. Currently, test approaches for QDI and BD families are fundamentally different from each other. Within each family there are different templates such as Micropipelines, Mousetrap, Click and GasP for BD, each with a different test strategy. This part of the tutorial surveys these test methods and present current techniques used for testing the BLADE template [15].

References

[1] P. Beerel, R. Ozdag M. Ferretti. “A Designer’s Guide to Asynchronous VLSI”. Cambridge University Press, 2010, 337 p.

[2] ITRS, “Design Section”, 2011, available at http://www.itrs.net.

[3] J. Pontes et al. “SCAFFI: An intrachip FPGA asynchronous interface based on hard macros”. In: ICCD, 2007, pp. 541-546.

[4] V. Ekanayak, R. Manohar. “Asynchronous DRAM Design and Synthesis”. In: ASYNC, 2003, pp. 174-183.

[5] R. Ginosar. “Fourteen Ways to Fool Your Synchronizer”. In: ASYNC, 2003, pp. 89-96.

[6] I. J. Chang et al. “Exploring asynchronous design techniques for process-tolerant and energy-efficient subthreshold operation”. IEEE JSSC, 45(2), pp. 401--410, Feb 2010.

[7] V. Lomné et al. “Evaluation on FPGA of Triple Rail Logic Robustness against DPA and DEMA”. In: DATE, 2009, pp. 634-639.

[8] M. Moreira, et al. “A 65nm Standard Cell Set and Flow Dedicated to Automated Asynchronous Circuits Design”. In: SOCC, 2011.

[9] M. Moreira et al. “Adapting a C-Element Design Flow for Low Power”. In: ICECS, 2011, pp. 45-48.

[10] M. Moreira et al. “Spatially Distributed Dual-Spacer Null Convention Logic Design”. Journal of Low Power Electronics, 10(3), September, 2014. pp. 313-320.

[11] R. Guazzelli et al. “Analysis of Supply Voltage Scaling on SDDS-NCL Designs”. In: SBCCI, 2015. pp. 2-8.

[12] M. Moreira, N. Calazans. “Design of Standard-Cell Libraries for Asynchronous Circuits with the ASCEnD Flow”. In: ISVLSI, 2013. PhD Forum Best Paper Award.

[13] M. Moreira et al. “LiChEn: Automated Electrical Characterization of Asynchronous Standard Cell Libraries”. In: DSD, 2013.

[14] D. Hand et al. Blade - A Timing Violation Resilient Asynchronous Template. In: ASYNC. 2015, pp. 21-28.

[15] F. Kuentzer, A. Amory. “Fault Classification of the Error Detection Logic in the Blade Resilient Template”. In: ASYNC, 2016, pp. 37-42.

[16] P. Beerel, N. Calazans. “A Path Towards Average-Case Silicon via Asynchronous Resilient Bundled-data Design”. In: ECCTD, 2015.

[17] M. Roncken, S. M. Gilla, H. Park, N. Jamadagni, C. Cowan and I. Sutherland, "Naturalized Communication and Testing". In: ASYNC, 2015, pp. 77-84.