

# ASCeND: A Standard Cell Library for Semi-Custom Asynchronous Design

Matheus Trevisan Moreira, Carlos Oliveira, Ney Laert Vilar Calazans

*GAPH – FACIN – Pontifical University of Rio Grande do Sul*

*Av. Ipiranga, 6681, B. 32 R. 726 - Partenon - Porto Alegre/RS – Brazil*

matheus.moreira@acad.pucrs.br

ney.calazans@pucrs.br

**Abstract**— The asynchronous circuit design paradigm provides a practical solution for several challenges and constraints of current and future technologies to build integrated circuits and systems. However, there is little electronic design automation support for this paradigm. This work presents a standard cell library designed to support a semi-custom approach in the design of asynchronous integrated circuits in 65nm. The library counts with a parameterizable design flow that allows it to be ported to different CMOS technologies with some degree of automation. Over five hundred components compose the current version of the library. Several case studies have been synthesized using the library components, all validated at the layout level.

**Keywords**— Asynchronous circuits, semi-custom design, standard-cell library, null-convention logic, C-element

## I. INTRODUCTION

At present, most system-on-chip (SoC) designs adopt the synchronous paradigm, which implies an assumption: a discrete notion of time. This abstraction significantly reduces the complexity of a digital circuit design and allows these to be more easily modeled using register transfer level (RTL) languages like VHDL or Verilog. However, with the evolution of CMOS technologies, limitations of this paradigm started to emerge. Synchronous design problems that were easily overcome in the past, such as clock skew and power consumption control, have become increasingly complex tasks to solve in modern designs [1] [2].

An alternative is to use asynchronous circuits, which do not employ any clock signal. In this way, time becomes a continuous variable. Contrarily to the synchronous paradigm, synchronization, sequencing and communication operations in asynchronous circuits require explicit handshaking protocols [3] between components. Consequently, registers are activated only when and where it is needed. Due to their nature, asynchronous circuits may present lower power consumption, higher operating speed, better composability and modularity, together with low electromagnetic emissions and higher robustness.

Perhaps asynchronous design main drawback is that it counts with little electronic design automation (EDA) support. Moreover, typical commercially available standard cell libraries do not provide components required to efficiently implement most circuit templates already devised for asynchronous circuits. In this way, asynchronous design is limited, in practice, to full custom approaches. This work presents a standard cell library containing components

required to implement asynchronous integrated circuits (ICs). The current version of the library counts 508 standard cells and is implemented in STMicroelectronics 65nm CMOS technology. It is open access to designers with access to the technology design kits, available for example through multi-project services as CMP. Moreover, the library is fully integrated with two higher level synthesis tools for asynchronous circuits, making a semi-custom approach viable to implement asynchronous circuits.

## II. ASYNCHRONOUS STANDARD CELLS

Asynchronous circuits can be classified according to several criteria. One important criterion is based on the delays of wires and gates. The most robust and restrictive delay model is the Delay-Insensitive (DI) model, which operates correctly regardless of gate and wire delay values. Unfortunately, this class is too restrictive. The addition of an assumption on wire delays in some carefully selected forks enables to define the Quasi-Delay-Insensitive (QDI) circuit class. Here, signal transitions occur at the same time only at each end point of the mentioned forks, which are called isochronic forks. According to Martin and Nyström, the QDI class comprises almost the entirety of options in practical asynchronous design [4].

Currently, in order to implement Boolean functions without losing the delay insensitivity property, two logic styles are usually employed for QDI circuits: Delay Insensitive Minterm Synthesis (DIMS) [3] and Null Convention Logic (NCL) [5]. The former relies on the extensive use of an asynchronous component called C-element [3], while the latter employs special components called NCL gates. Also, Mutual Exclusion Elements (MUTEXs) are crucial components to implement asynchronous circuits. This is due to the fact that they guarantee a robust implementation of control elements, like arbiters. The drawback is that none of the mentioned components is usually present in typical standard cell libraries, as they are not frequently employed in synchronous design and are usually not recognized by typical EDA tools.

### A. C-Element

The Muller C-element, or simply C-element is a sequential logic device that operates as an event synchronizer. Fig. 1 shows a truth table for one typical C-element and the associated logic symbol for a 2-input component. When inputs

A and B are the same, output Q assumes this value. When the inputs are different, the output keeps its previous value.

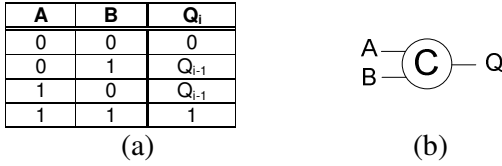


Fig. 1 Two input C-element truth table (a) and symbol (b).

Versions with distinct behaviour exist, where one or some of the inputs of the C-element may interfere only in the high-to-low or the low-to-high transition of the output. The importance of such variations in the synthesis of asynchronous circuits is discussed e.g. in [6]. Alternative C-element behaviours characterize the asymmetric, unbalanced or generalized C-elements. Also, C-Elements may be implemented using different transistor topologies, as discussed in [7], and with set/reset control signals.

### B. NCL gates

NCL was proposed by Theseus Logic, Inc. [5] and has been employed for implementing QDI asynchronous systems on silicon. It is an alternative to other design styles like delay insensitive minterm synthesis (DIMS) and one of its advantages is that it enables power-, area- and speed-efficient QDI design with a standard-cell-based approach. NCL gates are sometimes called threshold gates, but this is imprecise. In fact, NCL gates couple a threshold function [8] with positive integer weights assigned to inputs to the use of a hysteresis mechanism. Fig. 2 shows the NCL gate symbol, where N is the number of inputs and M is the gate threshold.

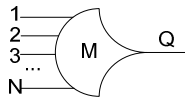


Fig. 2 Symbol of an M-of-N NCL gate.

### C. Mutual Exclusion Element

MUTEXs are essential to compute data and take decisions for the correct operation of the circuit [3]. The role of this component is to decide which of two concurrent events is to be served first. In this work, it is assumed that metastability filters have two request inputs (RA and RB), that receive requests, which can be concurrent, and two acknowledge outputs (AA and AB), that signal which event is to be served. The logical function of this component is showed in (1). In the case of two events taking place in the exact same instant of time, e.g. simultaneous low-to-high transitions in RA and RB, the filter will eventually decide for one, eliminating the possibility of metastable states. This scenario, which is not covered by the logical function, is decided through an electric race, where the request with best potential is served first. Because AA and AB are mutually exclusive, the request with

lower potential will only be served after the first finishes its communication.

$$AA = \overline{RB} \wedge RA ; AB = \overline{RA} \wedge RB \quad (1)$$

### III. THE ASCEND LIBRARY

Currently, ASCeND is available for the 65nm STMicroelectronics CMOS technology (ASCeND-ST65). The library counts with C-Elements, NCL gates and MUTEXs. A total of 504 C-elements compose the library as Fig. 3 shows. It counts with C-elements with varying driving strengths (speed to charge/discharge a load), functionalities and topologies. Moreover, these components are available in high speed and low power versions, each with its own advantages. ASCeND-ST65 C-elements tradeoffs are discussed in [7] and [9].

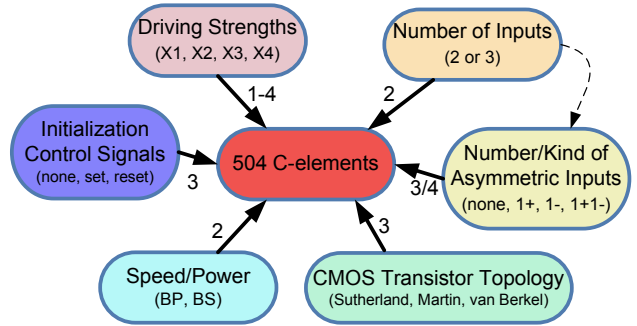


Fig. 3 ASCeND-ST65 C-elements. Rounded corner rectangles represent one parameter choice. Edges are labeled with the number of choices the parameter implies. The dotted edge represents dependencies between parameters choices.

A set of 14 different NCL functionalities is also available in ASCeND-ST65. Each of these functionalities is designed as a standard cell and has 4 different driving strengths. This results in a set of 14\*4=56 NCL gates. The symbols of these gates is represented in Fig. 4.

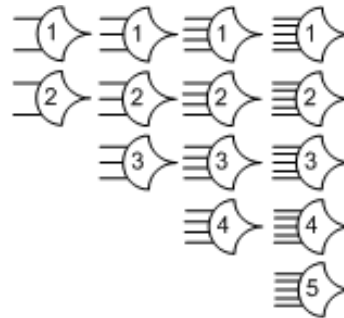


Fig. 4 ASCeND-ST65 NCL gates.

ASCeND-ST65 also counts with 4 versions of metastability filters. Each implementation employs different transistor sizes and, consequently, presents distinct driving strengths. Together with a typical standard cell library, containing combinational logic gates, the library can be used to implement asynchronous circuits of different classes through a

semi-custom approach, as showed in [10], which is much less restrictive than template based approaches.

In ASCEnD-ST65 all the components, but MUTEXs (which is typically manually designed), are designed through the flow depicted in Fig. 5. The flow starts with an electrical and a functional specification, a transistors schematic is designed and used in the ROGen tool, an in-house tool that generates a simulation circuit described in SPICE, described in detail in [10]. This circuit is simulated using Cadence Spectre and the resultant simulation report is the input of another in-house tool called CeS. This tool is responsible of dimensioning the transistors of the standard cell based on the provided report and is explained in detail in [10].

The dimensioned schematic is then used to manually design a layout using Cadence Virtuoso Layout Editor. In this step, all components are designed at employing design for manufacturability (DFM) techniques. DRC and LVS checking are performed using Mentor Calibre.

After correct layout design is verified, parasitics are extracted, also using Calibre, and electrically characterized using another in-house tool called LiChEn [11]. This tool generates power and timing models according to the Synopsys Liberty Format [12], which is compatible with most EDA vendors. In this step, it is verified if the generated circuit respects the specification. If not, it must be redesigned. Once it is verified to behave as specified, an abstract view is generated and exported to Cadence Library Exchange Format, which is compatible with most EDA vendors, and a symbol view is generated to allow structural hierarchic design. Finally, a Verilog behavioural view is generated to allow digital simulation.

Initially, the ASCEnD flow was proposed for C-Elements design only. However, it has recently been employed for NCL gates design. The integration of these gates with the ASCEnD flow is detailed in [13].

The resultant library was extensively validated through simulation after place and route of some complex ICs, such as

a RSA cryptographic core and three different network-on-chip routers. For more details on these designs, address [7], [9], [10], [14] and [15]. Also, the library can be easily ported to other CMOS technologies, given that the adopted design flow counts with automated and parameterizable tools. For detailed information about the flow, address references [6] and [7].

#### IV. CONCLUSIONS

ASCEnD is a freely available standard cell library for supporting asynchronous ICs semi-custom designs. The library has its components designed at the layout level and can be easily ported to different CMOS technologies. Currently, a new version of the library is being generated, for the IBM 130nm CMOS technology, available through MOSIS services. This will ease the prototyping of study case circuits, easing the validation of the library on silicon.

Future work includes the development of the tools of the flow in order to support gates required by other asynchronous design templates, such as pre-charged half-buffer and pre-charged full-buffer. Also, it is also part of future work to enable the ASCEnD flow to design standard-cells for operating at subthreshold voltages, in order to allow the design of low-power subthreshold asynchronous circuits.

Access to ASCEnD-ST65 can be obtained by contacting the authors of this paper. This access is free, provided that the requester has signed the required non-disclosure agreements with STMicroelectronics for the 65nm CMOS technology. Finally, having a freely available standard-cell for asynchronous design is very important for a wider adoption of asynchronous techniques, which are each day more necessary.

#### ACKNOWLEDGMENT

This work is partially supported by the CAPES-PROSUP and FAPERGS (under grants 11/0455-5 and 11/1445-0). Ney Calazans acknowledges CNPq support under grant 310864/2011-9.

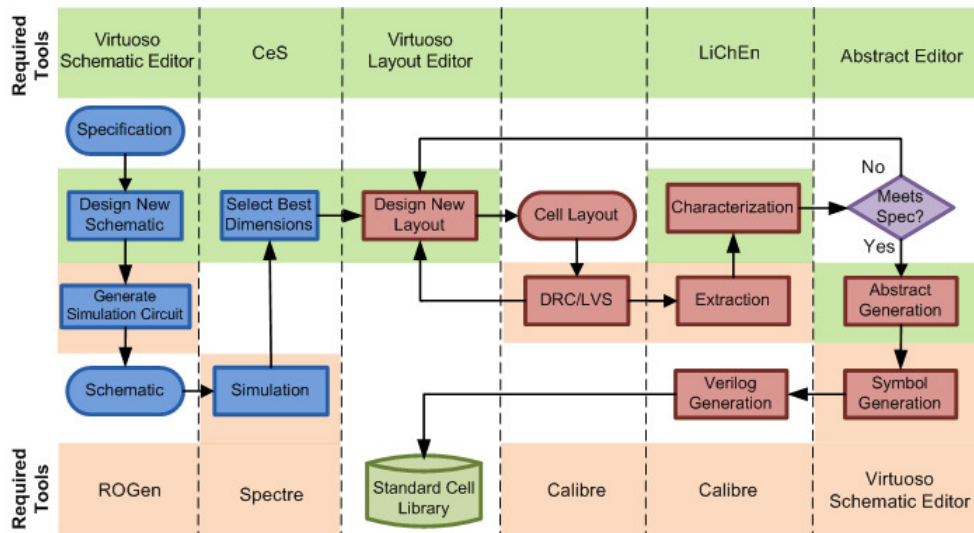


Fig. 5 Design flow adopted for ASCEnD standard cells. Actions are represented by boxes, decisions by diamonds, descriptions as rounded boxes and the repository as a cylinder.

## REFERENCES

- [1] M. Made, T. Felicijan, A. Efthymiou, D. Edwards, L. Lavagno "Asynchronous on-chip Networks". *Computers and Digital Techniques*, IEEE Proceedings, 152(2), March 2005, pp. 273-283.
- [2] N. Ekekwe, "Power dissipation and interconnect noise challenges in nanometer CMOS technologies," *IEEE Potentials*, 29(3), pp.26-31, May 2010.
- [3] J. Sparsø and S. Furber, "Principles of Asynchronous Circuit Design – A Systems Perspective," Kluwer Ac. Pub., Boston, 354p., 2001.
- [4] A. J. Martin and M. Nyström, "Asynchronous Techniques for System-on-Chip Design," *Proceedings of the IEEE*, 94(6), June 2006, pp. 1089-1020.
- [5] K. M. Fant and S. A. Brandt, "NULL convention logic: a complete and consistent logic for asynchronous digital circuit synthesis," In *International Conference on Application Specific Systems, Architectures and Processors (ASAP'96)*, 1996, pp. 261-273.
- [6] W. B. Toms, "Synthesis of Quasi-Delay-Insensitive Datapath Circuits," PhD thesis, Univ. of Manchester, 237 p., Feb. 2006.
- [7] M. T. Moreira, B. S. Oliveira, F. G. Moraes, and N. L. V. Calazans, "Impact of C-elements in asynchronous circuits," In: *ISQED '12*, 2012, pp. 438-444.
- [8] S. L. Hurst., "An Introduction to Threshold Logic: A Survey of Present Theory and Practice," *The Radio and Electronic Engineer*, 37(6), June 1969, pp. 339-351.
- [9] M. T. Moreira, B. S. Oliveira, J. J. H. Pontes, F. G. Moraes, N. L. V. Calazans "Adapting a C-Element Design Flow for Low Power" In: *IEEE International Conference on Electronics, Circuits and Systems*, Beirut, 2011. Pp. 45-48.
- [10] M. T. Moreira, B. S. Oliveira, J. J. H. Pontes, N. L. V. Calazans "A 65nm Standard Cell Set and Flow Dedicated to Automated Asynchronous Circuits Design" In: *IEEE International SoC Conference*, 2011. Pp. 99-104.
- [11] M. T. Moreira and N. L. V. Calazans, "Electrical Characterization of a C-Element with LiChEn," In: *IEEE International Conference on Electronics, Circuits and Systems*, Seville, 2012. Pp. 583-585.
- [12] Synopsys Liberty. "Open Source Liberty". Available at <http://www.OpensourceLiberty.org>, 2013.
- [13] M. T. Moreira, C. H. M. Oliveira, R. C. Porto and N. L. V. Calazans, "Design of NCL Gates with the ASCeND Flow," In: *Latin American Symposium on Circuits and Systems*, Cuzco, 2013.
- [14] J. J. H. Pontes, M. T. Moreira, F. G. Moraes and N. L. V. Calazans, "Hermes-A – An Asynchronous NoC Router with Distributed Routing," In: *International Workshop on Power and Timing Modeling, Optimization and Simulation*, LNCS volume 6448, Grenoble, 2010. Pp. 150-159.
- [15] J. J. H. Pontes, M. T. Moreira, F. G. Moraes, N. L. V. Calazans "Hermes-AA – A 65nm Asynchronous NoC Router with Adaptive Routing" In: *IEEE International SoC Conference*, 2010. Pp. 493-498.