From an extended study of asynchronous controllers to system level design: application to the design of digital and analog interfaces

E. Beigne*, P. Vivet* and M. Renaudin**

* CEA LETI, Minatec Campus, 38054 Grenoble, France
** Tiempo, 38330 Montbonnnot, France Grenoble, France
edith.beigne@cea.fr, pascal.vivet@cea.fr,
marc.renaudin@tiempo-ic.com

Abstract. Asynchronous circuits have been proposed for many years and have successfully shown their robustness to delay variations making them a key solution for digital and mixed-signal circuits. More specifically, asynchronous circuits have demonstrated their efficiency in solving control, arbiter and interfaces issues. In order to make them more widely adopted, modeling and synthesis of asynchronous circuits have been proposed through different methodologies that we will firstly detail and discuss in this paper. We will then demonstrate the use of these specific methodologies for the synthesis and design of digital and analog-to-digital interfaces. The overall paper aims at providing a study and analysis of specific interfaces and their dedicated controllers for synchronization and energy management applications while discussing the asynchronous community positioning accordingly.

Keywords: Asynchronous circuits, controllers, interfaces, digital, analog, modelling, synthesis, verification, STGs, Petri Nets.

1 INTRODUCTION: TOOLS TO SOLVE INTERFACES ISSUES

This is now a cliché to say that there is a lack of tools for the design and verification of asynchronous circuits. But is that true? Not really, if we consider how challenging it is to enumerate all the tools that were developed and/or are still in use: Sis, Assassin, Forcage, Meat, Verdect, Syn, Versify, Tangram, Balsa, Teak, Cast, NCL, Ack, Tast, Pipefitter, Chp2Vhdl, Verilog2Stg, Desynch, Class, Haste, Tiempo-Tools, Petrify, Minimalist, Proteus, 3D…

So, why are there so many of them, and why people are complaining about a lack of tool? This is probably because there is still no consensus, neither about a modeling-language nor about a circuit-style. Therefore, many modeling languages and many asynchronous circuit styles are in use, not giving the opportunity of visible ones to emerge.

Prof. Yakovlev vision is focused, the language has to be suited to model, verify and synthesize asynchronous controllers, i.e. control circuits that are reactive to their input signals and able to drive their output signals, following a formal specification of concurrency, causality and choice. With this respect, a high level language is not necessary, there's no need for communicating processes, nor channels, but instead there is the need for modeling signals' behavior formally using concurrency and choice at a very fine grain. Exit Tangram, Balsa, CHP, HDLs, etc. and welcome Petri Nets [1] [2], STGs [3], or FSMs, even if there might be some links between them [4]. The spectrum of formalisms being narrowed, what is the appropriate graph based methodology and the supporting modeling language? PN, STGs, FSMs all target speed independent circuits which are robust enough to accommodate the variations the controllers have to resist, but they are not equivalent with respect to their ability to efficiently model and synthesize controllers.

Prof. Yakovlev and his team contributed to a major evolution of these graph-based approaches with the introduction of CPOG [5], because it prevents the designer from explicitly enumerating all the signal events and their traces together with their causal relations. CPOG is based on a description of the scenarios, i.e. the events traces, using a compact functional form. The signal encodings are specified apart, thus simplifying the controller's behavioral specification, and at the same time enabling synthesizing the controller with different encodings but keeping the same structural specification. Both aspects brought significant improvements to the modeling and design of asynchronous controllers.

So, now it's fine, let's develop design tools using all the concurrency theory, the computer science and electrical engineering knowledge and skills underneath, in order to apply and prove that all this research is solving relevant real and physical problems. With this respect Prof. Yakovlev vision is very wide, especially in the domain of interfaces we want to focus on in this paper. He proposed the specification, implementation and verification of many arbiters, synchronizers and controllers for the communication between digital circuits both synchronous and asynchronous, or between digital and analog circuits, and also controllers dedicated to phase encodings and analog circuits, etc…

Prof. Yakovlev scientific contribution is significant and well recognized; it was used and continues to inspire many works in different domains, such as the design of interfaces we choose to briefly address in this paper, because it illustrates very well the broad scientific knowledges involved.

2 DIGITAL INTERFACES FOR SYNCHRONIZATION

In advanced digital systems with nowadays large System-on-Chips and multicore architectures, one of the primary concern is the interconnect infrastructure, and how to implement efficient system communication for on-chip or off-chip communication. With tens of cores, hundreds of different clocks, and due to the always larger delay in wires compared to delay in gates, communication architecture and its clean implementation is a great challenge. Deep pipelining must be implemented to transport

information from one die side to its other side. It is more costly to exchange data than performing computation.

In this context, Prof. Yakovlev was a precursor and very early applied his knowledge of asynchronous controller design to the design of generic and abstract interfaces to implement system interfaces. In [6], he early proposed the model and design of controllers to implement system communication of arbitrary protocols. It was applied to a simple fifo interface but claimed to be generic for application to any kind of interfaces: “a unified solution to the problem of synthesizing logic implementations from abstract specifications. Although easily applied to control circuits, it is suitable for all types of interface hardware”. It is worth to notice how abstract models can help to derive logic interfaces implemented in asynchronous logic. Prof. Yakovlev studied many kind of interfaces, such as [7] proposing system communication using STG, allowing reader & writer to indefinitely wait for an answer, compared to earlier work using fundamental mode : such interfaces was then latency insensitive (at protocol and system level) and Speed Independent or QDI at signal level. In [8], the proposed asynchronous interfaces could cope with various timing and protocol interfaces.

One of the primary aspects of system communication is obviously synchronization: a clean synchronization of low level signals must be performed to ensure robust communication. On this domain, Alex Yakovlev has made on the long time a very wide study of synchronization and its formalization, both at electrical level with the study of synchronizer cell elements, and at logical level with the design of a full range of asynchronous arbiters, using various specification methods and proposing different services. He proposed for instance flat arbiters [9], priority arbiters [10], and parallel multi-resource arbiters [11], among many other ones. More recently, he also proposed opportunistic merge elements, as a way to efficiently control asynchronous events within a switch cap converter. One can notice that smart digital interface implementing synchronization may also be required for efficient control of the analog domain, as it will be presented in section III.

These contributions on interface synchronization and arbiter design have been a regular source of inspiration for the design of large scale system level interconnects. In the 2000’s, the Network-on-Chip paradigm has been introduced and many flavors where proposed, in conjunction with the Globally Asynchronous Locally Synchronous (GALS) paradigm as early introduced by D. Chapiro. The NoC architecture offers first of all a clear separation of computation and communication, compared to previous bus-based topologies, and NoC perfectly fits the GALS paradigm, where computations (core, accelerators, etc…) are implemented using synchronous methods while system communication are implemented using asynchronous methods. Many flavors of GALS NoCs are then possible: either multi-synchronous, or mesochronous, or fully asynchronous, using either FIFO like interfaces or pausable clocks interfaces. Many asynchronous Network-on-Chip have been introduced and implemented, such as Chain, QNoC, Mango, ANOC, Hermes-A and more recently a 2-phase bundle-data NoC.

For the CEA-LETI ANOC architecture, the proposal is based on wormhole packet switching, it includes virtual channels for Quality of Service, while routers and links

are implemented using Quasi Delay Insensitive logic with 4-phase / 4-rail protocol for robust system communications. The early arbiters design within the ANOC router were initiated by early works from Marc Renaudin and by priority arbiters [9] from Prof. Yakovlev proposal. Within CEA-LETI, a full series of circuit demonstrators have been developed using ANOC protocol, which was continuously improved by adding new features, such as on-chip and off-chip NoC interfaces, Design-for-Test wrapper, automatic router power down using activity detection, integration of DVFS scheme within NoC units, and with an automated flow for timing analysis and physical implementation. An extensive comparative study showed a clear benefit of ANOC versus its synchronous counterparts with a gain in power consumption by a ratio of 5.

For NoC topologies, a challenge is the design of efficient and robust NoC links using dense encodings. Alex Yakovlev proposed original contributions on protocol signaling for system communication. For instance in [12], he proposed a novel self-timed communication protocol based upon phase-modulation of a reference signal. The reference and the data are sent on the same transmission lines and the data can be recovered observing the sequence of events on the same lines phase, offering a compact code, while being robust to faults. In the same period, for system communication, people tried also to extend and optimize the existing 2-phase protocol (for reduced number of transitions compared to 4-phase), but using denser codes. For examples, the LETS code offers a generic solution for 2-phase multi-rail codes, their protocols and their associated protocol converters. More recently, a 1-of-T multi-rail code was proposed for 2phase signaling with efficient 2-phase/4-phase protocol converters.

These few examples of asynchronous protocols and corresponding protocol converters can be applied to 3D architecture. 3D technology using so-called Through-Silicon-Via (TSV) offer a new paradigm for further system integration, with strong benefits in terms of yield and system partitioning, power reduction due to shorter connections, and a full spectrum of new architecture by using heterogeneous technologies (logic, memory, NVM, MEMs, etc). Again, in such 3D architecture, scalable and modular system communication is a challenge, and it can be elegantly implemented as an asynchronous 3D Network-on-Chip, by offering power efficient and robust 3D asynchronous communications without any global 3D clocking.

To conclude this section, asynchronous logic offers an efficient way to implement digital interfaces at system level. GALS has still a long story to play. The main design challenges are still the same: synchronization, arbitration, protocol communication and signaling. Detailed modeling of the interfaces is mandatory to ensure reliability and robust design. More automation would help but due to the large spectrum of all these interfaces, it will be difficult [16]. Even if mature solutions exist, research can always be carried-on on these complex topics to continue innovation on protocol signaling and associated control schemes!

3 MIXED-SIGNAL AND ANALOG INTERFACES

Interfaces issues are more and more problematic today with the advent of power management circuit techniques and mixed-signal systems of the Internet-of-Things (IoT). In the first case, voltage and frequency generators are analog circuits, integrated on-chip, but controlling digital circuits. They require internal control loops and also need to be efficiently interfaced with digital blocks. IoT systems, on their side, are mixed-signal circuits in which many different interfaces are managed. In that kind of circuit, analog information is coming from sensors, wireless transfer or even energy harvesters. In that context, Prof. Yakovlev proposed many novel solutions from mixed-signal circuit workflow using asynchronous control [13] to Buck-boost DC-DC converters controls [14] and specific efficient IT controller called opportunistic merge element [15]. In the following we will discuss some of these works and put in perspective our current works using energy efficient asynchronous design.

3.1 The advent of mixed-signal circuits

To start with, it is obvious to say that, today, most of our circuits are mixed-signal. We use to call mixed-signal, a circuit which contains an analog and a digital part, interfaced with Analog-to-Digital or Digital-to-Analog converters. The reality is more complex: small digital circuits are necessary to control analog blocks. It can be implemented as control loops or finite-state-machines. Prof. Yakovlev proposed a framework for the design of mixed-signal systems with asynchronous control [13]. It enables formal verification of AMS systems with asynchronous control and the workflow is efficiently demonstrated on a buck-boost converter. In our lab, we implemented in 2007 an integrated micro battery charger which can be charged by a thermo-generator's DC/DC output or by an HF converter. The power supply manager consisted in a specific unit along with an asynchronous finite state machine to manage priority between the two different sources. Electrical simulations were performed but no framework was available at that time to formally verify our circuit's functionality. This could have been solved by Prof. Yakovlev work and the proposed AMS methodology [13] and would have improved a lot our verification step.

3.2 Power management circuits and limitations

One of the main mixed-signal components in today's circuits is related to power management and more precisely to voltage generation. Prof. Yakovlev demonstrated that power control can significantly benefit from the use of asynchronous logic [14]. They demonstrated how to design and verify a speed-independent multi-phase buck-boost converter. It has been shown to be extremely robust to the changes in power demand. In the same topic, in our lab, we first tried to get rid of classical synchronous DC-DC converter by proposing a Vdd-Hopping solution based on voltage dithering. In this scheme, two or three supply voltages are available on chip and delivered through power switches to the digital block. The main advantage is very high power efficiency as it only consumes during transitions. To avoid any undershoots and overshoots on

the digital block power supply, a local control loop is implemented. It was originally designed using synchronous logic, but we finally realized that an asynchronous loop would have probably been more robust to voltage changes as demonstrated by Prof. Yakovlev in a paper entitled “Design and Verification of Speed-Independent Multi-phase Buck Controller”. Another interesting application of asynchronous circuits at the interface of power supply and energy harvesters is presented in our lab in 2010. An asynchronous state-machine controls the energy levels between different energy sources (Photo-voltaic, Thermoelectric, etc.). The control is based on voltage level crossings on capacitances representing the energy level of the sources. We are still working on this topic today as, ideally; it could bring infinite energy autonomy to IoT circuits and systems making them fully energy-driven (similarly to asynchronous data-driven circuits).

3.3 Low power IoT circuits and systems

We have just discussed interfaces and control for mixed-signal circuits related to power management issues. In addition, in emerging IoT systems, many analog sensors have to interface with digital controllers. Events coming from these sensors are purely asynchronous and it would be inefficient in terms of energy to continuously sample them as they can occur in very different time scales. An efficient solution is to handle these events as asynchronous Interrupts (ITs) computed by an asynchronous IT controller (IT-Ctrl) as shown in our very recent works. An asynchronous IT-Ctrl manages all the peripherals events and sends the priority number to a decoder. Interruptions are sorted by interruption numbers with the highest priority on interrupt number 0 and weakest priority on interrupt number 31. This scheme is interesting in terms of wake-up as, due to its asynchronous implementation, it immediately reacts on an incoming interrupt. However, the priority could be treated in a different way to improve this controller’s energy efficiency. Indeed, Prof. Yakovlev’s work proposes an element named opportunistic merge element which could greatly improve our IT controller performances. This innovative asynchronous component merges two or more request-acknowledge channels into one and is allowed to opportunistically send requests if they arrive close to each other. Our future works aim at improving our current IT controller with the help of an opportunistic scheme as described [15] by Prof. Yakovlev work.

4 CONCLUSION

We have discussed in this paper interfaces issues in digital and mixed-signal circuits. These interfaces can be implemented between synchronous domains or between analog to digital and digital to analog parts of circuits. They are also seen as a key issue between synchronous and asynchronous timing domains. We have shown how Prof Yakovlev’s work has been focused, in part, to solve these issues by proposing many different methodologies and innovative implementation schemes. Our aim has been to put in perspective our own work with respect to his research all along the paper and to

show that most of our proposals are complementary or could have been even greatly improved by Prof Yakovlev's proposals.

References

1. A. Yakovlev, A. Koelmans, A. Semenov, D. Kinniment, "Modelling, analysis and synthesis of asynchronous control circuits using Petri Nets", Integration: the VLSI Journal, 1996, Vol. 21, N°. 3, pp. 143-170.
2. A. Yakovlev, A. Koelmans, "Petri Nets and Digital Hardware design", in W. Reisig, G. Rosenberg ed., Lectures on Petri Nets II: Applications, Advances in Petri Nets, Berlin, New York, Springer-Verlag, 1998, 154-236.
3. D. Sokolov, A. Bystrov, A. Yakovlev, "STG optimization in the direct mapping of asynchronous circuits", Design Automation and Test in Europe (DATE), 2003, Munich, Germany.
4. M. Renaudin, A. Yakovlev, "From Hardware Processes to Asynchronous Circuits via Petri Nets: an Application to Arbiter Design", Workshop on Token Based Computing (ToBaCo), part of the 25th International Conference on Applications and Theory of Petri Nets (ICATPN'04), 2004.
5. A. Mokhov, A. Yakovlev, "Conditional Partial Orders Graphs: Models, Synthesis, and Application", IEEE Transactions on Computers, 2010, Vol. 59, N°. 11, 1480-1493.
6. A. Yakovlev, A. Koelmans, L. Lavagno, "High-level modeling and design of asynchronous interface logic", IEEE Design & Test of Computers, 1995, vol. 12, issue 1, pp. 32-40.
7. F. Xia, A. Yakovlev, D. Shang, A. Bystrov, A. Koelmans, D. Kinniment, "Asynchronous communication mechanisms using self-timed circuits", Sixth International Symposium on Advanced Research in Asynchronous Circuits and Systems, 2000, pp. 150-159.
8. F. Xia, A. Yakovlev, I. G. Clark, D. Shang, "Data communication in systems with heterogeneous timing", IEEE Micro, 2002, Vol. 22, Issue 6, pp. 58-69.
9. A. Mokhov, V. Khomenko, A. Yakovlev, "Flat Arbiters", Fundamenta Informaticae, 2011, 108(1-2), pp. 63-90.
10. A. Bystrov, D. Kinniment, A. Yakovlev, "Priority Arbiters", International Symposium on Advanced Research in Asynchronous Circuits, April 2000, pp. 128-137.
11. D. Shang, F. Xia, A. Yakovlev, "Highly parallel multi-resource arbiters", Proceedings of 2010 IEEE International Symposium on Circuits and Systems, 2010, pp. 4117-4120.
12. C. D'Alessandro, Delong Shang, A. Bystrov, A. Yakovlev, O. Maevsky, "Multiple-rail phase-encoding for NoC", 12th IEEE International Symposium on Asynchronous Circuits and Systems, 2006, pp.110-116.
13. V. Dubikhin, C. J. Myers, A. Yakovlev, D. Sokolov, "Design of Mixed-signal Systems with Asynchronous Control", IEEE Design & Test, 2016, 10.1109/MDAT.2016.2555916.
14. D. Sokolov, V. Khomenko, A. Mokhov, A. Yakovlev and D. Lloyd, "Design and Verification of Speed-Independent Multiphase Buck Controller," Asynchronous Circuits and Systems (ASYNC), IEEE International Symposium on, Mountain View, CA, 2015, pp. 29-36.
15. A. Mokhov, V. Khomenko, D. Sokolov and A. Yakovlev, "Opportunistic Merge Element," Asynchronous Circuits and Systems (ASYNC), 2015 21st IEEE International Symposium on, Mountain View, CA, 2015, pp. 116-123.
16. A. Yakovlev, P. Vivet, M. Renaudin, "Advances in Asynchronous Logic: from Principles to GALS & NoC, Recent industry Applications, and Commercial CAD Tools", in Conference on Design, Automation and Test in Europe (DATE'13), 2013, Grenoble, France.