Clockless computing or learning how to play "soft time" in "hard space"

A.V. Yakovlev

School of Electrical, Electronic and Computer Engineering, University of Newcastle, Newcastle upon Tyne, NE1 7RU, U.K.
Alex.Yakovlev@ncl.ac.uk

Abstract

Research on aperiodic automata, self-timed systems and implementation of concurrent processes in hardware was pioneered at LETI by V.I. Varshavsky. This paper addresses some of the important scientific results achieved in the last decade in the area of building clockless systems and assess potential for their future advancement.

1 Introduction

Research on aperiodic automata, self-timed systems and implementation of concurrent processes in hardware was pioneered at LETI by V.I. Varshavsky. Throughout the late 70s, 80s and early 90s he led a research group at the Department of Computer Software that was founded by V.I. Timokhin, whose constant support helped this research to flourish and produce an enormous wealth of knowledge. This work provided clear understanding of the principles of asynchronous communication and computation and the ways of their implementation in digital VLSI hardware [1, 2]. The key idea behind these principles was based on the concurrency and "soft timing" inherent in self-timed systems due to the use of local handshaking and signal acknowledgement. The latter helped computing engines to be liberated from the tirany of global clocking and rigid time-intervalling. The practical advantages of self-timing are in saving power and reducing heat dissipation, achieving more robust synchronisation and allowing design reuse for complex heterogeneous systems, and offering the possibility to build systems with self-checking and self-recovery. The results of V.I. Varshavsky's group's research laid a solid scientific foundation to a wide range of developments in microelectronics and computer engineering that took place in the 90's in the international arena. Industrial design examples are rapidly mushrooming around the world in signal processing, mobile computing and embedded systems in general. They exploit those principles and circuit solutions, in the form of fully working asynchronous microprocessors and microcontrollers [3]. As semiconductor technology marches through the new era of Systems and Networks on Chips [4] and faces the thrilling uncertainty of nanotechnology and quantum computing, the role and the future of systems with "soft" timing only seems brighter.

This short paper does not attempt to cover the complete state-of-the-art in the area of building clockless systems. It only addresses some of the scientific results achieved in the last ten years in this area and assesses potential for their future advancement. It particularly focuses on the research which involved the author himself. Having thoroughly enjoyed this involvement, the author feels extremely fortunate to have had such tremendous teachers as V. Varshavsky, L. Rosenblum, V. Marakhovsky and N. Starodoubtsev, and such wonderful research colleagues as V. Krasyuk, A. Petrov, Yu. Tatarinov, O. Maevsky, Yu. Mamrukov, I. Yatsenko, and many others, at the start of his research career at LETI. He feels honoured having been associated with these remarkable scientists and other brilliant colleagues from the UK and other parts of the world, including J. Cortadella, A.C. Davies, D.J. Kinniment, M. Kishinevsky, A.M. Koelmans, A. Kondratyev, M. Koutny, L. Lavagno, I. Mitrani, E. Pastor, A. Romanovsky, G. Russell, A. Taubin, W. Vogler and others (this list is by no means complete!), with whom he worked during the last decade. Last but not least the author is thankful to the young research associates and graduate students at Newcastle, F. Burns, A. Bystrov, I. Clark, V. Khomenko, L. Lloyd, A. Madalinski, M. Pietkiewicz-Koutny, A. Semenov, D. Shang, F. Xia, without whose hard work many of the results would not have been attained.

Perhaps, the most notable of all these investigations has been the work on developing formal models of concurrent and asynchronous systems behaviour and a Petri net based methodology for designing

asynchronous control circuits. This research, lying on the border between concurrent systems theory and digital circuit design, is sufficiently mature today and some of the achievements outlined in the following sections have reached the level of monographs, journal papers and tutorials.

2 Formal Models of Asynchronous Behaviour

Signal Transition Graphs. The key role here belongs to the model of *Signal Graphs* [15], better known today as *Signal Transition Graphs* or STGs (independently, a similar model was proposed at MIT by T.A. Chu in 1985). This model is based on Petri nets whose transitions are interpretted as the rising and falling edges of binary signals. Originally studied in the 80s, it was investigated in more detail later in [72, 71, 11, 12]. This model has found wide-spread use and led to further investigations by many researchers and asynchronous designers around the world in the last decade (monographs and papers in IEEE journals, and conferences such as ASYNC, ICCD, ICCAD, DAC, EDAC, EDTC).

Models with Relative Timing. The original STG model [15] was defined for *both untimed and timed cases*, thus not only enabling the design of circuits with unbounded delays (pessimistic, speed-independent, case) but also circuits with timing constraints (allowing optimisation for speed and area at the cost of being more definitive about delays). The latter aspect has also led to a variety of investigations in the asynchronous community, including the concepts of "Lazy" Transition Systems and Relative Timing [19] that is now being actively used at Intel Corp.

Causal Logic Nets. The *concept of causality* is a fundamental one in modelling asynchronous hardware behaviour. Particularly innovative has been the investigation of OR-causality, a concurrency paradigm implemented within the new Petri net extension called Causal Logic Nets[12]. The original "binary version" of the STG model has also been extended to multi-valued or symbolic STGs [68]. This work has resulted in wider research in this area in the last few years (e.g., at University of Kaiserslautern).

3 Asynchronous Circuit Synthesis

The **Petri net based design methodology** plays a key role in the synthesis of asynchronous control circuits [6, 14, 16] The methodology has two stages. The first stage, Abstract Synthesis, uses labelled Petri nets and their composition. The second stage, Logic Synthesis, uses the refinement of the nets obtained by Abstract Synthesis into Signal Transition Graphs (STGs) and synthesis of hazard-free logic circuits from STGs. The last ten years have seen the development of a set of new methods and algorithms supporting this methodology, namely:

- methods for synthesis of speed-independent circuits *directly* from STGs, *avoiding* the full state space exploration, using *lock (coupledness) classes* [74] (originally proposed in the author's PhD thesis in 1982) and *using Petri net unfoldings and approximate boolean covers* [59] (developed further in the PhD thesis of A.Semenov, who graduated from LETI in 1993);
- a method for synthesis of *safe Petri nets with read arcs* from transition systems and a method of solving the *state encoding problem* [16] in STG-based synthesis, both based the *theory of regions in transition systems* [8, 9, 10];
- a method for the *hazard-free implementation* of speed-independent circuits, using *monotonic cover conditions* [28];
- a method for *decomposition and technology mapping* of speed-independent circuits, using *boolean factorisation and binary relations* [7, 24];
- methods for *asynchronous circuit optimisation* (for speed and area factors), and constructing *locally speed-independent (or with bounded delays) and globally delay-insensitive* circuits, using various *STG transformation* techniques (under appropriate equivalence criteria) *concurrency reduction and expansion, handshake expansion* [49, 50]
- a method for *circuit decomposition for implementation in negative (standard logic library) gates* [45].

Many algorithms supporting this methodology have been implemented in software tools, particularly in **Petrify** [16, 46, 32], developed by J. Cortadella at the Polytechnic University of Catalunya, and in PUNT, developed by A. Semenov at Newcastle. The recent monograph [16] presents the main synthesis flow from STGs. Furthermore, a large community of asynchronous system designers in the UK and abroad (e.g., the designers of the first industrial-strength asynchronous microprocessor Amulet at the University of Manchester, designers at Intel, Philips, Theseus Logic, AT&T to name but a few) are using STGs and Petrify for synthesis and analysis of their circuits.

4 Asynchronous Circuit Verification

The main developments in the area of asynchronous circuit verification were based on the exploitation of *partial order techniques*, such as *Petri net unfoldings* [30, 57, 60, 63, 64, 67]. New methods and algorithms for Petri net unfolding have been developed to improve the efficiency of the partial order analysis approach for k-bounded Petri nets and Petri nets with read arcs, using the techniques based on *a (FIFO or LIFO) ordering of tokens* in nonsafe places, *representative sets* of transitions [67], *weak causality and contextual cycles* [53]. These methods have been implemented in the above-mentioned PUNT tool. These techniques and tools have been successfuly used in a number of applications such as verifying control logic for Amulet microprocessors and checking coherence of H.Simpson's four-slot asynchronous communication mechanism [48]. The most recent work has applied *unfoldings and integer programming to complete state coding* verification [39].

5 Asynchronous Circuit Design

Design Case Studies. In order to prove the usefulness of formal methods, techniques and tools in practice, a number of *asynchronous control circuits* have been designed using Petri nets, STGs and related techniques. These design case studies include: interface logic and bus controllers [74, 14, 16], asynchronous pipeline token-ring interface [6, 66], arbiters [69, 47, 65], A/D converters [21, 56], micropipeline circuits and processors (e.g., Sproull's Counterflow pipeline) [6, 9, 13, 30, 25], ESPRIT ACID-WG industrial design problems, e.g. loadable mod-N up/down counter and interrupt controller [61].

A demonstrator chip, called HADIC, was designed at Newcastle in 1999-2000 [5] and fabricated by EUROPRACTICE. The chip included samples of arbitration and asynchronous communication circuits. The HADIC chip was successfully tested and used in recent experiments [18, 5].

Other design supporting methods. Other methods supporting asynchronous circuit design include: the principle of *structural fault-masking* in asynchronous interfaces and methods for self-recovery in asynchronous systems [73, 70]; a method for *estimating power consumption* in asynchronous control circuits *based on Petri net T-invariants* [54]; a method for *performance analysis of asynchronous arbiters* [52]; a method for *using VHDL in the asynchronous circuit synthesis* [62]; a method for *self-checking and self-recovery* in self-timed systems [34], a method for estimating the worst-case execution time for CPU models using coloured nets [22], analysis of metastable and dynamic behaviour of synchronisers and multiway-arbiters [38, 18].

6 Recent and current work

The most recent work on synthesis has addressed a number of issues concerned with the design productivity and has progressed in the following directions.

HDL frontend and direct mapping of Petri nets. To gain greater practicality in the automated synthesis of asynchronous circuits, and achieve their wider adoption, the concept of asynchronous design flow, based on the Hardware Description Language (HDL) front-end and use of Petri nets and STGs as intermediate (internal for the design tools) languages, is being developed in the BESST project [5]. This approach will have the advantage of being oriented on a non-expert (standard) designer [33]. The idea of *direct mapping of Petri nets* to asynchronous control circuits is a way to achieve productivity and optimality of asynchronous designs. It was first presented in [6] and most recently in [41, 37].

Asynchronous behaviour visualisation and interactive synthesis. To assist wider use of asynchronous design, techniques and algorithms for the visualisation of asynchronous and concurrent behaviour are being developed in the MOVIE project [5]. Although it has its independent value from the theoretical point of view (new forms of representing concurrency semantics in its partial order form), this research comes close with the above-mentioned work on automated synthesis and verification of asynchronous circuits, especially where such tasks are interactive and involve human designer. For that, the idea of *separating concurrency and choice* has been developed and implemented in software [40]. In addition, the new techniques for checking state coding conflicts based on unfoldings have been combined with the visualisation methods [35].

Heterogeneous systems and asynchronous communications. To help solve design problems in the current decade, with Systems-on-Chip, which will have up to billion transistors on a single die and thousands of timing domains, the idea of *heterogenously timed networks (hets)* has been proposed. This work is now being investigated in the COHERENT project, which involves collaboration with Kingston University [5]. The key components of a het are asynchronous communication mechanisms (ACMs). A new taxonomy of ACMs has been proposed using the expertise gained in the COMFORT project [5]. It involves *automated synthesis of ACM protocols*, and their hardware and software implementation [48, 10]. The idea of *exploiting wait-free communication at the hardware level* offers a very promising advantage of creating harmony between the traditionally conflicting requirements of real-time and low-energy consumption. This may revolutionise the area of embedded systems design, with unlimited opportunities for miniaturisation and flexible operation [43].

7 Future work

We are still far from the widespread use of asynchronous design in microelectronic industry. Here, the situation is like in setting up a non-linear switching process. The real commercial world cannot take the revolutionary ideas about building systems without clock onboard instantly. The process requires some catalysis in order to overcome the natural inertia. Therefore the **short-term** research needs are as follows:

- Easy-to-use CAD tools for constructing self-timed circuits by non-expert designers trained in traditional synchronous design. These tools are first supposed to provide a seamless evolutionary way from synchronous designs, and should rely on the use of the existing commerical design flow, including placement and routing software.

- Methods for testing asynchronous circuits using existing automatic testing equipment. Testing asynchronous circuits, whose behaviour is inherently concurrent and whose controllability and observability with respect to primary inputs/outputs is limited, is a big problem. Self-testing, online testing and built-in testing approaches need to be investigated further. At the same time, adding self-test facilities must not impair the performance of high-speed logic in the normal operation mode.

- Interfaces between synchronous and asynchronous circuit domains, or the so-called globally asynchronous and locally synchronous (GALS) systems. Again, this would help a more gradual transition from synchronous design approach and would allow reuse of the massive amount of "synchronous" intellectual property within the new asynchronous System-on-Chip context.

- More demonstrator designs and products with asynchronous circuits, to prove their advantages in terms of power savings, EMC, modularity, design efficiency, and robustness. The best areas of demonstrating the advantages are likely to be systems with heterogeneous timing such as those from signal and image processing applications and systems involving high-bandwidth on-chip networking.

In the **longer term**, a more fundamental research on truly asynchronous and concurrent systems needs to be pursued. This research should effectively develop a mature understanding of the Token-Based Computing in various implementation technologies. It may include (but not limited by):

- Synthesis of concurrent specifications of asynchronous designs from partial order and sequential fragments.

- Verification of complex asynchronous behaviour, such as that produced by a mix of datapath and control flow with a range of causality paradigms.

- Methods for the direct mapping of concurrent specifications onto various implementation technologies, such as CMOS, nanotubes, quantum dots etc.

- Methods for analysis and synthesis of circuits with analogue components, whose behaviour is that of a complex dynamic non-linear system.

Acknowledgement. This research was mainly supported by the UK Engineering and Physical Research Council (see [5]). Other support came from EC (ACiD Working Group), Newcastle University, British Council and Royal Society.

References

[1] V.I. Varshavsky, Editor. Aperiodic Automata, Nauka, Moscow, 1976.

[2] V.I. Varshavsky, Editor. Self-Timed Control of Concurrent Processes: The Design of Aperiodic Logical Circuits in Computers and Discrete Systems. Kluwer Academic Publishers, Dordrecht, 1990.

[3] I.E. Sutherland and J. Ebergen, Computers without clocks. Scientific American, 287(2), August 2002.

[4] A. Allan, et. al., 2001 Technology Roadmap for Semiconductors, *Computer*, January 2002, pp. 42-53.

[5] async.org.uk - Newcastle Asynchronous Systems Research Group web site.

[6] A.V. Yakovlev and A.M. Koelmans. Petri nets and Digital Hardware Design Lectures on Petri Nets II: Applications. Advances in Petri Nets, *Lecture Notes in Computer Science (LNCS)*, vol. 1492, Springer-Verlag, 1998, pp. 154-236.

[7] A. Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, and A. Yakovlev. Logic decomposition of speed-independent circuits. *Proceedings of IEEE*, vol. 87, no. 2, pp. 347-362, Feb. 1999.

[8] J. Cortadella, M. Kishinevsky, L. Lavagno and A. Yakovlev. Deriving Petri Nets from Finite Transition Systems, *IEEE Transactions on Computers*, vol. 47, no. 8, pp. 859-882, Aug. 1998.

[9] A. Yakovlev. Designing Control Logic for Counterflow Pipeline Processor Using Petri nets, *Formal Methods in Systems Design*, vol. 12, no. 1, pp. 39-71, Kluwer AP, Jan. 1998.

[10] A. Yakovlev, F. Xia and D. Shang, Synthesis and implementation of a signal-type asynchronous data communication mechanism. *Proc.International Symposium on Advanced Research in Asynchronous Circuits and Systems*, Salt Lake City, March 2001, pp. 127-136.

[11] A. Yakovlev, L. Lavagno, and A. Sangiovanni-Vincentelli. A unified signal transition graph model for asynchronous control circuit synthesis. *Formal Methods in System Design*, vol. 9, no. 3, pp. 139-188, Kluwer AP, Nov. 1996.

[12] A. Yakovlev, M. Kishinevsky, A. Kondratyev, L. Lavagno and M. Pietkiewicz-Koutny. On the Models for Asynchronous Circuit Behaviour with OR Causality. *Formal Methods in Systems Design*, vol. 9, no. 3, pp. 189-234, Kluwer AP, Nov. 1996.

[13] A. Yakovlev, A.M. Koelmans, A. Semenov, D.J.Kinniment, Modelling, Analysis and Synthesis of Asynchronous Control Circuits Using Petri Nets, *Integration: the VLSI journal*, vol. 21, pp. 143-170 (1996).

[14] A. Yakovlev, A.M. Koelmans, and L. Lavagno. High level modelling and design of asynchronous interface logic, *IEEE Design and Test of Computers*, Spring 1995, pp. 32-40.

[15] L. Rosenblum and A. Yakovlev. Signal graphs: from self-timed to timed ones, *Proc. of the Int. Workshop on Timed Petri Nets*, Torino, Italy, July 1985, IEEE Comp. Soc. Press, 1985, pp. 199-207.

[16] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev, *Logic Synthesis of Asynchronous Controllers and Interfaces*. Springer Series in Advanced Microelectronics, vol. 8, Springer, 2002, ISBN-3-540-43152-7

[17] A. Yakovlev, L. Gomes and L. Lavagno (Eds.) *Hardware Design and Petri Nets*. Kluwer Academic Publishers, Boston, 2000, ISBN 0-7923-7791-5.

[18] D.J.Kinniment, A. Bystrov, A.V. Yakovlev Synchronization Circuit Performance, IEEE Journal on Solid-State Circuits, vol. 37, no. 2, Feb. 2002, pp. 202-209

[19] J. Cortadella, M.Kishinevsky, S.M. Burns, K.S. Stevens, A. Kondratyev, L. Lavagno, A. Taubin, A. Yakovlev, Lazy Transition Systems and Asynhronous Circuit Synthesis with Relative Timing Assumptions. *IEEE Transactions of CAD*, Vol. 21, No. 2, Feb. 2002, pp. 109-130.

[20] A. Burns, A.J. Wellings, F. Burns, A.M. Koelmans, M. Koutny, A. Romanovsky, A. Yakovlev. Modelling and Verification of an Atomic Action protocol implemented in ADA, *Int. J. of Comp. Systems and Eng.*,vol. 16 (3): 173-182, May 2001.

[21] D. J. Kinniment, A.V. Yakovlev, and B. Gao. Synchronous and Asynchronous A-D Conversion, *IEEE Transactions on VLSI systems*, Vol. 8 No. 2 Apr. 2000, pp. 217-219.

[22] F. Burns, A. Koelmans and A. Yakovlev. WCET Analysis of Superscalar Processors Using Simulation With Coloured Petri Nets, Real-Time Systems, *International Journal of Time-Critical Computing Systems*, Volume 18, Issue 2/3, May 2000, Kluwer Academic Publishers, pp. 275-288.

[23] D.J. Kinniment, A.V. Yakovlev. Low power, low noise micropipelined flash A-D converter, *IEE Proc. Circuits, Devices Systems*, vol. 146, no.5, October 1999, pp 263-267

[24] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno, E. Pastor, and A. Yakovlev. Decomposition and technology mapping of speed-independent circuits using boolean relations. *IEEE Trans. on CAD*, Vol. 18, No. 9, Sep. 1999, pp. 1221-1236

[25] L. Lloyd, K. Heron, A. Yakovlev, and A.M. Koelmans. Asynchronous microprocessors: from high level model to FPGA implementation. *J. of Syst. Arch.* , vol. 45 (1999), pp. 975-1000, Elsevier

[26] A. Bystrov and A. Yakovlev. Ordered arbiters. *Electronics Letters*, 27th May 1999, Vol. 35 (11): 877-879.

[27] F.B. Burns, A.M. Koelmans, and A.V. Yakovlev. Analysing superscalar processor architectures with coloured Petri nets. *Int. Journal on Software Tools for Technology Transfer*, Vol.2, No.2, December 1998, Springer, pp. 182-191

[28] A. Kondratyev and M. Kishinevsky and A. Yakovlev. Hazard-free implementation of speed-independent circuits, *IEEE Trans. on CAD*, vol. 17, no. 9, pp. 749-771, Sept. 1998

[29] I.G. Clark, F. Xia, A.V. Yakovlev and A.C. Davies, Petri net models of latch metastability, *Electronics Letters*, 2nd April 1998, Vol. 34, No.7, pp. 635-636

[30] A. Semenov, A.M. Koelmans, L. Lloyd and A. Yakovlev. Designing an asynchronous processor using Petri nets, *IEEE Micro*, Vol. 17, No. 2 (March/April 1997), pp. 54-64

[31] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev. A region-based theory for state assignment in speed-independent circuits, *IEEE Trans. on CAD*, vol. 16, no. 8, pp. 793-812, Aug. 1997

[32] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev, Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers, *IEICE Trans. Inf. & Syst.*, Vol. E80-D, No.3, March 1997, pp. 315-325.

[33] A. Yakovlev. Is the Die Cast for the Token Game? (invited paper) Proc. of 23rd International Conference on Applications and Theory of Petri nets (ICATPN 2002), Adelaide, Australia, June 2002, LNCS 2360, Springer, pp. 70-79.

[34] V. Varshavsky, A. Yakovlev, V. Marakhovsky and I. Levin, Self-Timing, Self-Checking and Self-Recovery, Proc. 4th Euromicro Conference on Massively Parallel Computing Systems, Ischia, Italy, 9-12 April 2002. (http://www.mpcs.org/MPCS02/Papers/8/Paper.pdf)

[35] A. Madalinski, A. Bystrov, A. Yakovlev, Visualisation of Coding Conflicts in Asynchronous Circuit Design, IEEE/ACM 11th International Workshop on Logic and Synthesis, New Orleans, June 2002.

[36] A. Bystrov, A. Yakovlev, Synthesis of Asynchronous Circuits with Predictable Latency, IEEE/ACM 11th International Workshop on Logic and Synthesis, New Orleans, June 2002.

[37] D.Shang, F.Xia and A.Yakovlev, Asynchronous Circuit Synthesis via Direct Translation, Proc. ISCAS'02, Scottsdale, Arizona, May 2002, IEEE, Volume III, pp. 369-372.

[38] O. Maevsky, D.J. Kinniment, A.Yakovlev, A. Bystrov Analysis of the oscillation problem in tri-flops, Proc. ISCAS'02, Scottsdale, Arizona, May 2002, IEEE, volume I, pp.381-384.

[39] V. Khomenko, M. Koutny and A. Yakovlev Detecting state coding conflicts in STGs using integer programming, Proc. DATE'02, Paris, March 2002, IEEE CS Press, pp. 338-345.

[40] A. Bystrov, M. Koutny and A. Yakovlev, Visualization of partial order models in VLSI design flow, Proc. DATE'02, Paris, March 2002, IEEE CS Press, pp. 1089-1090.

[41] A. Bystrov and A. Yakovlev, Asynchronous Circuit Synthesis by Direct Mapping: Interfacing to Environment, Proc ASYNC'02, Manchester, April 2002, IEEE CS Press, 127-136.

[42] D.J. Kinniment, O. Maevsky, A. Bystrov, G. Russell and A. Yakovlev, On-chip Structures for Timing Measurements and Test, Proc ASYNC'02, Manchester, April 2002, IEEE CS Press, pp. 190-197

[43] F. Xia, A. Yakovlev, I.G. Clark and D. Shang. Data Communication in Systems with Heterogeneous Timing, *IEEE Micro*, November-December 2002.

[44] A. Yakovlev and F. Xia, Towards synthesis of asynchronous communication algorithms, *Proc. Int. Workshop on Synthesis of Concurrent Systems*, ICATPN'01 and ICACSD'01, Newcastle upon Tyne, June 2001. Appeared as book chapter in: *Synthesis and Control of Discrete Event Systems* (Eds. B. Caillaud, P. Darondeau, L. Lavagno and X. Xie), Kluwer Academic Publishers, ISBN-0-7923-7639-0, January 2002, pp. 57-75.

[45] N. Starodoubtsev, A. Bystrov, and A. Yakovlev Semi-modular latch chains for asynchronous circuit design. *Proc. 10th Int. Workshop on Power and Timing Modelling, Optimization and Simulation (PATMOS'2000)*, Goetingen, Germany, Sept. 2000, D. Soudris, P. Pirsch, E. Barke (Eds), Lect. Notes Comp. Sci. 1918, Springer, pp. 168-177.

[46] J. Cortadella, M. Kishinevsky, A. Kondratyev, L. Lavagno and A. Yakovlev, Hardware and Petri Nets: Application to Asynchronous Circuit Design, *Application and Theory of Petri Nets 2000*, Lecture Notes in Computer Science vol. 1825, pp. 1-15, Springer Verlag, June 2000

[47] A. Bystrov, D. Kinniment, and A. Yakovlev. Priority Arbiters. *Proc. Sixth Int. Symp. on Advanced Research in Asynchronous Circuits and Systems (ASYNC2000)*, April 2000, Eilat, Israel, IEEE Computer Society Press, pp. 128-137.

[48] F. Xia, A. Yakovlev, D. Shang, A. Bystrov, A. Koelmans, and D. Kinniment. Asynchronous Communication Mechanisms using Self-timed Circuits. *Proc. Sixth Int. Symp. on Advanced Research in Asynchronous Circuits and Systems (ASYNC2000)*, April 2000, Eilat, Israel, IEEE Computer Society Press, pp. 150-159.

[49] H. Saito, A. Kondratyev, J. Cortadella, L. Lavagno, A. Yakovlev. What is the cost of delay-insensitivity? *Proc. IEEE/ACM Int. Conf. on CAD (ICCAD'99)*, San Jose, Nov. 1999, IEEE Comp. Soc. Press.

[50] A. Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, A. Yakovlev. Automatic synthesis and optimization of partially specified asynchronous systems. *Proc. of 36th ACM Design Automation Conference (DAC'99)*, June 1999, New Orleans, LA, pp. 110-114.

[51] A. Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, A. Taubin, and A. Yakovlev. Lazy Transition Systems: Application to Timing Optimization of Asynchronous Circuits. *Proc. IEEE/ACM Int. Conference on CAD (ICCAD'98)*, November 1998, San Jose, IEEE Comp Soc. Press, pp. 324-331

[52] I. Mitrani and A. Yakovlev. Tree Arbiter with Nearest-Neighbour Scheduling. Proc. of the 13th Int. Symp. on Comp. and Inf. Sci. (ISCIS'98), Oct. 1998, Belek-Anatlya, Turkey. In: Adv. in Comp. and Inf. Sci. (Eds. U. Gudukbay, T. Dayar, A. Gursoy, E. Gelenbe) Concurrent Systems Engineering Series Vol. 53, pp. 83-92, (IOS Press)

[53] W. Vogler, A. Semenov and A. Yakovlev. Unfolding and Finite Prefix for Nets with Read Arcs. *Proceedings of CONCUR'98*, Nice, France, Sept. 1998, LNCS, vol. 1466, Springer pp. 501-516

[54] L. Lloyd, A. V. Yakovlev, E. Pastor, A.M. Koelmans Estimations of power consumption in asynchronous logic as derived from Graph Based Circuit Representations. *Proc. of PATMOS'98*, Techn. Univ. of Denmark, Oct. 1998, pp. 367-376

[55] A.Kondratyev, J. Cortadella, M. Kishinevsky, L. Lavagno, A.Taubin and A. Yakovlev. Identifying state coding conflicts in asynchronous system specifications using Petri net unfoldings, *Proc. of Int. Conf. on Appl. of Concurrency to System Design (CSD'98)*, March 1998, Aizu-Wakamatsu, Japan, IEEE Comp. Soc. Press, pp. 152-163

[56] D.J. Kinniment, B. Gao, A. Yakovlev, F. Xia. Towards asynchronous A-D conversion, *Proc. 4th Int. Symp. on Adv. Res. in Asynchronous Circuits and Systems (Async'98)*, March-April 1998, San Diego, CA, IEEE Comp. Soc. Press, pp. 206-215

[57] A. Semenov, Verification and Synthesis of Asynchronous Control Circuits Using Petri Net Unfoldings. PhD Thesis, Newcastle upon Tyne, 1998 (available on request from Alex.Yakovlev@ncl.ac.uk).

[58] M. Pietkiewicz-Koutny. Relating Formal Models of Concurrency for the Modelling of Asynchronous Digital Hardware. PhD Thesis, Newcastle upon Tyne, 2000 (available on request from Alex.Yakovlev@ncl.ac.uk).

[59] A. Semenov, A. Yakovlev, E. Pastor, M. Pena, J. Cortadella, and L. Lavagno Partial order approach to synthesis of speed-independent circuits. *Proc. 3rd Int. Symp. on Adv. Res. in Async. Circuits and Systems (Async'97)*, Eindhoven, Holland, April 1997, IEEE Comp. Soc. Press, pp. 254-265

[60] A. Semenov, A. Yakovlev, E. Pastor, M. Pena, and J. Cortadella, Synthesis of Speed-independent circuits from STG-unfolding segment *Proc. 34th ACM/IEEE Design Automation Conference (DAC'97)*, Anaheim, CA, June 1997, pp. 16-21

[61] A. Yakovlev Solving ACiD-WG design problems with Petri net based methods. Proc. ESPRIT ACiD-WG Workshop on Asynchronous Circuit Design, Groningen, Sept. 9-10, 1996, TR CSN9602, Computer Science Notes Series, University of Groningen. (available on request from Alex.Yakovlev@ncl.ac.uk).

[62] N.Starodoubtsev, A. Yakovlev and S. Petrov, Use of VHDL-based environment for interactive synthesis of asynchronous circuits. Proceedings VHDL Forum in Europe Spring'96 Working Conference, Dresden, Germany, May 1996, pp. 21-33.

[63] A.Semenov and A. Yakovlev, Verification of asynchronous circuits based on Time Petri Net unfolding, Proc. 33rd ACM/IEEE Design Automation Conference (DAC'96), Las Vegas, June 1996, pp. 59-63.

[64] A. Semenov and A. Yakovlev. Combining partial orders and symbolic traversal for efficient verification of asynchronous circuits. Proc. IFIP Int. Conference on Computer Hardware Description Languages, (CHDL'95),Chiba, Japan, August-September 1995, pp. 567-573.

[65] Yakovlev, A. Designing arbiters using Petri nets. Proceedings of the 1995 Israel Workshop on Asynchronous VLSI, Nof Genossar, Israel, March 1995, VLSI Systems Research Center, Technion, Haifa, Israel, pp. 178-201 (available on request from Alex.Yakovlev@ncl.ac.uk).

[66] Yakovlev,A., V.Varshavsky, V.Marakhovsky and A.Semenov, Designing an asynchronous pipeline token ring interface, Proc. of 2nd Working Conference on Asynchronous Design Methdologies, London, May 1995, IEEE Comp. Society Press, N.Y., 1995, pp. 32-41.

[67] Semenov, A., and Yakovlev, A. Event-Based Framework for Verifying High-Level Models of Asynchronous Circuits, Department of Computing Science. TR Series No. 487, May 1994.

[68] Yakovlev, A., and Petrov, A., Rosenblum L. Synthesis of asynchronous control circuits from symbolic signal transition graphs, Asynchronous Design Methodologies (Ed. S. Furber, M. Edwards), IFIP Transactions A-28, North-Holland, 1993 (Proc. IFIP 10.5 Working Conf., Manchester, March-April 1993), pp. 71-85.

[69] Yakovlev, A., Petrov, A., Lavagno, L. A Low Latency Asynchronous Arbitration Circuit, IEEE Trans. on VLSI Systems, vol. 2, No. 3, Sept. 1994, pp. 372-377

[70] Yakovlev, A.V. Structural technique for fault-masking in asynchronous interfaces. IEE Proceedings E (Computers and Digital Techniques), Vol. 140, No.2, March 1993, pp. 81-91

[71] Yakovlev, A.V., Lavagno, L., and A. Sangiovanni-Vincentelli. A unified signal transition graph model for asynchronous control circuit synthesis. Proc. Int. Conf. on CAD (ICCAD'92), Santa Clara, CA, November 1992, IEEE Comp. Society Press, N.Y., pp. 104-111.

[72] Yakovlev, A.V. On limitations and extensions of signal transition graph model for designing asynchronous control circuits. Proc. Int. Conf. on Computer Design (ICCD'92), Cambridge, MA, October 1992, IEEE Comp. Society Press, N.Y., pp. 396-400.

[73] Yakovlev, A.V. A structural technique for fault-protection in asynchronous interfaces. Proc. Int. Symp. on Fault-Tolerant Computing, (FTCS'92) Boston, MA, July 1992, IEEE Comp. Society Press, N.Y., pp. 288-296.

[74] Yakovlev, A.V. and Petrov, A. Petri nets and parallel bus controller design. Proc. of 11th Int. Conf. on Applications and Theory of Petri Nets, Paris, France, June 1990, pp. 244-263.