IGR Report GR/R16754/01
Behavioural Synthesis of Systems with Heterogeneous Timing (BESST)

1 Background/Context

The state of the art in asynchronous design tools is presented in the 3nd edition of ACiD-WG report "Design, Automation and Test for Asynchronous Circuits and Systems" by D.A.Edwards and W.B.Toms, which is available http://www.scism.sbu.ac.uk/ccsv/ACiD-WG/. This site also contains a report on the status of asynchronous design in industry. It clearly reflects an increased industrial interest in asynchronous design both in the UK and abroad, particularly in response to the design and test challenges of the deep submicron era (cf. ITRS-2003). The role of and demand for a flexible design flow is rapidly increasing as was emphasized at the last four (EU-funded) ACiD-WG workshops. One of the key stages in a successful design flow, with a good balance between productivity and optimality, would be automatic synthesis from behavioural descriptions.

The BESST project has been part of Newcastle's steady involvement in research in asynchronous synthesis over the last ten years. This research found its support from EPSRC through the ASAP (GR/J52327) grant, in collaboration with the University of Bristol (Prof. E. Dagless), the MOVIE (GR/M94366) and visiting fellowships (GR/J72486, J78334, L24038, M94359) of Professors M. Kishinevsky, L. Lavagno, A. Kondratyev and N. Starodoubtsev, which was accompanied by travel grants from the British Council and Spanish Ministry of Science and Culture to support collaboration with the Group of Prof. J. Cortadella from UPC (Barcelona). The main result of this work was a set of synthesis methods and a software tool Petrify described in the monograph [1], and more recently visualisation support for synthesis presented in [2].

The research carried out under BESST was a natural progression from these developments as well as Newcastle's long-standing research in the area of concurrent system modelling, analysis and synthesis. This project tackled a number of limitations of the state-of-the-art of Petrify (and more indirectly, other relevant tools such as MINIMALIST) and was aimed at extending the power of synthesis techniques and the applicability of synthesis tools for a broader range of users – a critical issue for the process of promulgation of asynchronous design techniques into industrial domains. Particular emphasis was put on extending existing methods for the synthesis of asynchronous circuits to cope with much larger complexity, by means of using partial order representations for logic synthesis and direct mapping techniques, as well as to work with more pragmatic front-end languages such as VHDL and Verilog.

This research was closely related with other projects being conducted at Newcastle, such as one on asynchronous communication mechanisms (COHERENT, GR/R32666, in collaboration with Kingston University), on synthesis and testing of low latency asynchronous circuits (STELLA, GR/S12036), and more recently the SCREEN project (GR/S81421), focused on the development of a design flow for security hardware.

2 Key Advances and Supporting Methodology

The prime target of BESST was Distributed, Concurrent and Asynchronous (DCA) (hardware) communication and operation kernels of Systems-on-Chips (SoCs). Such kernels were seen in two areas. Firstly, they were in protocols and associated "glue logic", used to interface functional parts of SoCs, such as interface "wrappers" for existing IPs. Secondly, they were controllers for new components, which could themselves be seen as future IPs, whose designs are susceptible to the effects of greater interconnect dominance. Existing synthesis tools were observed to fall short of providing good support for designing sufficiently complex asynchronous controllers in an efficient and reliable way.

BESST proposed to synthesise behaviourally critical DCA parts of SoCs (semi-)automatically from their high-level specifications. The practical results of BESST were aimed to support the design of such systems by people with limited asynchronous design experience and at a much lower cost. In the longer term we aimed at a theoretical underpinning for self-timed systems synthesis, and a set of techniques and tools enabling the application of theory to designing interfaces and controllers of SoCs. This step in further development of system synthesis tools is crucial for their greater exploitation by practical designers and their incorporation into commercial CAD packages. Such tools also play an increasingly key role in training asynchronous designers.

The objectives of the project were twofold. The first group of objectives (P1 and P2) was concerned with a *practical* development of the modelling and algorithmic components of a fully automated design flow for complex self-timed controllers. Standard HDLs (e.g. VHDL and Verilog) were seen as front-end specification languages and Petri nets as an intermediate representation form. As pointed out by the referees of the proposal these objectives were to be central to the project.

The second group of objectives (T1 and T2) was of more speculative nature. It was concerned with the development of *theoretical* methods for constructing DCA systems from sequential specifications, where new formal foundations

had to be laid, namely the definition of adequate behavioural semantics, equivalence notions and related semantic transformations.

2.1 P1: Automatic extraction of Petri net models from HDL specifications

The basic ideas for the new flow were outlined in [34, 42]. The following advancements were made in this area.

High-level Synthesis from HDLs. The focus of this research has been on the *semantical* translation of standard HDLs, such as VHDL and Verilog, in their behavioural domain, into Petri nets used for intermediate representation. The method described in [13, 43, 30] for VHDL, generates two types of nets, Labelled Petri nets for control and Coloured Petri nets for datapath. Speed-independent circuits are then obtained from these nets via direct translation using David cells. The method was applied to a number of design examples and demonstrated speed improvements (30%) in circuits obtained by our approach to those of syntax-based translation techniques (via handshake components) used in Tangram and Balsa. This method was subsequently extended to Verilog in [25] and has led to a new tool, VERISYN, described in detail in [47]. For 'more asynchronous' HDLs, such as Communicating Hardware Processes, with channels and probes, the translation to Petri nets was developed in [19].

Scheduling and allocation. One of the referees of our proposal suggested, quite rightly, that our behavioural synthesis should also include scheduling and allocation. We followed this recommendation and addressed the problem of scheduling using the new heuristic approach to scheduling called "closeness tables" fully described in [11]. This approach allows to pack operations for scheduling and binding tightly, exploiting similarity in their input and output resources. Low interconnect solutions can be generated in fast execution time. The experimental results applied to non-pipelined and pipelined architectures showed the advantages against known scheduling algorithms. This method is being applied in the VERISYN tool.

Within BESST we continued our collaboration with the Petrify consortium, involving Intel Res. Labs, on the development of models of asynchronous circuits with timing constraints. The latter play crucial role in supporting logic synthesis in Petrify, through solving implementability problems (such as CSC) and logic circuit optimization [3].

2.2 P2: Synthesis of controllers combining direct translation and logic synthesis

The overview of a number of asynchronous synthesis methods available to date and their comparison in terms of computation time, area and speed has been carried out in [10]. It shows the superiority of logic synthesis approaches over syntax-driven translation as the latter allow only peep-hole optimizations. Within logic synthesis methods techniques based on direct mapping can cope with complexity by avoiding state-space explosion but at the cost of more area. Direct mapping techniques used in the OptiMist tool are very competitive compared to Petrify. Overall, the following advancements were made in this area.

Direct mapping of Labelled Petri nets. The method of direct mapping of Petri net specifications into control circuits based on David cells originates from from Varshavsky's work in the 80s. It has now been fully automated in the PN2DCS tool [46], and augmented with semi-automatic optimisation strategies for high-speed and smaller area as presented in [32].

Direct mapping of STGs. As our pre-BESST research showed, the use of direct mapping applied to STGs (Petri nets with binary signal transitions) could often result in higher speed control circuits than those obtained by logic synthesis by Petrify. Intuitively this was due to the fact that the depth of logic produced by Petrify in the cases where complete state encoding was solved by inserting additional signals was quite large. Contrary to that, direct mapping could produce circuits without inserting state signals between inputs and outputs, thus leading to low latency (see next paragraph). These ideas were investigated and formalised in [33], using the decomposition of the controller into a tracker and i/o interface. The method was further automated in the OptiMist tool [29, 28], with a number of optimisation heuristics helping to reduce area to be competitive with Petrify solutions.

Synthesis of low latency controllers and interfaces. This part of research has been particularly important because it addresses the timing-driven aspects of controller synthesis – one of the crucial objectives of BESST. Several useful results were obtained here.

The idea of structuring the control into state-evaluation part (called "bouncer") and i/o interface part ("tracker") has also been investigated in the logic synthesis context [31], aiming to retain the advantages of Petrify in exploring the state space for optimum size. This work has shown an interesting potential for exploring synergy between logic synthesis and direct mapping to generate low latency controllers of small size (e.g. for low lower).

Asynchronous design is known to benefit from the application of weak precedence and early evaluation to achieve greater performance in pipelines and communications. These ideas have recently attracted the attention of many researchers. Our pre-BESST experience in theoretical modelling of structures with OR-causality (from the 80s and 90s) helped us tackle these problems in a generic way using truly causal semantics of Petri nets. The problem of the

implementation of buffered OR-causality remained open until our work in [26], where a number of Petri net and logic level solutions are presented.

The practical application of the synthesis for low-latency was in designing control logic for a challenging duplex communication protocol (invented by S.B. Furber). This design was carried out from a formal specification of the protocol (involving two arbitration points) down to the circuit simulation in Cadence as described in [12]. It clearly demonstrates the advantages of the direct mapping methods for medium-size communication controllers compared to logic synthesis using Petrify.

Synthesis of controllers and interfaces with relative timing. One of the aspects of timing-driven synthesis is to exploit knowledge of delay information in logic design of high-performance controllers. The application of timing constraints allows to restrict concurrency without introducing causal dependencies and thus obtain many don't care conditions "for free". The theoretical investigation of the models in the context of the STG-based synthesis has been carried out, and its use in many asynchronous controller benchmarks was presented in [3, 1].

Synthesis of controllers with arbiters. Part of this objective has been addressed in the work describing synthesis of arbiters from CHP specifications using Petri net models of probe [19]. Ways of direct mapping of Petri nets specifications using arbitration components have been presented in [41, 12].

Synthesis of circuits with more delay-insensitivity. In addition to timing-driven synthesis, the project targetted layout-driven synthesis, based on transformation at the STG level to address the issues of interconnect delay and other physical variations such as varying thresholds in transistors, crucial for deep-submicron (100nm and beyond). The above-mentioned work on direct mapping of STGs and OptiMist use the fact that i/o interface between the circuit and the environment must be delay-insensitive(DI). STG transformations and re-synthesis of controllers with DI interface was investigated in [14]. A method of behavioural refinement for controllers built of NAND and NOR gates modified for additional transistor switching acknowledgement has been developed in [27]. This work has also indicated perspectives for constructing non-trivial DI circuits in CMOS processes.

Synthesis of controllers using Petri nets unfoldings While work on direct mapping was the major focus of our research, we continued to explore the boundaries for logic synthesis based on optimisation, clearly having much potential according to our comparative analysis in [10]). This research went in two directions. One was to advance the methods of interactive synthesis of asynchronous controllers, started under MOVIE, where the visualisation of state coding conflict cores [2, 40] and conflict detection based on STG unfoldings [45, 35, 39, 7], played a key role. The most recent results were in developing a method for solving encoding conflicts using both signal insertion and concurrency reduction [48].

The other direction was developing a completely synthesis flow for large and medium size STGs based on unfoldings, to avoid state-space explosion [20]. This work complements nicely the most recent developments at UPC, Barcelona, on structural methods for synthesis of large STGs, and both approaches were compared in [15], which presents the state of art in logic synthesis from STGs (reflecting also on most recent work in this area by W.Vogler et al. at Augsburg and T.Yoneda et al.in Tokyo).

2.3 T1: Equivalences and transformations for synthesis

Theoretical work was mainly concentrated on the development of important semantical foundations: (1) between process-level models of DCA systems and Petri nets, (2) between Petri nets of different type and their partial order semantics, and (3) equivalence preserving refinements for process models and Petri nets.

New Petri net semantics for Process algebras Petri net semantics for pi-calculus, important for modelling distributed and heterogeneous systems, has been dveloped in [21, 22]. An entirely new branch of Petri nets box calculus, to model DCA systems with non-rendez-vous (asynchronous) interaction and causal timing, was developed in [9, 37]. A general interpretation of Process Algebras via Petri nets has been presented in [17].

Partial order semantics based on unfoldings In [6, 38], we defined branching processes and unfoldings of high-level Petri nets and proposed an algorithm which builds finite and complete prefixes of such nets. We established an important relation between the branching processes of a high-level net and those of its low-level *expansion*, viz. that the sets of their branching processes are the same, allowing us to import results proven for low-level nets. Among such results are the canonicity of the prefix for different cutting contexts. Asynchronous circuits are often efficiently modelled by Petri nets with inhibitor arcs. In [5, 36] we laid a formal foundation for further developments in the area of unfolding nets with inhibitor arcs by constructing theory of causal processes for them.

Refinements at the process and net levels The quality of synthesis of DCA systems will increasingly rely on the behavioural model transformations as observed in [34, 15]. It was realised that the basis for such transformations lies in equivalences at the level of process interfaces. A method for relating processes with different interfaces, due to refinements between levels of abstraction, was developed in [8, 18]. Examples of such refinements in practice were

found e.g. in our duplex channel design [12]. Work of Prof.E.Best at Newcastle has contributed to the development of methods for analysis of k-bounded conflict-free nets (which are typically produced from HDL-type system specifications) by separating their sequence behaviours into k sequences of 1-safe nets [49].

2.4 T2: Synthesis of self-timed interfaces and protocols

The problem of distributing concurrent actions between interactive components of a DCA system with delay-insensitive interfaces is is highly important for future developments in SoCs. Here our work was focused on two major parts. One was characterising classes of transition systems describing asynchronous interactions in an adequate way. The other was extending the power of theory of regions and tools such as Petrify to address synthesis of Petri nets which could model asynchronous communications as a composition of finite state processes either via shared memory or via handshakes, as defined by a specification.

Petri net synthesis from labelled transition systems. In this part a class of synthesizable transition systems and corresponding nets was extended to those of systems with step transition semantics (which, e.g., could nicely capture GALS systems) and Petri nets with inhibitor arcs [4].

Synthesis of asynchronous communication mechanisms (ACMs). Addressing this issue in [16], we developed a formal way of synthesizing Petri net models for ACMs with multi-slot storage and blocking and non-blocking write and read interfaces, using state graphs. These nets are synthesised by an appropriate modification of the canonical procedure for synthesis of elementary nets from transition systems. In our work we have also addressed possible ways of further translation of Petri net models of ACMs to hardware and software implementations.

2.5 Prototype tools

The **fourth and final objective** of BESST was the development of software tools and integration with the related existing synthesis software such as Petrify. The following tools (can be downloaded from `http://async.org.uk/besst/`) were developed within the project, and were extensively used to obtain experimental results:

- VERISYN [47] – a tool for the high-level synthesis of asynchronous circuits. It inputs behavioural descriptions in Verilog and outputs an intermediate direct mapping format in the form of Petri nets (high and low level) which can subsequently be passed to direct mapping tools (PN2DCS) for asynchronous speed-independent implementation of datapath and control circuits.
- PN2DCS [46] – a tool for the direct mapping of Petri net specifications into speed-independent control circuits based on David Cells. It inputs specifications similar to Petrify and produces an output optimized control netlist description The tool has subsequently been extended for the direct mapping from higher level net descriptions to both control and datapath.
- OPTIMIST [28, 10] – a tool for direct mapping of STGs to control logic. It uses the same input format as Petrify and produces Verilog netlists for logic implementations. One of the options allows to generate transistor level circuits based on the so called "fast David cells", operating under relative timing.
- ConfRes [2, 40] – a tool for visualising cores of state coding conflicts in STG unfolding prefix and interactive resolution of conflicts using height map heuristics. The tool is compatible with Petrify. For actual graph displaying ConfRes calls `dot`, a graph drawing software from Graphviz.

2.6 Experimental designs

Our experimental designs included a duplex communication channel [12], DMA controller [41, 13] and an asynchronous unit for Advanced Encryption Standard (AES) [23, 24, 47]. These examples were designed using the methods and tools developed in this project. They were implemented using AMS 0.6um and 0.35um cell libraries with a number of in-house designed cells for mutex elements, C-gates etc, and simulated using SPICE in Cadence. The development of a demonstrator chip (for security applications) was phased out of the BESST project due to shortage of funds caused by the redirection of some of the consumables budget to address staff retention needs. The demonstrator development is planned to be combined with the STELLA and SCREEN to make it cost effective in the overall research context.

2.7 Research output

Overall, the project has resulted in contributions to one monograph, 13 journal papers, 4 book chapters, 26 conference papers (22 of which are peer-refereed), two PhD Theses (plus one PhD thesis, by D.Sokolov, is in write-up), 14 technical reports, one tutorial, and four tools.

3 Research Impact and Benefits to Society

Our research group at Newcastle considers this research to have been very successful. All of the tasks have been completed and in many cases the results exceeded the orginal expectation considerably. For example, the major impact on the practical side has been in developing the high level design flow based on Verilog (including scheduling), direct mapping of STGs for low latency circuits, and in advancing STG unfoldings to logic synthesis. On the theoretical side, the major advances are in creating formal links between process algebras (including pi-calculus) and Petri nets, new refinement and transformation techniques and automating the construction of communicating mechanisms from sequential specifications. Publications, numerous presentations of academic and visiting staff, RAs, PG students in our weekly seminar of the Asynchronous Systems Laboratory (see `http://www.cs.ncl.ac.uk/events/ASL`), have played an important role in promoting these results into a number of follow-up projects and external contacts. With this research Newcastle has been part of the Petrify team shortlisted for the Descartes 2002 Prize (see ftp://ftp.cordis.lu/pub/descartes/docs/2002descartes_prize_catalogue-en.pdf)

The results of the project are relevant to potential users at three different levels:

- The use of theoretical results, especially in the unfolding-based characterisation and ACM synthesis, will benefit researchers in area of designing complex asynchronous and concurrent systems (verification and synthesis) in a wide range of industrial applications.
- The use of Petri net based algorithms and tools in asynchronous design flow will benefit CAD developers and vendors working on promoting asynchronous design techniques in microelectronics. The new synthesis tools have been presented at the First EDA Tools forum at IEE in November 2004.
- The use of synthesis tools has already opened up a new active collaborative link with Atmel UK, a well known smart card design company. This has led to SCREEN and industrial test chip development. Alongside Petrify, the new tools will be used in the Winter school on asynchronous design in Cambridge in January 2005.

For example, these results will be useful to the UK research in asynchronous design, supported by EPSRC at Manchester and Cambridge (projects on Balsa synthesis environment and GALS). We are also planning to use these techniques in promoting asynchronous design at Atmel-UK.

4 Explanation of Expenditure

The expenditure plans in the original proposal have been followed without significant changes. The biggest change was the relocation of some of the consumables budget (chip fabrication) in order to cover the shortfall on salaries and travel. (This possibility was actually anticipated by the panel, which in the specific grant conditions, recommended to treat chip fabrication as lower priority to the fulfilment of the objectives P1 and P2.) One of the RAs on the project, Dr. F.Burns, was required to be placed on a higher grade than was budgeted for. Additionally the project involved a monthly appointment of Prof.E.Best, an internationally recognised authority in the area of concurrent systems, whose contribution in the theory part was very useful and helped to lay the foundation for future work in developing formal models for GALS. The extra travel costs were necessary to support our high conference actitivity, for example Dr. V.Khomenko's and A.Madalinski's attendance at ACSD conferences. BESST partly supported the visit of S.Tyerman, from the Univ. of South Australia, who contributed to our high-level synthesis work (cf. [44]) The fabrication of a chip has been rescheduled to other projects, STELLA and SCREEN, combining their budgets, where we plan to tape out and produce a test chip with extra security measures using tools developed in BESST and further.

5 Further Research or Dissemination Activities

The research outcome from this grant is directly applicable for exploitation in our four EPSRC grants, on synthesis and testing of low latency asynchronous logic (STELLA, GR/R12036), Secure circuit design (SCREEN, GR/S81421), in collaboration with Atmel Smart Cards UK, Causal Semantics of Nets with inhibitr arcs (CASINO, GR/T19407) and most recently, Next Generation Of Interconnection Technology For Multiprocessor SoC (EP/C512812/1) in collaboration with Southampton University and MBDA UK Ltd.

The ideas of interactive synthesis from STGs and analysis of related properties is now part of PhD research of D. Koppad, J. Murphy, Yu Zhou, S.Dasgupta and J. Forbes.

Besides publications, this research has had its impact on preparing two advanced tutorials at the international conferences: Adv. Tutorial on Hardware Design and Petri Nets at ICATPN in Aarhus, June 2000, and Tutorial on Asynchronous Circuit Design in ASP-DAC-VLSI, Bangalore, Jan. 2002 [50], our invited lectures at the fourth advanced course on Petri nets in Eichstätt in September 2003 (`http://www.acpn.de`), and ACiD-WG Summer School in Grenoble in July 2002. It will also be put forward in our presentations in the forthcoming conferences on Applications of Concurrency in Systems Design (ACSD'05), DATE 2005, ACiD-WG Winter school in Cambridge in Jan. 2005, and tutorials on asynchronous systems at Petri nets 2005 and Formal Methods Europe in July 2005.

References

[1] 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)

[2] A. Madalinski, A. Bystrov, V. Khomenko and A. Yakovlev. Visualisation and resolution of encoding conflicts in asynchronous circuit design, IEE Proc. CDT, vol. 150, No.5, Sept. 2003, pp. 285-293 (Special issue of Best papers at DATE2003).

[3] J. Cortadella, M.Kishinevsky, S.M. Burns, K.S. Stevens, A. Kondratyev, L. Lavagno, A. Taubin, and A. Yakovlev: Lazy Transition Systems and Asynhronous Circuit Synthesis with Relative Timing Assumptions. IEEE Trans. of CAD, Vol. 21, No. 2, pages 109-130 (2002).

[4] M.Pietkiewicz-Koutny: Synthesising Elementary Net Systems with Inhibitor Arcs from Step Transition Systems. Fundamenta Informaticae 50 (2002)

[5] H.C.M.Kleijn and M.Koutny: Process Semantics of General Inhibitor Nets, Information and Computation, Volume 190, Issue 1, pp 18-69 Elsevier Science Publishers BV, 2004.

[6] V.Khomenko, M.Koutny and W.Vogler: Canonical Prefixes of Petri Net Unfoldings, Acta Informatica, Volume 40, pp 95-118 Springer-Verlag, 2003.

[7] V.Khomenko, M. Koutny, M. and A. Yakovlev. Detecting State Coding Conflicts in STG Unfoldings Using SAT, Fundamenta Informaticae, 62 (2), pp 221-241 IOS Press, 2004. (Spec. Issue on Best Papers from ACSD2003).

[8] J. Burton, M. Koutny, and G. Pappalardo. Relating Communicating Processes with Different Interfaces, Fundamenta Informaticae, Volume 59, Issue 1, pp 1-37 IOS Press, 2004

[9] R.Devillers, H.Klaudel, M.Koutny, and F.Pommereau, Asynchronous Box Calculus, Fundamenta Informaticae, Volume 54, Issue 4, pp 295-344

[10] D.Sokolov and A.Yakovlev, Clock-less circuits and system synthesis, IEE Proceedings, Computers and Digital Techniques, Vol. 151, No.5, November 2004 (to appear).

[11] F. Burns, D. Shang, A.M. Koelmans and A. Yakovlev. Scheduling and allocation using closeness tables, IEE Proceedings-CDT, Vol. 151, No.5, pp. 332-340, September 2004.

[12] A. Yakovlev, S. Furber, R. Krenz, A. Bystrov. Design and Analysis of a Self-timed Duplex Communication System, IEEE Transactions on Computers, Vol. 53, No.7, pp. 798-814, July 2004.

[13] D. Shang, F.Burns, A.Koelmans, A.Yakovlev, F. Xia. Asynchronous system synthesis based on direct mapping using VHDL and Petri nets, IEE Proceedings-CDT, Vol. 151, No.3, May 2004, pp. 209-220.

[14] H. Saito, A. Kondratyev, J. Cortadella, L. Lavagno, T. Nanya and A. Yakovlev. Design of asynchronous controllers with delay insensitive interface, IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences, Vol. E85-A(12): 2577-2585, December 2002.

[15] J. Carmona, J.Cortadella, V. Khomenko and A.Yakovlev. Synthesis of Asynchronous Hardware from Petri Nets, Lectures on Concurrency and Petri Nets: Advances in Petri Nets, LNCS 3098, 2004, (Eds. J. Desel, W. Reisig, and G. Rozenberg) Publisher: Springer-Verlag Heidelberg, ISBN: 3-540-22261-8, pp 345-401

[16] A. Yakovlev, F. Xia. Towards synthesis of asynchronous communication algorithms, 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.

[17] E.Best and M.Koutny. Process Algebra: A Petri-Net-Oriented Tutorial, Lectures on Concurrency and Petri Nets: Advances in Petri Nets, Lecture Notes in Computer Science, Volume 3098, 2004, (Eds. J. Desel, W. Reisig, and G. Rozenberg) Publisher: Springer-Verlag Heidelberg, ISBN: 3-540-22261-8, pp. 180-209.

[18] J.Burton, M.Koutny, G.Pappalardo, and M.Pietkiewicz-Koutny. Compositional Development In the Event of Interface Difference, In: Concurrency in Dependable Computing, Ezhilchelvan, P.and Romanovsky, A. (eds.), pp. 3-22, Kluwer Academic Publishers, Boston ISBN: 1-4020-7043-8, 2002

[19] M. Renaudin and A. Yakovlev. From Hardware Processes to Asynchronous Circuits via Petri Nets: an Application to Arbiter Design, Proc. Workshop on Token Based Computing (ToBaCo2004), sattelite to 25th Int. Conf. on Appl. and Theory of Petri nets, Bologna, Italy, 22 June 2004, pp. 59-66.

[20] V. Khomenko, M. Koutny and A. Yakovlev. Logic synthesis for asynchronous circuits based on Petri net unfoldings and incremental SAT, Proc. ACSD 2004, Edited by M. Kishinevsky and Ph. Darondeau, Hamilton, Canada, 16-18 June 2004, IEEE Press, pp. 16-25. (Best paper award of ACSD 2004).

[21] R.Devillers, H.Klaudel, and M.Koutny. Petri Net Semantics of the Finite pi-Calculus, Proc. FORTE 2004, IFIP Conf., Madrid Spain, Sept. 27-30, de Frutos-Escrig, D. and Nunez, M. (eds), LNCS 3235, pp. 309-325, 2004.

[22] R.Devillers, H.Klaudel, and M.Koutny. Context-Based Process Algebras for Mobility, Proc. ACSD 2004, Edited by M. Kishinevsky and Ph. Darondeau, Hamilton, Canada, 16-18 June 2004, IEEE Press, pp. 79-88.

[23] D. Shang, F. Burns, A. Koelmans, and A. Yakovlev. A Balanced Power Implementation of the AES Security Algorithm using Self-timed circuits, Technical Track Proceedings of the 2nd International Conference of Applied Cryptography and Network Security (ACNS 2004), pp. 84-93, June 8-11, 2004, Yellow Mountain, P.R. China.

[24] D. Shang, F. Burns, A. Bystrov, A. Koelmans, D.Sokolov and A.Yakovlev. A low and balanced power implementation of the AES security mechanisms using self-timed circuits, PATMOS 2004, Santorini, September 2004, LNCS 3254, pp. 471-480.

[25] F. Burns, D.Shang, A.Koelmans and A.Yakovlev. An Asynchronous Synthesis Toolset Using Verilog, Proc. Design and Test Europe (DATE'04), Paris, Feb. 2004, IEEE CS Press pp.724-725.

[26] A. Bystrov, D. Sokolov, and A. Yakovlev. Low latency control structures with slack, Proc. of ASYNC'03, May 2003, Vancouver, IEEE CS Press, pp. 164-173.

[27] N. Starodoubtsev, S. Bystrov, and A. Yakovlev. Monotonic circuits with complete acknowledgement, Proc. of ASYNC'03, Vancouver, IEEE CS Press, pp. 98-108.

[28] D. Sokolov, A. Bystrov and A. Yakovlev. STG optimisation in the direct mapping of asynchronous circuits, Proc. DATE 2003, Munich, March 2003, pp. 932-937.

[29] D. Sokolov, A. Bystrov, A. Yakovlev, Tools for STG Optimisation in the Direct Mapping of Asynchronous Circuits, Second UK ACM SIGDA Workshop on EDA, Bournemouth, 16-17 Sept. 2002.

[30] F. Burns, D. Shang, A. Koelmans, A. Yakovlev, Using Petri Nets for Asynchronous Data Path and Controller Synthesis, Second ACM SIGDA UK Workshop on EDA, Bournemouth, 16-17 Sept. 2002.

[31] A. Bystrov, A. Yakovlev, Synthesis of Asynchronous Circuits with Predictable Latency, IEEE/ACM 11th International Workshop on Logic and Synthesis, New Orleans, June 2002, IEEE Computer Society and ACM SIGDA, pp. 239-243.

[32] 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.

[33] 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.

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

[35] V.Khomenko, M.Koutny, and A.Yakovlev: Detecting State Coding Conflicts in STGs Using Integer Programming. Proc. of DATE'2002, Kloos C.D., Franca J. (Eds.). IEEE Computing Society (2002)

[36] H.C.M.Kleijn and M.Koutny: Causality Semantics of Petri Nets with Weighted Inhibitor Arcs. Proc. of CONCUR'02, L. Brim, P. Jancar, M.Ketinsky, A. Kucera (Eds.), LNCS 2421 (2002).

[37] R.Devillers, H. Klaudel, M.Koutny, and F.Pommereau. An Algebra of Non-safe Petri Boxes, Proc. of AMAST 2002, Reunion Island, France, Sept. 2002, Kirchner, H. and Ringeissen, C. (eds.) LNCS 2422, pp. 192-207.

[38] V. Khomenko and M. Koutny: Branching Processes of High-Level Petri Nets. Proc. of TACAS'03, Garavel, H., Hatcliff, J. (Eds.). Springer-Verlag, LNCS 2619 (2003)

[39] V. Khomenko, M. Koutny, and A. Yakovlev: Detecting State Coding Conflicts in STG Unfoldings SAT. Proc. of ICACSD'2003, Balarin, F., Lilius, J. (Eds.). IEEE Computing Society (2003)

[40] A.Madalinski: ConfRes: Interactive Coding Conflict Resolver based on Core Visualisation. Proc. ICACSD'2003, Balarin, F., Lilius, J. (Eds.). IEEE Computing Society (2003)

[41] D. Shang, F. Burns, A. Koelmans, and A. Yakovlev. An Asynchronous DMA System Design Based on Direct mapping using VHDL and Petri nets, Proc. of 13th UK Async. Forum, University of Cambridge, Dec 17-18, 2002

[42] F. Burns, D. Shang, A. Koelmans, and A. Yakovlev. Synthesis of Asynchronous Data Paths and Controllers using PNs, Proc. of 12th UK Async. Forum, London South Bank University, June 17-18, 2002

[43] F. Burns, D. Shang, A. Koelmans, and A. Yakovlev. Translating from Asynchronous FSM Specifications in VHDL to Petri nets, Proc. of 11th UK Async. Forum, University of Manchester, Dec 17-18, 2001.

[44] S.Tyerman. Control Data Flow Graphs and High Level Synthesis of Digital Circuits, Proc. of 12th UK Async. Forum, London South Bank University, June 17-18, 2002

[45] V.Khomenko: Model Checking Based on Prefixes of Petri Net Unfoldings, PhD Thesis, School of Computing Science, University of Newcastle upon Tyne (2003).

[46] D.Shang:Asynchronous Communication Circuits: Design, Test and Synthesis. PhD Thesis. School of EECE, University of Newcastle upon Tyne (2003) (Tech Report: NCL-EECE-MSD-TR-2003-100, http://async.org.uk)

[47] F. Burns, D. Shang, A. Koelmans, and A. Yakovlev. VERISYN: Tool suport for Verilog Asynchronous Synthesis - A Tutorial Guide, Technical report, School of EECE, Univ. of Newcastle (to be published soon).

[48] V.Khomenko, A.Madalinski, and A.Yakovlev. Resolution of Encoding Conflicts by Signal Insertion and Concurrency Reduction Based on STG Unfoldings, CS-TR: 858, School of Computing Science, Univ. of Newcastle, Sept. 2004.

[49] E.Best, J.Esparza,K. Schmidt and H.Wimmel. Separability in Conflict-free Petri nets, Technical report, Carl von Ossietzky University, Oldenburg, April 2004.

[50] J. Cortadella, J. Garside and A. Yakovlev: Logic Design of Asynchronous Circuits, ASPDAC/VLSI Design 2002, Bangalore, Jan. 2002.