A Simple Approach for the Design of Asynchronous State Machines Using One-Hot Encoding

Duarte L. Oliveira, Tiago Curtinhas, Lester A. Faria
Divisão de Engenharia Eletrônica – Instituto Tecnológico de Aeronáutica – IEEA – SJC – SP – Brazil

Abstract — **This paper proposes a new approach for the synthesis, by direct mapping, of burst-mode asynchronous finite state machines (BM_AFSM). Considering typical aerospace environments, asynchronous circuits tend to avoid several kinds of problems related to high-complexity circuits designing, what is highly desirable in military systems. They do not present clock skew and minimizes the power consumption, leading to long-life battery-fed systems. They are also more robust to noise, temperature variations and to technology migration, presenting low electromagnetic emission. The great advantage of the synthesis by direct mapping is the simplicity of the methodology, not demanding any knowledge on asynchronous logic concepts, hazard-free circuits and critical race theory. The synthesized BM_AFSMs operate in Ib/Ob mode, which present superior performance when compared to generalized fundamental mode asynchronous circuits. It shows to be faster than those previously proposed in the literature, presenting a good modularity, what is provided by the simple interface with the external world. Starting from a well-known burst-mode (BM) specification, the direct mapping allows implementing large specifications with just little computational effort. The new approach has, as main advantage when compared to other direct mapping methods, the reduction of the memory elements and thus, of the circuit area. For a large set of benchmarks specified in burst mode, it obtained an average reduction of 49% of the memory devices.**

***Key-words* — control cells, finite state machine, asynchronous logic, direct mapping, BM specification**

I. INTRODUCTION

A typical aerospace environment demands highly integrated systems, where some design requisites show to be of major importance. In this context, asynchronous circuits tend to avoid several kinds of problems, being highly desirable for military applications. They do not present clock skew and minimizes the power consumption, leading to long-life battery-fed systems. They are also more robust to noise, temperature variations and to technology migration, presenting low electromagnetic emission.

Duarte L. Oliveira, duarte@ita.br, Tel +55-12-3947-6813, Fax +55-12-3947-6930; Tiago Curtinhas, thiagohd@ita.br; Lester A. Faria, lester@ita.br

In this scenario, the application of VLSI (Very Large Scale Integration) circuits has grown quickly. This growth is only possible from reducing transistor dimensions (MOS – Deep-Sub Micron – MOS-DSM), increasing the density and number of transistors, and increasing the operation frequency. DSM-MOS technology needs to operate with low noise and low power [1].

Synchronous digital systems use a global clock to synchronize their operations and are quite popular due to simplicity of design. There is also an abundant offer of commercial CAD tools for automatic synthesis. But, a serious problem in the MOS-DSM technology is the consequences of the global clock signal, once it shows to be a major cause of noise, presents high electromagnetic emission, consumes a significant portion of power and shows an increasing complexity to define its distribution among the IC (leading to clock skew). Timing analysis of high integration synchronous digital circuits (MOS-DSM) is extremely difficult. In a digital system, the sequential part is the main contributor to dynamic power dissipation [1,2]. Recent studies have shown that, in such systems, clock consumes a large percentage (15% to 45%) of the power of the system [3]. An interesting alternative to digital design, once it eliminates the clock related problems, is the *asynchronous paradigm* [4].

Asynchronous circuits have several advantages when compared to their synchronous counterparts: they do not present clock skew and minimizes the power consumption. Asynchronous circuits are more robust to noise, temperature variations, technology migration, and low electromagnetic emission [5]. Although that, the main disadvantage of asynchronous circuits is the intrinsic difficulty for designing, considering hazards-free and critical races circuits [5,6]. There have been, in the last years, several attempts to demonstrate the potential advantages of the asynchronous circuits over its synchronous counterpart [7]–[11]. In these asynchronous systems, the asynchronous controller is an important component.

A. *Classes of asynchronous controllers*

Asynchronous controllers (AC) may be classified according to three different kinds of delay models, where

"Bounded delay" relates to a well-known range of delays, while "unbounded delay" relates to finite but unknown ones [6]. These AC can be classified as:

- Speed independent (**SI**) circuits - They operate according to the unbounded gate and zero wire delay model. **SI** circuits are synthesized from a signal transition graph (STG), a Petri-net like specification [12]. Restrictions must be imposed to the circuits, in order to incorporate a "free of hazards" behavior in such descriptions. STG specifications are not familiar to VLSI designers, tending to be confused in large specifications. Also, the description of a conditional signal "by level" is not natural (level sensitive signal).
- Delay insensitive (**DI)** circuits - They operate according to the unbounded gate and wire delay model. **DI** circuits are synthesized from a concurrent language specification [13]. The resulting circuits are netlists of complex gates, tending to be large and slow (poor optimization) [13]. A relaxation in the model leads to quasi-delay insensitive circuits (QDI), where the wires that present forks must satisfy the concept of isochronic fork [6]; and
- Asynchronous finite state machines (**AFSMs**) - Originally proposed by Huffman [13] and extended as burst mode asynchronous finite state machines in [15,16], they operate according to the bounded gate and wire delay model. Their synthesis starts from a burst mode (BM) specification, as proposed by Nowick [15]. The BM specification is very familiar to VLSI designers. Later, it was generalized to the, so called, extended burst-mode specification (**XBM_AFSM**), as proposed by Yun [17,18], and then to the multi-burst graph specification (**MBG_AFSMs**), as proposed by Oliveira et al. [18]. X**BM_AFSMs** operate in the generalized fundamental mode (GFM). A new transition may start only when the circuit reaches a stable state. Input and/or output bursts are allowed. The XBM_AFSMs presents good performance, especially for those applications that interact with a slow environment. However, they are not modular and present timing and interface problems [6,17].

B. Asynchronous controllers (AC): approaches

The two main approaches for the designing of asynchronous controller are: logic synthesis [15]-[19] and direct mapping [20]-[27].

Logic synthesis works with the low-level specifications, which captures the behavior of a circuit at signal transition level. In this approach, the main goal is to obtain the minimized Boolean equations of the output and of the next state signals. This optimization requires an exploration of all possible events of the specification. The logic synthesis methods are well established and supported by automatic synthesis tools, as Petrify [28], Minimalist [29], 3D [18] and SAGAAS [30]. However, it suffers from excessive computational complexity and memory requirements. These limitations make it difficult to be applied to large specifications. For STG specification, there is a variant method that reduces the computational effort of the synthesis, called direct synthesis [31,32].

The 3D and SAGAAS tools synthesize BM_AFSMs and XBM_AFSMs. Minimalist tool only summarizes BM_AFSMs. The three tools are able to create circuits that operate in GFM mode and follow the conventional logic synthesis steps, but using requirements of asynchronous logic: i) state minimization step cannot allow merging of states, which can lead to logical hazard [14]; ii) the step of states assignment should perform the critical race free encoding [5.14]; iii) logic minimization step should extract a logical hazard free coverage [14.33]; and iv) the technological mapping step should perform a decomposition, which does not introduce logical hazard [34].

In *Direct Mapping* approach, the graph specification of a system is translated to a circuit netlist. Each node of the graph corresponds to a circuit element, while the arcs between these elements represent the interconnections of the circuit. Different methods for direct mapping have been proposed. In [20] and [21] the AFSM was adopted, with the limitation of the GFM assumptions. In [22], STG specification was adopted. In [23], the method starts from the FSM specification in synchronous style. Oliveira et al. [24] proposed a method for XBM_AFSMs synthesis by direct mapping that operates in the input burst/ output burst (I_b/O_b) mode, obeying the bounded gate and wire delay model. While the I/O mode accepts new inputs when one of the output signals is changing (concurrency of inputs and outputs), in the (I_b/O_b) mode a new input burst is only accepted when all the output signals have changed their value. The direct mapping approach in [24] starts from the extended burst mode specification and produces XBM_AFSMs, which are more modular, presenting a simpler interface. The great advantage of the method is the simplicity of the procedure. It does not require previous knowledge of the "hazard-free" and "critical race" circuit theory. Another advantage of the method is that it allows the implementation of large specifications with just a little computational effort. In the direct mapping approach, each state of the XBM specification requires a memory element (control cell – to see Fig.1) [24].

Other proposals were also presented, which aims to XBM specification and direct mapping synthesis. In [25] a proposal based on rules to optimize the XBM specification can be found, consequently reducing memory elements. References [26] and [27], respectively, proposed a new memory element: the first with smaller area and the second allowing a smaller latency, both of results compared to the memory element of [24].

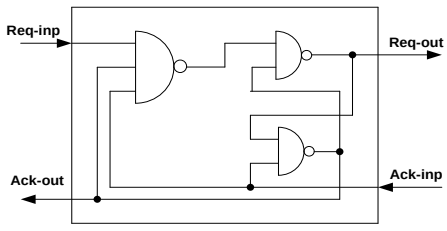

Fig. 1. Memory element: synthesis by direct mapping of [24].

Despite the many advantages previously cited for the synthesis of XBM controllers by direct mapping [23]-[27], a major drawback is the potential explosion of the memory elements, for large specifications. The reason is that each state of XBM specification is associated to one memory element, as shown in Fig. 1. Another disadvantage is the increased latency time when compared to the target architecture used in logic synthesis approach.

In this article we present a direct mapping method to design BM_AFSMs that operate in the Ib/Ob mode and obeys the bounded gate and wire delay model. Unlike the methods proposed in [23]-[27], this proposal starts from the BM specification and primarily performs the states minimization, free of logical hazard. The introduction of this step in the procedure reduces the number of states and consequently reduces the number of memory elements, used for direct mapping. The direct mapping procedure encodes the minimized BM specification in one-hot style, allowing flexibility in the synthesis method. The proposed method achieves best optimization results when compared to the optimization rules of [25], being also able to reduce the latency time for the states that are merged. Figure 2 shows the target architecture used in the proposed method. Figure 3 shows the used cell to implement each state of the minimized BM specification, coded as one-hot and Fig. 4 shows the latch RS used as memory element of the cell.

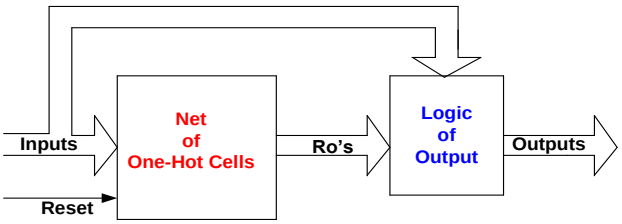

Fig. 2. Target asynchronous architeture.

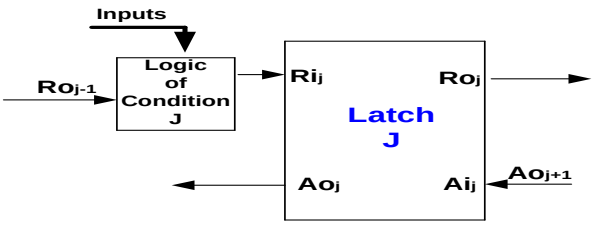

Fig. 3. Cell structure for each state of [24].

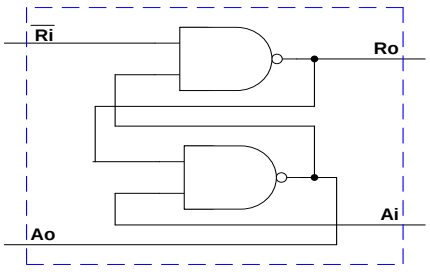

Fig. 4.Memory element of [27].

II. BM SPECIFICATION: ASYNCHRONOUS FSMS

In this paradigm, Nowick [14] proposed a specification denominated Burst-Mode (BM). Transitions may occur when one or multiple inputs/outputs (bursts) change their logic level, *0→1*, or *1→0* (transition sensitive signals – TSS). When there is no input change, the machine remains in its stable state. The input/output bursts must be monotonic, i.e., they can change only once during each transition. An initial state must exist. In BM specification, the state transition is labeled with input burst/output burst, where output burst may be empty and only the labeled signals in the state transition can change.

The BM specification must satisfy three properties to be considered for implementation [14]: i) polarity of the signals, which is switching a signal transition {+,–}; ii) unique input point; and iii) maximal set. The polarity property defines the condition of stable state flow. This property eliminates the oscillation of states and creates the concept of stable and unstable states. It facilitates the encoding of states free of critical race. The unique input point property allows, in the step of logic minimization, a logical hazard-free coverage. For an asynchronous FSM having a deterministic behavior, the branches (decisions) of BM specification must satisfy the property maximal set.

Figure 5 shows a BM specification of the HP-Pe-send-ifc asynchronous controller. The input signals are: xsend, TReq, yIQ, ADBIdout and AckPkt. The output signals are: TAck, PEAck and ADBId. One good example is the arc labeled with [TReq+ yIQ+ ADBIdout- / ADBId+] which means that if TReq+ and yIQ+ inputs changes from 0 to 1 and ADBIdout input change from 1 to 0, the output ADBId will change from 1 to 0. The state 0 is the initial state.

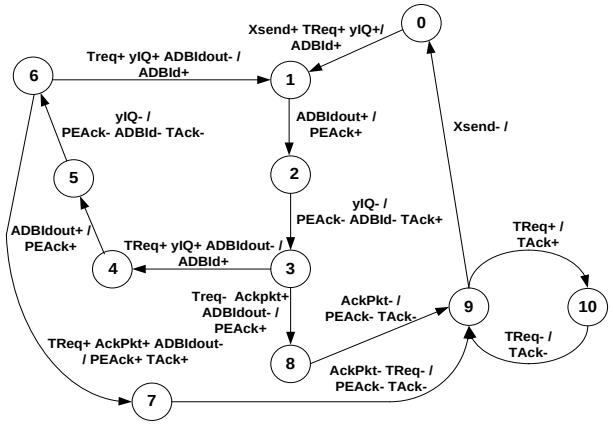

Fig. 5. BM Specification: HP-Pe-send-ifc controller

III. PROCEDURE SYNTHESIS: BM-AFSMS

The proposed method synthesizes BM-AFSMs and starts from a BM description. The method consists of four steps:

1. Perform the state minimization of the BM-AFSM, using the Minimalist [29] or SAGAAS [30] tools, generating the BM minimized flow table (BM-FT_{MIN}).
2. From the BM-FT_{MIN}, generate the BM-$FT_{MIN\text{-}COD}$, using the one-hot encoding.
3. Extract the conditional logic equations for each state, directly from the BM-FTMIN_COD, and perform the connection between cells, according to the method of [24] (see Fig. 3).
4. Extract the two-level output equations, type sum of product, using the ESPRESSO tool of SIS [36] tool. The output equations are generated from Ro's signals and from the input signals.

Figures 2, 3 and 4 describe the target architecture for the proposed method. In the methods [24],[26],[27], the output equations were generated only with the Ro's signals. In the new method, the output equations are generated with the Ro's signals and with the input signals.

IV. STUDY CASE

To illustrate the proposed architecture and its robustness in relation to the requirements of the asynchronous logic, it was applied to a well-known benchmark called dme-fast-e, which is described in Fig. 6. Following the synthesis method, shown in Section III, Fig. 7 shows the minimized XBM flow table, obtained by the STAMINA tool (step 1), where the following merging were obtained: A=(0,1,3), B=(4,5,6) and C=(2,7). Step 2 generates the one-hot encoding of the minimized XBM flow table, as shown in Fig. 8. The third step accomplishes the network construction of one-hot cells, where the conditional logic is obtained and the connections of the "request" and "acknowledge" signals are made according to the method of [24],[26],[27] (see Fig. 9 and 10). Finally, step 4 performs the logic minimization, using ESPRESSO tool, obtaining the output signals equations (to see Fig.11).

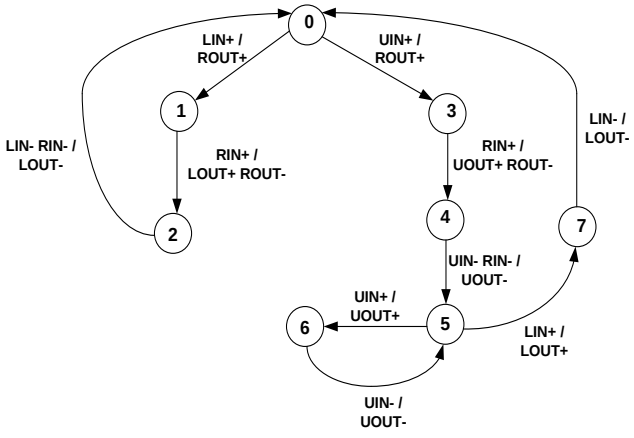

Fig. 6. BM Specification: dme-fast-e controller

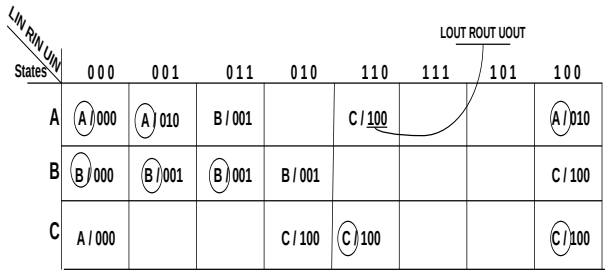

Fig. 7. Dme-fast-e: BM flow table minimized by STAMINA [34].

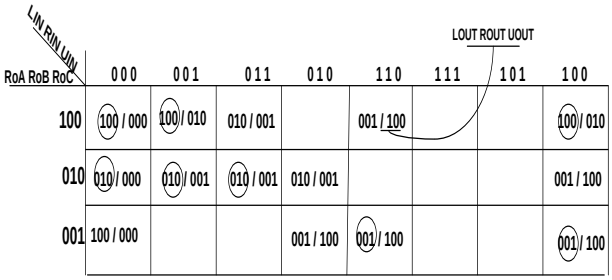

Fig. 8. Dme-fast-e: BM flow table encoded by one-hot.

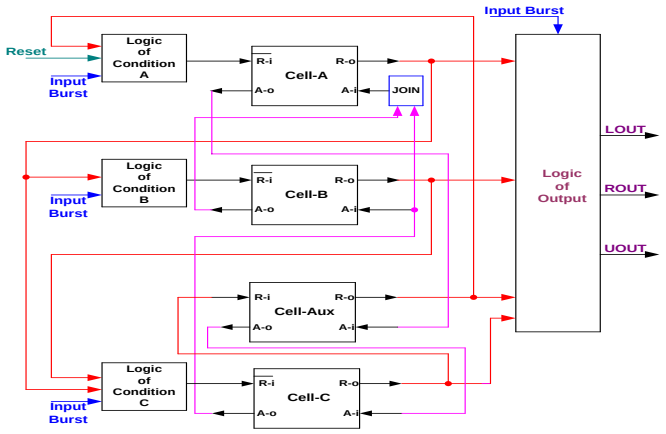

Fig. 9. Dme-fast-e:Net of control cells →connections (request and acknowledge).

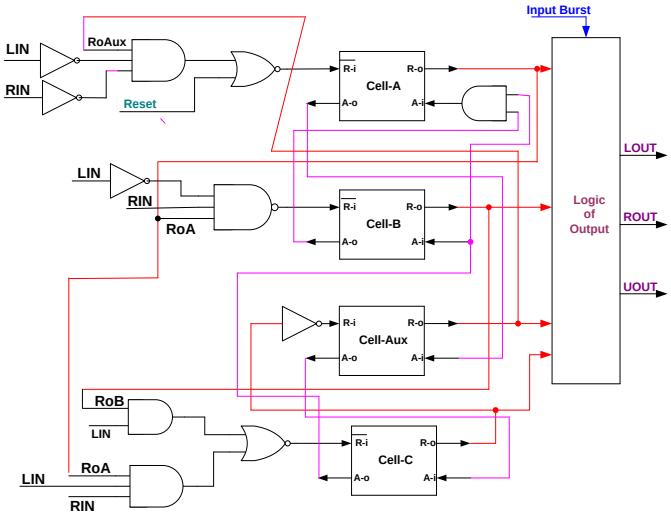

Fig. 10. Dme-fast-e: Net of control cells and logic of condition.

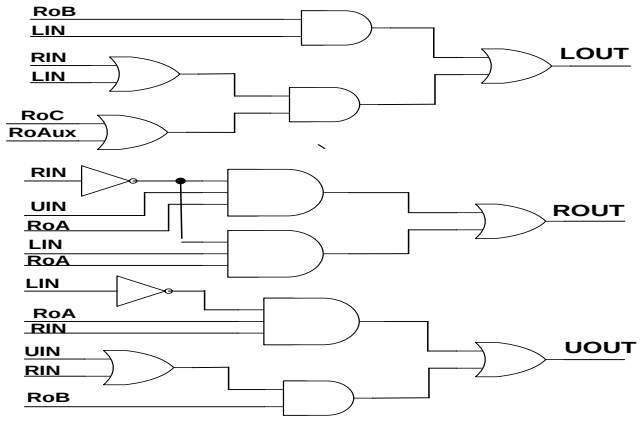

Fig. 11. Dme-fast-e: logic circuits of output signals.

V. DISCUSSION & SIMULATION

FSMs are important components in a digital system, being used in different applications, such as in aerospace [37], where robustness to radiation effects and many others are of major importance [38]. In order to demonstrate the feasibility of our proposed method, a well-known benchmark called Dme-fast-e was simulated in Altera QUARTUS II, version 9.1, target family CYCLONE III, device EP3C25F324C6 [39]. Figure 12 shows simulation of the Dme-fast-e controller, which was synthesized by direct mapping method. The Dme-fast-e needed 18 LUTs and presented a latency time of 8ns.

Table I shows BM specifications of 33 benchmarks, where *in/out/st/tran* mean, respectively, the numbers of input signals, output signals, states and state transitions. Table I also shows the reduction of the number of final states, obtained by Minimalist tool, which also provides the state assignment (number of states signals – *mvs*). When compared with the number of initial states, the state minimization obtained an average reduction of 49%, therefore obtaining an 49% reduction in memory elements and it was not counted the number of auxiliary memory elements.

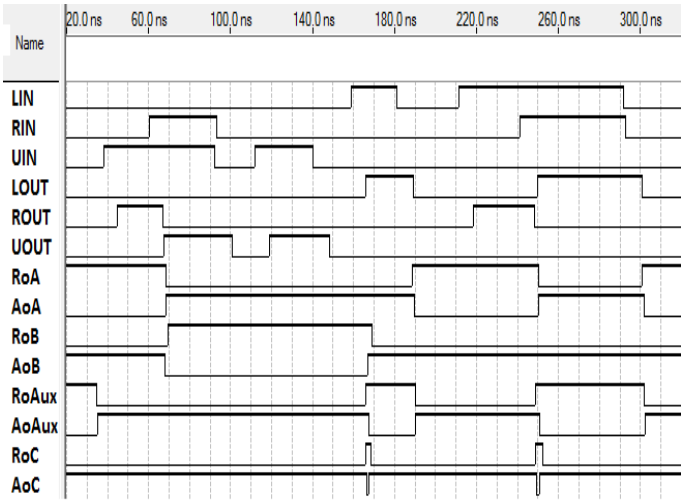

Fig. 12. Simulation: XBM-AFSM of Dme-fast-e by direct mapping.

TABLE I BM SPECIFICATIONS BY DIRECT MAPPING

Benchmark	Specification In	Out	St	Tr	MINIMALIST St	Mvs
alloc-outbound	4	3	8	9	5	3
call-proc	3	3	12	16	2	1
concur-mixer	3	3	5	6	3	2
counter	2	9	17	17	16	4
counter-bin	1	5	32	32	* 32	11
dme-e	3	3	8	10	3	2
dme-fast-e	3	3	8	10	5	3
hp-ir	3	2	6	8	2	1
hp-ir-rf-ctrl	6	5	12	13	6	3
hp-ir-sc-ctrl	13	14	33	42	13	4
isend	4	3	9	11	6	3
isend-bm	5	4	10	12	4	2
isend-csm	5	4	8	9	3	2
it-control	5	7	10	12	4	2
mp-f-p	3	4	4	4	2	1
nak-pa	4	5	6	6	2	1
nowick	3	2	6	6	2	1
opt-token-dist	4	4	12	12	6	3
pe-rcv-ifc	4	4	12	15	7	3
pe-send-ifc	5	3	11	14	5	3
pscsi	10	5	45	62	10	5
pscsi-tesend-bm	4	4	10	12	6	3
ptrcvb1	4	4	7	9	4	2
qr42	2	3	4	4	3	2
re-setup	3	2	6	7	2	1
ring-counter	1	2	8	8	8	1
sbuf-sct	3	3	8	9	4	2
scsi-tsend-bm	5	4	11	13	5	3
sdcont2	8	12	27	32	13	4
stetson-p2	8	12	25	28	13	4
stetson-p3	4	2	8	11	3	2
strcv-bm	5	4	10	12	4	2
yun-diffeq-alu1	3	5	7	9	5	3
Total			405		208	91

* O The benchmark *counter-bin* was synthesized by SAGAAS tool [30], where the Minimalist tool was unsuccessful.

VI. CONCLUSION

In this paper it was proposed a new approach to the synthesis of BM_AFSMs by direct mapping. The direct mapping approach presents several advantages if compared with the logic synthesis approach, like a very simple methodology and the non-necessity of previous knowledge about hazard-free circuits and critical race theory. This approach allows synthesizing large specifications with very little computational effort. The synthesized BM_AFSMs operate in mode I_b/O_b, which presents superior performance when compared to generalized fundamental mode asynchronous circuits, which are synthesized obeying the same delay model (bounded gate and wire delay). Theses circuits showed to be faster because new input bursts are processed sooner, when compared to those in the previous literature solutions (it is not necessary to wait for a stable state in order to accept a new input burst), while keeping a simple interface with the external world (good modularity).

The great advantage of the proposed method when compared to other methods using direct mapping is the large reduction of memory elements, and consequently minimizing area. For future work it is intended to adapt the new method for the extended burst-mode specification (XBM), whose highest difficulty shows to be the allowed level signal, which presents non-monotonic behaviour.

REFERENCES

[1] D. Goldhaber-Gordon, et al., "Overview of Nanoelectronic Devices," Proc. of the IEEE, vol. 85, No. 4, April 1997, pp.521-540.

[2] C. Constantinescu, "Trends and Challenges in VLSI Circuits Reliability," IEEE Micro, 23 (4), 2003.

[3] A. Jain et al., "A 1.2 GHz alpha microprocessor with 44.8 GB/s chip pin bandwidth," in IEEE Int. Solid-State Circuits Conf. Tech. Dig., pp. 240–241, February, 2001.

[4] I. E. Sutherland, and J. Ebergen, "Computers without clocks," Sci. Amer., pp. 62-69, August 2002.

[5] S. H. Unger, "Hazards, Critical Races, and Metastability", *IEEE Transaction on Computer*, June 1995, Vol. 44:6, pp. 754-768.

[6] C. J. Myers, "*Asynchronous Circuit Design*," Wiley & Sons, Inc., 2004, 2ª edition.

[7] S. M. Nowick et. al, "The Design of a High Performance Cache Controller: A Case Study in Asynchronous Synthesis" *Integration, the VLSI Journal*, Vol. 15, no 3, pp. 241-262, October 1993.

[8] K. Y. Yun e D. L. Dill, "A High-Performance Asynchronous SCSI Controller," Proc. Int. Conf. Computer Design (ICCD), pp. 44-49, 1995.

[9] K. Y. Yun, et al., "The design and verification of a high-performance low-control-overhead asynchronous differential equation solver," IEEE Transactions on VLSI Systems, vol. 6, no 4, pp.643-655, Dec.1998.

[10] S. Rotem, et al., "RAPPID: An asynchronous instruction length decoder,"in Proc. International Symposium on Advanced Research in Asynchronous Circuits and Systems, pp. 60-70, April, 1999.

[11] T. Chelcea, et al., "Burst-Mode Oriented Back-End for the Balsa Synthesis System," Proc. of DATE Conf. and Exhibition, 2002.

[12] T. -A. Chu, "Synthesis of Self-Timed VLSI Circuits from Graph-Theory Specifications," Ph.D. thesis, June, 1987, Dept. of EECS, MIT.

[13] J. Martin, "The Limitations to Delay Insensitive in Asynchronous Circuits," 6th MIT Conference on Advanced Research in VLSI Processes, pp.263-277, 1990.

[14] D. H. Huffman, "The Synthesis of Sequential Switching Circuits," *J. Franklin Ins.*, Vol. 257, pp. 161-190, March 1954 e pp. 275-303, April 1954.

[15] S. M. Nowick, "*Automatic Synthesis of Burst-Mode Asynchronous Controllers*," Ph.D. thesis, Stanford University, 1993.

[16] S. M. Nowick e Bill Coates, "UCLOCK: Automatic Design of High-Performance Unclocked State Machines,' Proc. ICCD, pp. 434-441, 1994.

[17] K. Y. Yun, "Synthesis of Asynchronous Controllers for Heterogeneous Systems", Ph.D. thesis, Stanford University, 1994.

[18] K. Y. Yun and D. L. Dill, "Automatic Synthesis of Extended Burst-Mode Circuits: Part I (Specification and Hazard-.Free Implementation) and Part II (Automatic Synthesis)," IEEE Trans. on CAD of Integrated Circuit and Systems, Vol. 18:2, pp. 101-132, Feb. 1999.

[19] D. L. Oliveira, et al., "Miriã: a CAD toll synthesize multi-burst controllers for heterogeneous systems," *Microelectronics Reliability*, 43 (2003) 209-213.

[20] R. David, " Modular design of asynchronous circuit defined by graphs," *IEEE Trans. Computer*, vol. C-26, no. 8, pp.727-737, August 1977.

[21] L. A. Hollaar, "Direct implementation of asynchronous control unit," *IEEE Trans. Computer*, vol.C-31, no. 12, pp.1133-1141, Dec., 1982.

[22] D. Sokolov, A. Bystrov and A. Yakovlev, "Direct mapping of low-latency asynchronous controllers from STGs," *IEEE Trans. CAD of Integration Circuits and Systems*, vol. 26, no. 6, June 2007.

[23] D. L. Oliveira, et al., "Design of Asynchronous Controllers by Direct Mapping and Synchronous Specification," 4th Workshop on Circuits and System Design, WCAS 2014, Aracaju, 2014.

[24] D. L. Oliveira, L. Shibuya, O. Saotome, "Synthesis by Direct Mapping of Asynchronous Controllers from Extended Burst-Mode Specification," XVI Workshop Iberchip, Foz de Iguaçu – Brazil, 2010.

[25] D. L. Oliveira, et al., "A new Memory Element for Synthesis by Direct Mapping of Asynchronous FSMs from XBM Specification," XIII SIGE, São José dos Campos – Brazil, 2011.

[26] D. L. Oliveira, et al., "Synthesis of Extended Burst-Mode Asynchronous Controllers by the Use of Optimized Direct Mapping," IEEE VI Andean Region International Conference, Cuenca. 2012.

[27] D. L. Oliveira, N. Alles, L. A. Faria, "Synthesis by direct mapping of asynchronous extended burst-mode controllers using RS latch," IEEE 3rd Latin American Symposium on Circuits and Systems (LASCAS), 2012, Playa del Carmen, 2012.

[28] J. Cortadella, et al., "Petrify: A tool for manipulating concurrent specifications and synthesis of asynchronous controllers," *IEICE Trans. Inf. Syst.*, vol.E80-D, no. 3, pp.315-325, March 1997.

[29] R. Fuhrer, et al., "Minimalist: An environment for the synthesis, verification and testability of burst-mode asynchronous machines," Columbia Univ., NY, Tech. Rep. TR-CUCS-020-99, July 1999.

[30] T. Curtinhas, et al., "A novel state assignment method for Extended Burst-Mode FSM design using Genetic Algorithm," 27th Symposium on Integrated Circuits and Systems Design, SBCCI, 2014.

[31] E. Pastor, et al., "Structural methods for the synthesis of speed-independent circuits," *IEEE Trans. Computer-Aided Design*, vol. 17, pp. 1108-1129, November 1998.

[32] J. Carmona, "Structural methods for the synthesis of well-formed concurrent specifications," PhD thesis, Software Dept., Universitat Politécnica de Catalunya, Barcelona, Spain, March 2004.

[33] S. M. Nowick e D. L. Dill, "Exact Two-Level Minimization of Hazard-Free Logic with Multiple-Input Changes," *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol. 14, no 8, August 1995.

[34] P. S. K. Siegel, "Automatic Technology Mapping Asynchronous Designs," PhD thesis, Stanford University, February, 1995.

[35] B. Lin and R. Newton, "Synthesis of multiple level logic from symbolic high-level description languages", *Proc. Int. Conf. on VLSI*, 187–206 (1989).

[36] E. Sentovich, et al., "SIS: System for Sequential Circuit Synthesis," Tech. Rep. M92/41, Electronic Research Laboratory, College Enginerring, University of California, Berkeley, 1992.

[37] L. Yuan, et al., "Research on the Problems of Satellite Borne FPGA Based Finite State Machine," 2nd Int. Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA), pp. 1-4, 2008.

[38] D. J. Barnhart, et. al. "Radiation Hardening by Design of Asynchronous Logic for Hostile Environments", IEEE Journal of Solid-State Circuits, vol. 44, No. 5, pp.1617-1628, May 2009.

[39] Altera Corporation, 2015, www.altera.com.

.