

Synthesis by Direct Mapping of Asynchronous Control Circuits from Bursts Transition Graph

Duarte L. Oliveira¹, Sandro S. Sato², Lester A. Faria¹

¹Divisão de Engenharia Eletrônica – Instituto Tecnológico de Aeronáutica – SJC – Brazil

²Departamento de Informática – EET Ferraz de Vasconcelos – SP – Brazil

Abstract— Several proposals have been made to generalize the specifications extended burst-mode (XBM) and signal transition graph (STG), which describe asynchronous controllers and that are popular. All proposals direct to methods for design of controllers by logic synthesis. These methods lead to a complex synthesis that can invalidate the automatic synthesis. In this paper we propose a method by direct mapping to synthesis of asynchronous controllers that are described by the bursts transition graph (BTG) specification. The BTG specification combines the strong of the specifications XBM and STG and the method by direct mapping has the advantages of simplicity, requires little computational effort thus allows synthesizing large specifications and requires no knowledge of asynchronous logic.

Keywords—Petri-net, XBM specification, control cell, signal transition graph

I. INTRODUCTION

In recent years, there has been a growing interest in asynchronous circuits due to the increase in performance and complexity of digital systems [1]. Asynchronous circuits present several potential advantages over their synchronous counterparts: they tend to be faster, dissipate less power, do not present clock skew nor clock distribution problems, they are more robust in respect to temperature variations and electromagnetic interactions [1].

Concerning to the asynchronous circuits, two remarkable specification styles have been proposed to describe the asynchronous controllers, which lead to an optimized synthesis:

1) The signals transition graph (STG,) proposed by Chu [2,3], is a subclass of sampled Petri nets, where each transition is sampled as a physical transition of some of the electrical signals. The STG focus primarily in the description of asynchronous circuits that communicate with external environment in input/output mode (M_I/O). In M_I/O, the occurrence of an output event immediately enables the occurrence of an input event. Different design styles: by logic synthesis [4,5] Synthesis by direct mapping from STG was proposed in [6,7,8] and by direct synthesis in [9,10,11]; and

2) Burst Mode (BM) is a kind of specification based on the state transition graph, which was firstly proposed by Davis [12] and later formalized by Nowick [13] and improved by Yun [14], as extended burst-mode (XBM), and by Oliveira et al. [15] as multi-burst graph (MBG). This mode allows multiple input changes, being used to describe asynchronous Mealy Finite State Machines (FSM). These machines interact with the environment in “general fundamental mode” (GFM), where a new burst input can only occur if the controller is already stabilized (no electrical activity in the ports and in the lines).

Limitations in STG and XBM specifications

An important and promising area, focusing on the implementation of asynchronous controllers shows to be the different kinds of heterogeneous systems (controllers and/or synchronous processors). In this kind of systems the signals may have non-monotonic behavior and the decisions can be made by “signal level” and not “by transition”. Despite of STG and XBM specifications are well accepted, the existent limitations are evident when it is necessary to describe asynchronous controllers focusing on heterogeneous systems:

1) The STG is a “per signal” description, having its activation per signal transition. Therefore: a) the STG is able to describe decisions with level sensitive signals (LSS) and non-monotonic behavior, however performing it in a confused and difficult way of description [16,17]; b) Despite one of the strengths of the STG shows to be its ability to describe sequences and concurrences between inputs and outputs, certain types of concurrence lead the STG to an excessive generation of transitions (explosion); c) many signals lead to a non-compact and hard to interpret specification; and d) describing signals with irrelevant behavior (directed don't-care) leads STG to a confused description [16,17]; and

2) The XBM is a kind of description based on “signals burst”, therefore some of the the strengths of the XBM are: a) to describe decisions with LSS signals, and non-monotonic behavior; b) to describe signals with irrelevant behavior (directed don't-care); and c) the description in “burst level” makes the specification more compact. On the other hand, the main weakness of XBM mode is describing with a limited

Duarte L. Oliveira, duarte@ita.br, Tel +55-12-3947-6813, Fax +55-12-3947-6930, Sandro S Sato, shoiti@uol.com.br, Lester A. Faria, lester@ita.br - ITA - IEAA - Praça Marechal Eduardo Gomes, 50, Vila das Acácias - São José dos Campos - SP - Brasil.

capacity the signal sequencing and input and output signals concurrence [18].

Generalizations in STG and XBM specifications

Due to limitations of the conventional STG, different generalizations were proposed for the GSTG [16,19,20]. These generalizations allow describing, in a natural way, the “directed don’t-care”, “toggle” and LSS signals. It is possible to find in literature some of these proposed generalizations as the ones in [19,20] and [21]. In [21], for example, the specification denominated Bursts Transition Graph (BTG) is based on Petri-net and incorporates the XBM specification. All of these proposals to GSTG synthesize the circuits in the logic synthesis style. In this style, the first step of the synthesis is to generate the state graph, which, in GSTG, besides showing to be very complex, easily explodes its size (becomes excessively large). Due to the limitations of XBM, Oliveira et al. [15] has proposed the MBG that incorporates the XBM, besides being able to describe various types of sequence and concurrence through the use of operators. Considering the MBG, there can be found methods based on logic synthesis [15] and direct mapping synthesis [22]. Despite of the efforts of several previous works found in literature to present specifications that can provide a greater descriptive capacity and greater simplicity for the synthesis method, some drawbacks still remain. As an example, although the MBG does not present synthesis problems (even large MBG can be performed by direct mapping), its ability in describing concurrence is limited. All existing proposals for generalized STG, for example the BTG, have as limitation a high difficulty, and sometimes the impossibility, of performing logic synthesis.

Based on the strengths and drawbacks previously commented, in this paper it is proposed a new method for direct mapping synthesis of asynchronous controllers that are described by BTG specification. Unlike the previous logic synthesis designs proposals concerning GSTG specifications, the direct mapping technique allows performing the synthesis without any previous knowledge of asynchronous logic and with just a little computational effort. These features enable the synthesis of a more compact specification that is simpler, with high descriptive capacity and easy to understand, which is called Burst Transition Graph (BTG). This specification incorporates both STG and XBM specifications.

II. BACKGROUND FOR THE SPECIFICATIONS

Extended burst-mode(XBM)

The XBM specification supports the BM specification, after introducing two kinds of input signals: a) LSS signals with non-monotonic behavior; and b) “directed don’t-care” signals, which can be activated concurrently with the output signals. It is possible to illustrate this paradigm with the benchmark Biu-fifo2dma of HP. Fig. 1 shows an XBM specification with 4 inputs (*cntgt1*, *dackn*, *fain*, *ok*), 2 outputs (*dreq*, *frou*) and an initial state 0. The description *fain-dackn+ / frou+* in transition 4→3 means that the output

(*frou*: 0→1) will follow the input burst (*fain*: 1→0 AND *dackn*: 0→1). The level sensitive signal (LSS) *cntgt1* is used to describe the mutual exclusion between transitions 2→5 and 2→4. The “directed don’t care” signal *fain** in transition 2→4 means that *fain* may either change its value or remain in its previous value. All state transition should have, at least, a signal denominated “compulsory”, where compulsory signal means an input signal that, in the previous state transition, is not “directed don’t-care”.

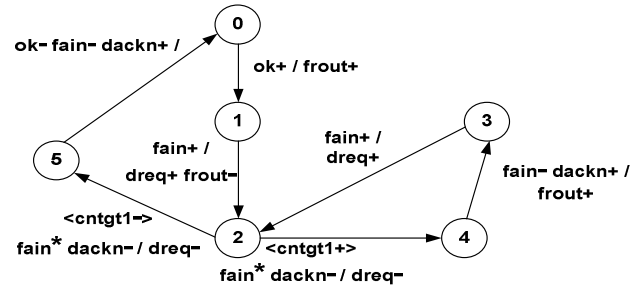


Fig. 1. Extended burst-mode specification of Biu-fifo2dma.

Signal transition graph (STG)

The STG proposed in [2,3] is a free-choice network, safe, free and “output-persistent”, with T transitions interpreted as signal transitions $S \times \{+, -\}$ where the initial label M_0 of the STG can have exactly one label in each state. The signal $s+$ indicates a transition from 0 → 1 and the signal $s-$ indicates a transition 1 → 0. The STG specification is used to describe various types of asynchronous circuits operating in M_I/O, for example, the quasi-delay-insensitive (QDI), which follows both unbounded gate model and wire delay model. In this model, there is a relaxation of the restriction fan-out > 1, once it allows isochronic fork (the different branches of the same connection have the same delay or, if there is any delay difference, it is negligible compared to the delay of the gates) [4, 5]; is speed-independent, SI (which follows the unbounded gate delay model [1]), and timed (which follows the bounded wire delay model [1]).

Fig. 2 illustrates a STG specification of the VME (Versa Module European) cycle-read of the VME-bus interface [5]. The input signals are: LDTACK and DSr. The output signals are LDS, D and DTACK.

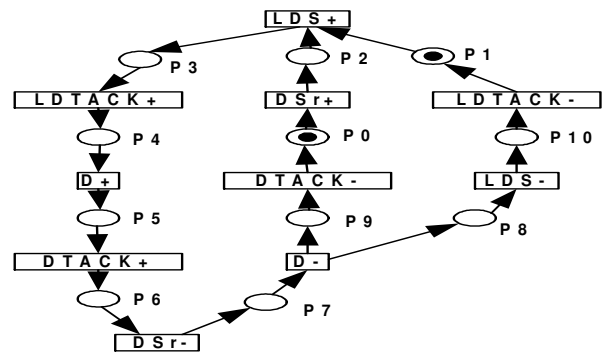


Fig. 2 Signal transition graph from the VME read cycle.

Generalized STG

An interesting generalization of STG was proposed by Peter et al. [20]. Fig. 3 shows the benchmark IC2 described in GSTG [20], where the SDA is a level signal, and, during the transition, it is possible to describe the activation of both input and output signals. As the main drawback of this kind of description there must be highlighted the difficulty of its logic synthesis.

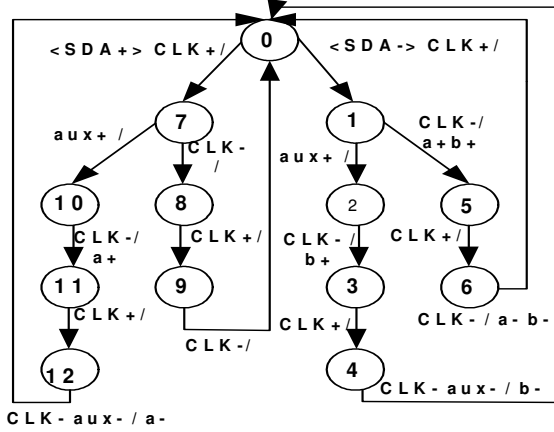


Fig. 3. Generalized signal transition graph of [20].

Explosion (excessive creation of states) in state graph

The high capacity in describing concurrence allows the conventional STG to describe signals with level sensitive behavior, but by the use of signals with transition sensitive behavior. Fig. 4 shows an example [23] of a conventional STG, where the signal "C" is TSS, but presents a LSS behavior. The commonly used procedure for controllers' design that starts from the STG and that uses the logic synthesis style generate a state graph as an intermediate structure. The conventional STG that presents TSS signals with LSS behavior tends to generate a very complex state graph and leads to an overgrowth in size (explosion). Fig. 5 shows the SG generated from STG of Fig. 4. Besides the overgrowth problem (explosion) of the SG, there must be highlighted a high difficulty in visualizing the behavior of LSS signals in the conventional STG, as is illustrated in Fig. 4.

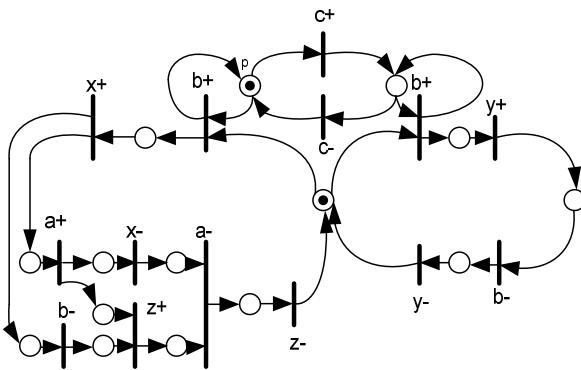


Fig. 4. STG with TSS signal but with behavior of LSS signal [23].

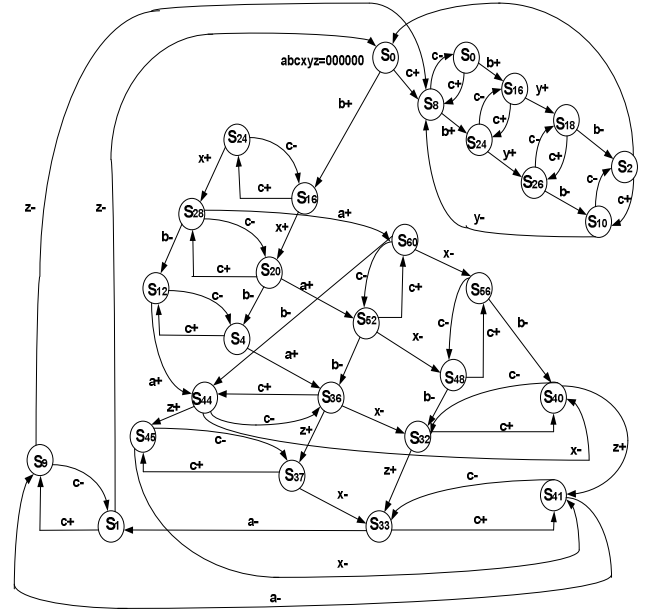


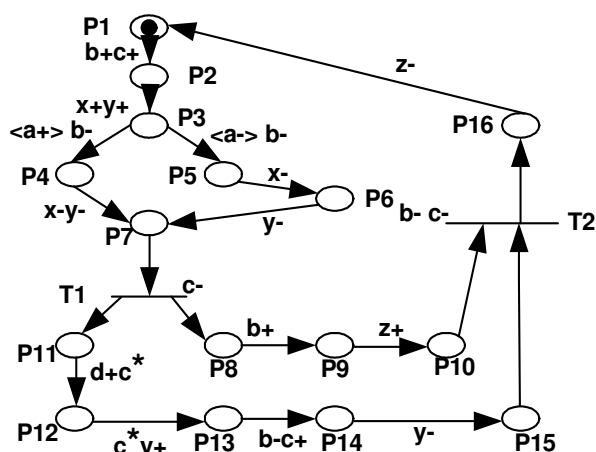
Fig. 5. State graph of STG in Fig. 4.

III. BURST TRANSITION GRAPH (BTG)

The burst transition graph (BTG) proposed in [21] is a subclass of sampled Petri nets, where each transition is sampled as a burst. The activation of the burst signals in the enabled transition can happen in any sequence or at any time. The possible signals are classified into three types: a) LSS signals with non-monotonic behavior; b) SST input signals with monotonic behavior, which are the terminal signals or irrelevant signals (directed don't-care); and c) SST output signals.

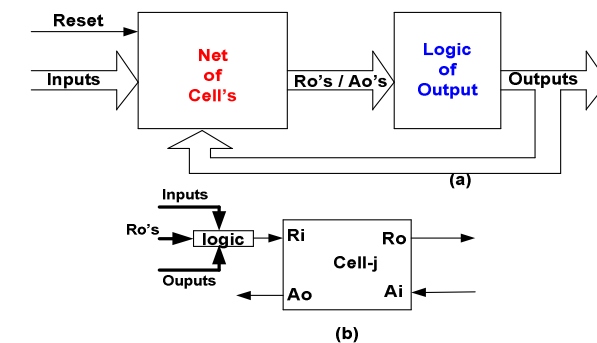
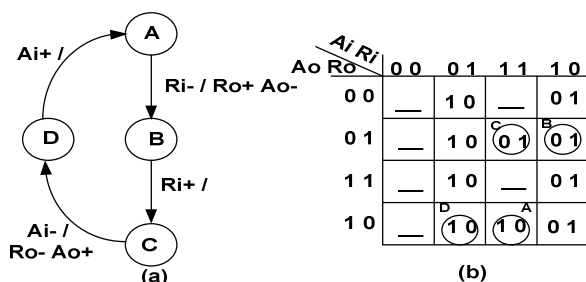
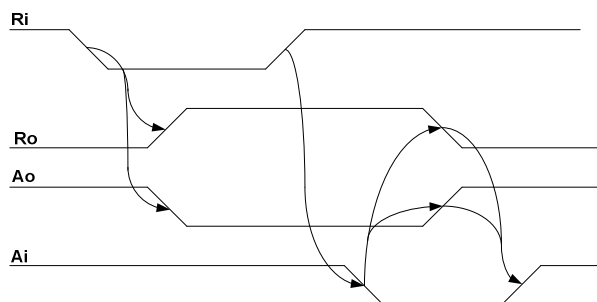
Definition 1. A burst transition graph (BTG) G is a 6-uple $G = \langle N, C, E, S, R, Mo \rangle$, where N is a free Petri net, safe, "persistent-output" and "free-choice"; C is a set of LSS signals with non-monotonic behavior; E is a set of SST input signals with monotonic behavior; S is a set of output signals; R is a burst function $R: T \rightarrow \{C x \{+, -\} \cup \{E x \{+, -, *\}\} \cup \{S x \{+, -\}\}$ labeled for each one of the transitions of N and Mo is the starting label.

The BTG obeys to the restrictions of the STG and XBM specifications [3,6]. Fig. 6 shows an example of BTG specification, where the TSS input signals are b , c and d . The LSS signal is a , and the output signals are x , y and z . For transitions with fan-in = 1 and fan-out = 1, the "bar" is omitted. In Fig. 6 the transitions (bar) $t1$ and $t2$ present respectively fan-in = 2 and fan-out = 2. In state transitions $P11 \rightarrow P12$, the signal c^* is "don't-care" and in transition $P13 \rightarrow P14$, the signal b^- is compulsory. The state $P1$ has the initial TOKEM.



V. SYNTHESIS PROCEDURE

1. Browse the complete BTG and point all the places like that: PI place, when there is, at least, one transition that arrives at this place that is activated by a burst input; and place as P0 when all transitions that arrive at this state are activated by a burst output.
2. Each input PI place of the BTG will be represented by a control cell. Paths with two PI places (scale two-loop) must have auxiliary control cells
3. Each control cell is associated to a logic block with output in **Ri**. Auxiliary control cells do not need the logic block.
4. Perform the connections **Ro** $\rightarrow$ *Logic_Block* for each PI state, where **Ro** is the initial PI place and logic block is the final PI place.
5. Perform connections **Ao** $\rightarrow$ **Ai** for each PI place in the opposite direction. When there are two or more connections arriving at **Ai**, it will generate a “joint”.
6. For each place *j* of the logic_block_{*j*}, extract the Boolean equation of type “sum-of-product”.
7. For each joint, replace it by a gate AND.
8. Using signals **Ro**’s, **Ao**’s and the P0 places, get the minimum Booleans equations of type “sum-of-product” of output signals.



VI. CASE STUDY

In this section it will be illustrated the proposed approach with the example of Fig. 11. This is the BTG description of the conventional STG, previously shown in Fig. 4. Fig. 12 shows the step #1. Once it is the path of two states ($p2 \rightarrow p4$), it must be inserted an auxiliary control cell between the states $p4 \rightarrow p2$ (step two – Fig. 13). Fig. 14 shows the general structure of controller. Fig. 15 shows the steps three and four, that are the insertion of connections for enable state and initialization. Fig. 16 and 17 shows the step five, which is the insertion of connections that enable and disable the states. Fig. 18 shows the step six, that is, get the Booleans equations of logic_block and finally, Fig. 19 shows steps seven and eight that are, get the output signals.

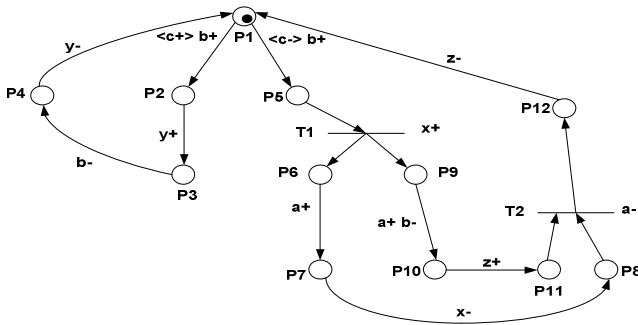


Fig. 11. BTG specification of Fig. 4.

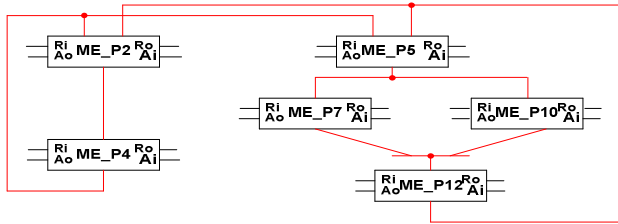


Fig. 12. Net of control cells: initial.

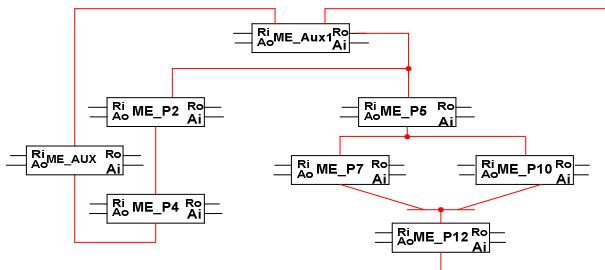


Fig. 13. Net of control cells: initial with ME aux.

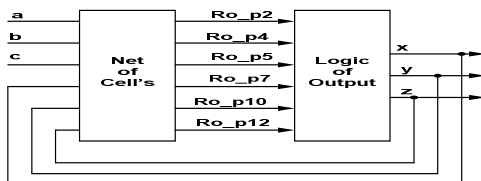


Fig. 14. General structural: net of cells + output logic.

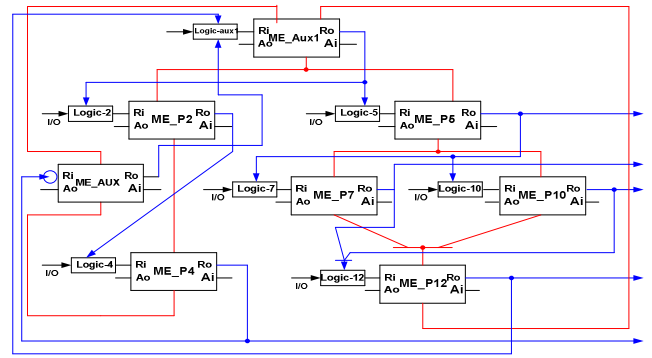


Fig. 15. Net of control cells: connections Ro.

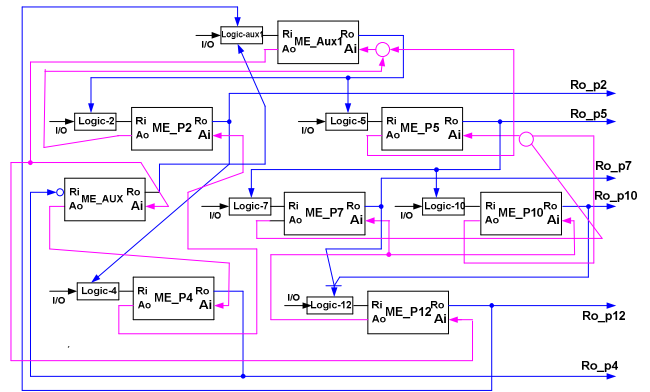


Fig. 16. Net of control cells: connections final.

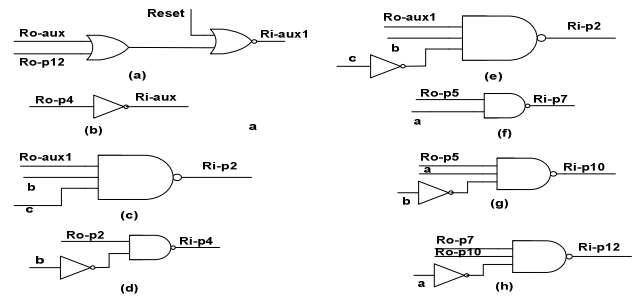


Fig. 17. Block_logic (circuits): a) logic-aux1; b) logic-aux; c) logic-2; d) logic-4; e) logic-2; f) logic-7; g) logic-10; h) logic-12

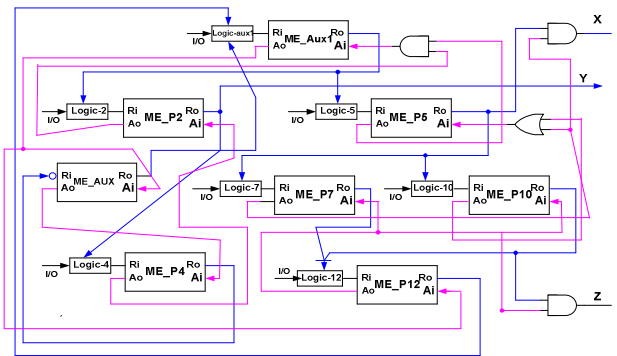


Fig. 18. Net of control cells: connections final and output logic.

VII. DISCUSSION & SIMULATION

The complex interfaces are characterized by the acceptance of LSS signals with not monotonic behavior, that appear in the heterogeneous systems (modules synchronous / asynchronous mixed) and may have a high concurrency. The specifications XBM and STG have difficulty describing these interfaces. The BTG specification describes naturally, because the BTG has the stronger of the two specifications.

The example used in the study case was simulated and compiled in the tool of ALTERA, software QUARTUS II, version 9.1, family CYCLONE II, and device EP2C8T14418. Fig. 19a and 19b show the hazard-free waveforms extracted from the simulation of the well-known benchmark output port controller, synthesized in the section VI of this work.

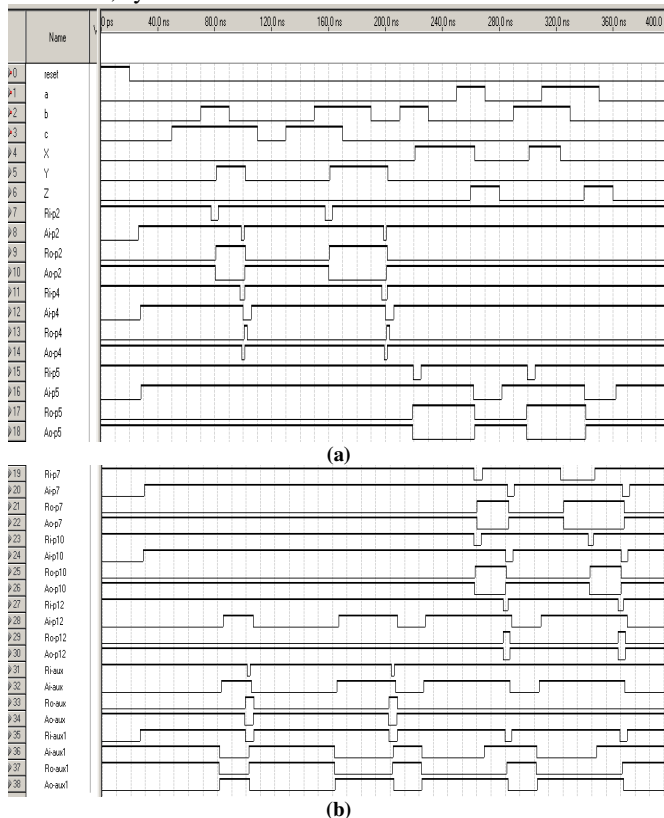


Fig. 19. Simulations of case study

VIII. CONCLUSION

In this paper we propose a method for direct mapping for BTG specification, where the control logic can easily be obtained. The BTG specification combines the strong of the specifications of asynchronous paradigm, which are STG and XBM. For future work, to develop a tool for automatic synthesis by direct mapping of the controllers described in BTG.

REFERENCES

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

[2] T. -A. Chu, "On the Models for Designing VLSI Asynchronous Digital Systems," *INTEGRATION, the VLSI Journal* 4, pp.99-113, 1986

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

[4] 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.

[5] Kondratyev, A., et. al., "Logic Decomposition of Speed-Independent Circuits," *Proc. of the IEEE*, vol. 87, NO. 2, February, 1999.

[6] Bystrov, A., et al., "Low-latency control structures with slack," *Proc. of the Ninth Int. Symposium on Asynchronous Circuits and Systems*, pp. 164-173, 2003.

[7] 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.

[8] D. Sokolov and A. Yakovlev, "Clockless Circuits and Systems Synthesis," *IEE Proceedings Comput. Digit. Tech.*, vol. 152, nro. 3, pp. 298-316, May 2005

[9] 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.

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

[11] J. Carmona, and J. Cortadella, "Encoding Large Asynchronous Controllers with ILP Techniques," *IEEE Trans. on CAD of Integrations Circuits and Systems*, vol. 27, nro. 1, pp. 20-33, Januray 2008.

[12] Davis, A. L., et al., "A data-driven machine architecture suitable for VLSI implementation," In C.L. Seitz, editor, *Proc. of the Caltech Conf. on Very Large Scale Integration*, pp.179-194, 1979.

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

[14] 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, February, pp. 101-132, 1999

[15] D. L. Oliveira, M. Strum, W. J. Chau, and W. C. Cunha, "Miriã: a CAD toll synthesize multi-burst controllers for heterogeneous systems," *Microelectronics Reliability*, 43, pp. 209-213, 2003.

[16] Yakovlev, A. V., "On Limitations and Extensions of STG Model for Designing Asynchronous Control Circuits," *Proc. ICCD*, pp. 396-400, 1992.

[17] Moon, C. W., "Synthesis and Verification of Asynchronous Circuits from Graph Specifications," PhD thesis, University of California, Berkeley, 1992.

[18] O. Kraus and M. Pedeffke, "XBM2PLA: A Flexible Synthesis Tool for Extended Burst Mode Machines," *Proc. Of the Design Automation and Test in Europe Conference and Exhibition*, , pp.1301-1303, 2003.

[19] Yakovlev, A., "Synthesis of Hazard-free Asynchronous Circuits from Generalized Signal Transition Graphs," 6th IEEE Int. Conf. on VLSI Design, pp.21-24, 1993.

[20] Vanbekbergen, V., et al., "A generalized signal transition graph model for specification of complex interfaces," *European Design and Test Conference, EDAC*, March, pp.378-384, 1994.

[21] D. L. Oliveira, "Grafo de Transição de Rajadas: Uma Especificação para Controladores Assíncronos de Interfaces Complexas," XI Workshop Iberchip, Salvador – Brazil, 2005.

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

[23] J. Beister, et al., "From STG to Extended-Burst-Mode Machines," *Proc. Fifth Int. Symposium on Advanced Research in Asynchronous Circuits and Systems*, pp.145-156, 1999.