Synthesis of Multi Burst-Mode gRS Machines from Flexible MBG Specification

Duarte Lopes de Oliveira • Marius Strum∗ Wang Jiang Chau∗
• Departamento de Eletrônica Aplicada do Instituto Tecnológico de Aeronáutica
duarte@ita.br
Praça Marechal Eduardo Gomes, 50 - CEP 12228-900 - São José dos Campos - São Paulo - Brazil
∗ Laboratório de Microeletrônica da Escola Politécnica da USP
strum@lme.usp.br jcwang@lme.usp.br
Av. Prof. Luciano Gualberto, Trav 3, 158 - CEP 05508-900 - São Paulo - SP - Brazil

Abstract
Asynchronous controllers specification used in heterogeneous (synchronous+asynchronous) systems rely on two types of signals: level sensitive signals (LSS) that are used as conditionals and transition sensitive signals (TSS). However several applications may contain signals that change from one type to the other during the controller's operation. We call such signals as "4 phase signals". Two previous publications showed how to handle these signals. One captures the signals using a generalized signal transition graph and then transforms it into a state graph. They target the I/O mode of operation. It is known that the use of state graphs limits the use of this method to small examples. That is not the case when non-monotonic LSS signals are present. A more recent work proposed to capture such 4PS signals as an extension of the extended burst-mode (XBM). The authors showed that the XBM2PLA tool was able to describe and synthesize a small application containing a 4PS signal. In this work we show that there are three different cases to be considered when a 4PS signal is present in a burst-mode specification. They depend on the dynamic behavior of the signal during the controller operation. The results obtained when using the flexible Miriã synthesis tool starting from a flexible multi-burst graph specification for all three cases were superior to all previously reported studies.

Keywords: asynchronous logic, flexible XBM, hazard, critical races, gRS architecture and synthesis.

1. Introduction
There has been a growing interest in asynchronous circuits in recent years 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 problems of clock skew and clock distributed network, are more robust in respect to temperature variations and electromagnetic interactions [2]. However it is not easy to design asynchronous circuits free of hazards and critical races [3].

Asynchronous circuits design methodologies may be classified in three styles [1,2]: translation methods, graph-based methods and asynchronous finite state machines (**AFSM**).

The AFSM style was originally proposed by Huffman [4]. Later, Nowick [6,10,17] introduced the burst-mode (BM) machines synthesized from a burst-mode specification. Several interfaces implemented as BM machines presented very good performance like DRAM [7], SCSI [8], HP Post Office [5] and Cache [9]. In order to describe heterogeneous (synchronous /asynchronous) systems behavior, Yun [11,12,13] proposed the extended burst-mode (XBM) machines [16]. The XBM specification added two signal types to the BM specification: directed don't-care (DDC) and conditionals. The latter are level sensitive signals (LSS), i.e., they are active during their "0"or '1" phase as opposed to all others that are transition sensitive signals (TSS), i.e., they are active during the "0→1 or 1→0" transition. LSS signals may present non-monotonic behavior. A signal is either LSS or

TSS (it may NOT change from one type to the other). Very good results were presented for XBM controllers like SCSI [14], differential equation solver [15]. Oliveira et al. [20] further extended the XBM specification adding two operators: concurrency (CO) between a pair of sequential transitions and OR between a pair of input signals. The multi-burst graph (MBG) specification was adopted as the input specification of the Mirifã synthesis tool. Again good results were obtained for controllers implemented as gRS architectures[1] (figure 1) [21].

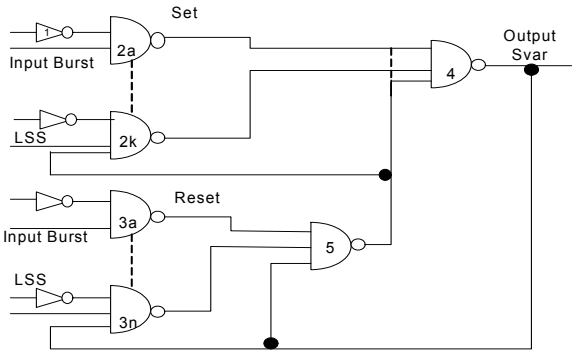

Figure 1 - The gRS architecture.

There exist applications whose behavior contain "four phase signals" (4PS). Such signals may change from TSS to LSS behavior during the controller's operation [4]. Hence, the signal is active either during its value transition ($0\rightarrow1$, $1\rightarrow0$) or during its stable value ($=1$, $=0$). In [22,25] Vanbekbergen et all proposed the use of 4PS signals targeting their generalized STG (signal transition graph) based synthesis tools. [23,24] One limitation of their approach is the fact that the STG description is later transformed into a SG (state graph) that grows exponentially for large problems (which is the case when non-monotonic LSS signals are present). Recently Kraus [18,19] proposed the XBM2PLA tool for XBM machines, starting from a more flexible XBM specification (F-XBM) that accepts 4PS signals. He showed that the use of a 4PS signal reduces the size of the circuit's description producing the same result.

In this paper we present the flexible multi-burst graph specification (F-MBG) that also accepts 4PS signals. We show that three different cases may occur according to the dynamic behavior of such signals. For each case our method leads to some optimization. Let us call St_L the state where the signal s_H changes from TSS to LSS behavior and St_T the previous state.

1. The controller specification may be reduced in respect to any other specification if during the $St_T \rightarrow St_L$ transition, the 4PS signal s_H either assumes a known value (s_H^+ , s_H^-) or remains constant. In other words the 4PS signal has a *defined value* during all state transitions. We call such a signal as DV_4PS (defined-value 4PS signal).
2. The resulting controller may require less state variables (less area) than any other solution if during the $St_T \rightarrow St_L$ transition, the 4PS signal s_H assumes a directed don't care value (s_H^*)[2]. In other words the 4PS signal has an *undefined value* just before becoming an LSS signal but it is *always* accompanied by at least one compulsory TSS signal during all other state transitions. We call such a signal as UV_4PS (undefined-value 4PS signal).
3. The controller's specification would need an extra input signal in any other input specification if during the $St_T \rightarrow St_L$ transition, the 4PS signal s_H assumes a directed don't care value (s_H^*)[1] and s_H is the only signal present in any other state transition. In other words the 4PS signal has an *undefined value* just before becoming an LSS signal but it is the *only active signal* during any other state transition. We call such a signal as SUV_4PS (single-undefined-value 4PS signal).

All three cases are handled by the F_Mirifã synthesis tool[3].

In sections 2, 3 and 4 we detail each case illustrating with small examples. Section 5 presents our conclusions and future work

2. Defined-Value 4-Phase Signal

Figure 2 shows part of the timing diagram of the mutual exclusion benchmark [18]. Signals R1 and R2 are 4PS signals. Figures 3 and

[1] The Mirifã tool also targets the gC architecture.

[2] The optimization is also true if the signal s_H behavior is non monotonic during the $St_T \rightarrow St_L$ transition. In this case it is denoted by $s_H^{\#}$.

[3] F_Mirifã is the extended version of the Mirifã tool [20].

4 show the same behavior specified as a multi-burst graph. Signals R1 and R2 were captured as TSS signals in figure 3 and as 4PS signals in figure 4. Figure 5 shows the table of signal transition cubes that results from the specification in figure 4. The state minimization step (that follows from the table of STCs) produced the same number of state variables for both specifications. Figure 6 shows the resulting gRS architecture produced by Miriã and F_Miriã for both specifications (same circuit). This example shows the improvement that may be achieved when capturing and dealing directly the 4PS signals instead of transforming them into pure TSS signals.

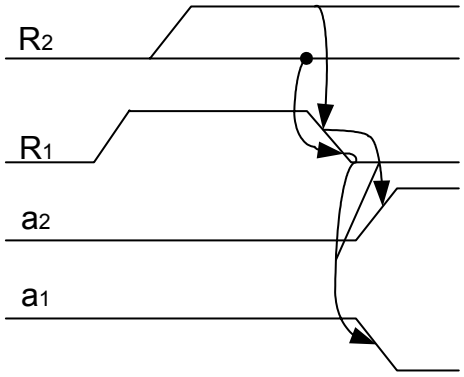

Figure 2 – Timing diagram with the 4PS signal *R2*.

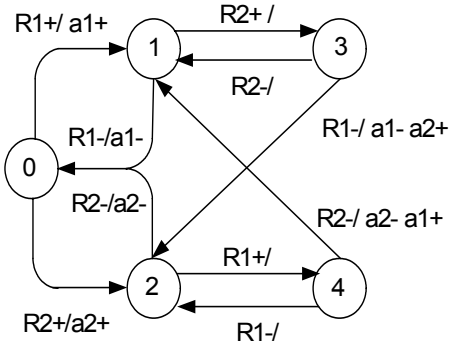

Figure 3 – MBG specification of figure 2 (*R2* has been captured as a TSS signal).

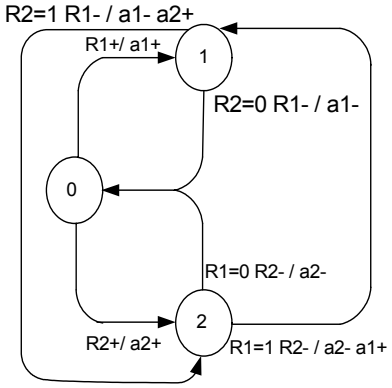

Figure 4 – F-MBG specification of figure 2 (*R2* has been captured as a 4PS signal).

R1 R2 a1 a2=0000

ARC	STC
0--1	2000 1020
0--2	0200 0102
1--0	2110 0122
1--2	2010 2010
2--1	1201 1022
2--0	0201 0002

Figure 5 - Table of STCs for the F-MBG specification (figure 4).

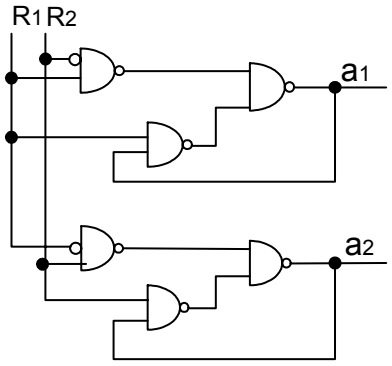

Figure 6 – Resulting gRS architecture for both specifications.

3. Undefined-Value 4-Phase Signal

Figure 7 shows a state diagram that contains the 4PS signal *a*. We assume that *a* may present non-monotic behavior before it becomes level sensitive. Figures 8 and 9 show the same behavior specified as a multi-burst graph. Signals *a* was captured as a TSS signal in figure 8 and as a 4PS signal in figure 9. The state minimization step (using Miriã) produced 3 state variables for the first specification and 2 for the second (using F_Miriã).

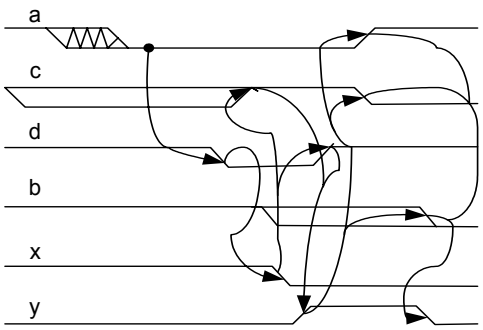

Figure 7 – Timing diagram with the 4-phase signal *a*.

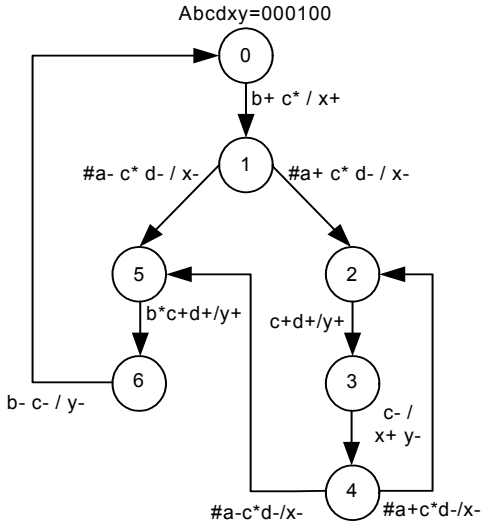

Figure 8 – MBG specification of figure 7 (*a* has been captured as a TSS signal).

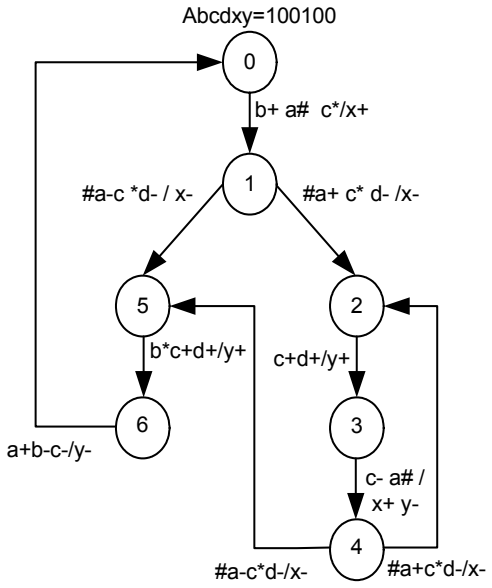

Figure 9 – F- MBG specification of figure 7 (*a* has been captured as a 4-phase signal).

Figure 10 shows the HP-Sbuf-send-pkt2 benchmark timing diagram contains the 4PS signal *Done* whose behavior is non-monotonic [7]. The same number of state variables was required for both cases (Miriã and F_Miriã) for this benchmark.

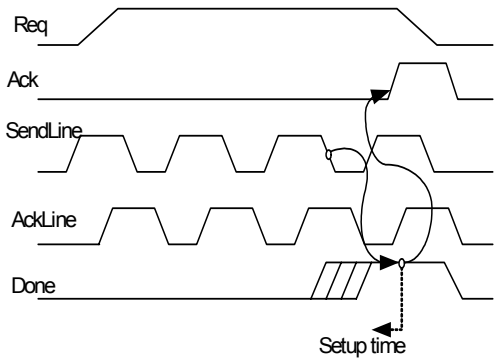

Figure 10 – Timing Diagram of the HP-Sbuf-send-pkt2 (4PS signal *Done*).

4. Single-Undefined-Value 4-Phase Signal

Figures 11a,b,c,d show part of the timing diagram of the I^2C benchmark illustrating the behavior of the 4PS signal SDA [22]. Figure 12 shows the F-MBG specification of the same behavior. Notice that SDA is the only signal active during the 7→10 and 1→2 state transitions. The authors don't know any synthesis tool that could handle this case. In order to capture this behavior using the MBG specification[4], a dummy TSS signal *aux* would have to be added as shown in Figure 13.

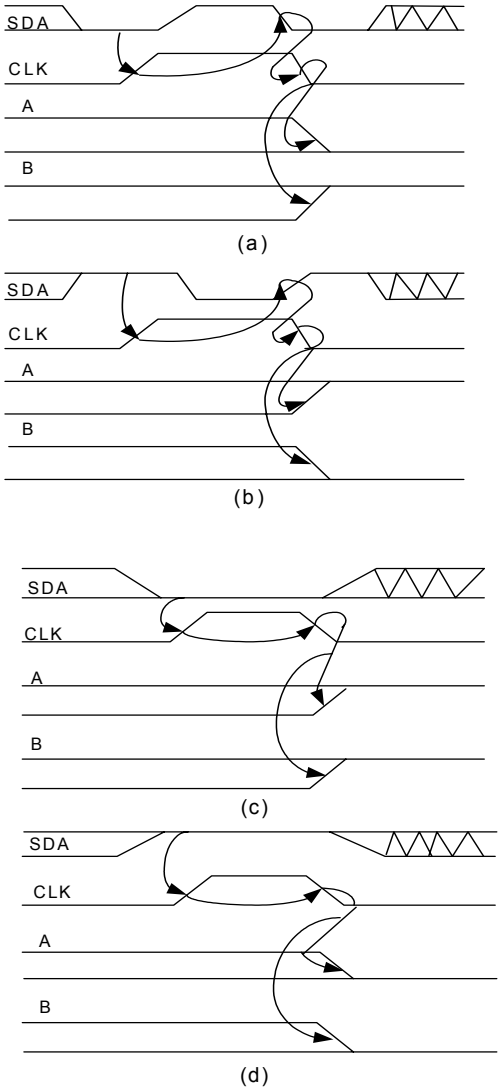

Figure 11 – Timing diagram of the I2C benchmark (4PS signal *SDA*).

[4] The same is true for the XBM specification.

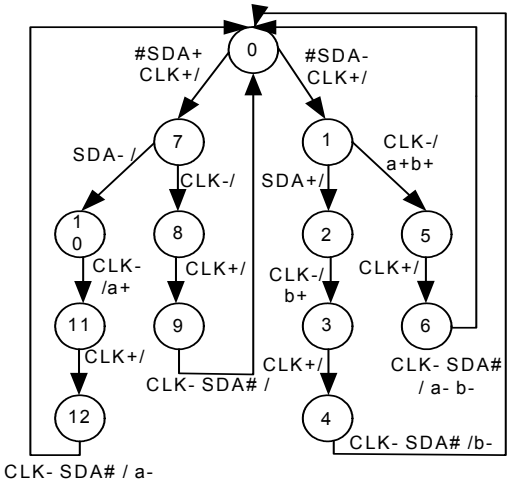

Figure 12 – F-MBG specification of figure 11 (*SDA* is a 4-phase signal).

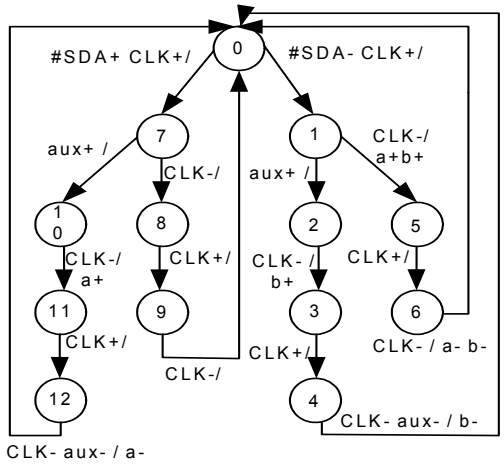

Figure 13 – MBG specification of figure 11 adding a dummy TSS signal (*aux*).

5. Conclusions and future Work

We presented extended versions of the multi-burst graph representation (called F_MBG) and the Miriã synthesis tool (called F-Miriã) able to capture and handle asynchronous controllers whose behavior contains 4-phase signals (4PS). Such signals may change from signal transition sensitive (TSS) to signal level sensitive (LSS) behavior during the controller's operation. Such signals are considered by other synthesis tools. We showed that three different situations may occur according to the dynamic behavior of the

4PS signals. If its value is always defined, we obtain a smaller description than any other specification. For all the examples we tested there was neither area nor performance penalty for our results compared to the Mirià and 3D tools. If its value is undefined (and non-monotonic) during one or more state transitions, but the 4PS signal is always accompanied by at least one TSS signal during all other state transitions we obtained a more efficient description and less state variables (hence less area) than any other tool. Finally, if its value is undefined and it is the only signal active during at least one state transition, as far as we know our F_Miriã tool is the only capable of finding a solution for such a case. All three cases were illustrated with known and homemade benchmarks. We intend to include this kind of signal into the SI-Miriã tool [20,21]. We also expect to test our tool for larger examples.

REFERENCES

[1] M. Renaudin, "Asynchronous circuits and systems: a promising design alternative," Microelectronic Engineering, vol. 54, 2000, pp133-149.

[2] S. Hauck, "Asynchronous Design Methodologies: An Overview", Proc. of the IEEE, January 1995, Vol. 83:1 pp.69-93.

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

[4] S. H. Unger, "Asynchronous Sequential Switching Circuits," John Wiley & Sons Inc, 1969.

[5] A. L. Davis, 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.

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

[7] Nowick, S. M. et al., " Practical Asynchronous Controller Design," Proc. ICCD, October, 1992, pp.341-345.

[8] Nowick, S. M., Dill, D. L., "Automatic Synthesis of locally-cloked asynchronous state machines," Proc. IEEE Int. Conf. CAD, November, 1991, pp 318-322.

[9] Nowick, S. M., et al, "The design of a high performance cache controller: A case study in asynchronous synthesis," Integration the VLSI Journal, vol.15 no 3, October, 1993, pp.241-262.

[10] S. M. Nowick and 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, pp.986-997.

[11] K. Y. Yun, "Synthesis of Asynchronous Controllers for Heterogeneous Systems", PhD thesis, Stanford University, 1994.

[12] K. Y. Yun and D. L. Dill, "Automatic Synthesis of Extended *Burst*-Mode Circuits: Part I (Specification and Hazard-Free Implementation)," IEEE Trans. on CAD of Integrated Circuit and Systems, Vol. 18:2, February 1999, pp. 101-117.

[13] K. Y. Yun e D. L. Dill, "Automatic Synthesis of Extended *Burst*-Mode Circuits: Part II (Automatic Synthesis)," IEEE Trans. ON CAD of Integrated Circuit and Systems, Vol. 18:2, February 1999, pp 118-132.

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

[15] K. Y. Yun, et al. "The design and verification of a high-performance low-control overhead asynchronous differential equation solver," IEEE Trans. on VLSI Systems, vol.6, no4, December, 1998.

[16] K. Y. Yun and D. L. Dill, "Unifying synchronous/ asynchronous state machine synthesis," Proc. Int. Conf. Computer Aided Design (ICCAD), Nov., 1993, pp. 255-260.

[17] R. M. Fuhrer, S. M. Nowick et al., "MINIMALIST: An environment for the synthesis and verification of burst-mode asynchronous machines," Proc. IEEE/ACM Int. Workshop logic Synthesis, 1998.

[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, 2003, pp.1301-1303.

[19] O. Kraus, "*Synthese von digitalen Asynchronen Zustandsautomaten*," thesis – Doktor-Ingenieur Universitat Erlangen-Nurnberg, 2003.

[20] 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 (2003) 209-213.

[21] D. L. Oliveira, "*Miriã: uma ferramenta de síntese para controladores multi-rajada*,", Tese de Doutorado, Universidade de São Paulo, 2004.

[22] P. Vanbekbergen, et al. "A generalized signal transition graph model for specification of complex interfaces," European Design and

Test Conference, EDAC, pp.378-384, March, 1994.

[23] Yakovlev, A. V., "On Limitations and Extensions of STG model for Designing Asynchronous Control Circuits," Proc. Int. Conf. Computer Design, October, 1992, pp.396-400.

[24] J. Cortadella, et al., "Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers," IEICE Trans. on Inf. And Systems, vol.E80-D, no3, March, 1997, pp.315-323.

[25] C. Ykman-Couvreur, et al., "Assassin: A Synthesis for Asynchronous Control Circuits," Tech. Rep. IMEC, User and Tutorial manual, September, 1994