Asynchronous Control for Low Power FSM Partitioning with One-Hot Encoding

Duarte L. Oliveira, Lester A. Faria, Luiz S. Ferreira, Noé Alles
Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – ITA – IEEA
Marechal Eduardo Gomes, 50 – CEP 12.228-900 – SJC – SP – Brasil

***Abstract* - Currently, many digital systems are described by an architecture composed by synchronous finite state machines (FSM) networks and data paths. Generally, they are battery powered and implemented in PLD technology (Programmable Logic Device) what leads to the demanding of batteries that have long useful life. Therefore, reducing power consumption is one of the most important tasks in circuit design. In order to design Low Power devices, an interesting approach is the partitioning of the FSM into two or more sub-FSM. In this paper it is proposed a method of FSM-partitioning, focusing on power consumption reduction. The proposed method generates sub-FSM with Shared State Memory and asynchronous communication control, being encoded in one-hot code. Through a study of case, it is possible to perceive a 25% reduction in latency and 10% in cycle time, despite an increase of 19% of transistors, in comparison to a similar case implemented by another established method of literature, showing high potential of implementation in aeronautical and aerospace systems that demand low power.**

***Key-words* — synchronous finite state machines, low-power synthesis, asynchronous control, partitioning, shared state memory, one-hot code.**

I. INTRODUCTION

Currently, the advancement of microelectronics leads to the designing of increasingly complex digital systems. Most of these systems are battery powered and focus on different applications such as wireless, laptops, aerospace (satellite and missile), aviation, automotive and medical. Being battery powered it is desirable that these devices have long useful life, making power dissipation to be an important parameter in the design of these systems [1,2]. In this context, the synthesis of synchronous finite state machines (FSM) has an important role in the design of digital circuits powered by batteries.

Many digital circuits are described by an architecture composed by a network of controllers with data paths and/or processors [3]. The synchronous controllers of these devices are often described as FSM. They can always be specified by a State Transition Graph (STG).

Generally, these circuits are implemented in VLSI technology, in PLD (Programmable Logic Devices), especially CPLDs (Complex PLD), and FPGA (Field Programmable Gate Array).

This last one has grown enormously in recent years, generating FPGA with up to 50 million gates, allowing increasingly complex digital systems to be programmed [4].

Duarte L. Oliveira, duarte@ita.br, Lester A. Faria, lester@ita.br, Luiz S. Ferreira, lsperna@terra.com.br, Noé Alles, noealles@ita.br

This last one has grown enormously in recent years, generating FPGA with up to 50 million gates, allowing increasingly complex digital systems to be programmed [4].

On the other hand, PLDs have become popular means to implement digital circuits, due to its dramatic reduction in the cost and time consuming.

Digital circuits are implemented with CMOS components. In this technology the major part of the power dissipation occurs during switching (dynamic power) [5], consisting mainly of two parts: the combinatorial part, related to the excitation and output equations, and the sequential part, related to the flip-flops (FFs). In general, dynamic power is given by:

$$P_{dinamic} = \frac{1}{2}.C.V_{DD}^{2}.f.N \qquad (1)$$

where V_{DD} is the supply voltage, f is the operation frequency, N is the switching factor (number of transitions in the output of a gate or FF) and C is the capacitance seen at the gate or FF output [5].

In digital circuits, the sequential part is the largest contributor to the power consumption. Recent studies indicate that the clock of these circuits consume a large percentage of the total power (15% to 45%) [1] [2]. This consumption is related to the buffers, clock distribution network and registers.

Techniques for reducing dynamic power can be applied at different levels of digital design [1] [2]. Currently, the dynamic power reduction methods to the synthesis of FSM are mainly focusing in the clock control (clock-gated) [6] [7], “double-edges triggered flip-flops” [8], partitioning of FSM [9-11], multi-criteria state assignments [12,13] and logic minimization and technology mapping [14-16].

In this paper, it is proposed a partitioning design model for a Moore FSM, which is encoded in one-hot code. Structurally, the model uses shared state memory and, as communication mechanism between sub-FSM, asynchronous control. The method is able to partition a Moore FSM in two or more sub-FSM, encoding them in one-hot code. If compared to other projects templates [9-11] [17-22] this method avoids the insertion of an extra state, as generally done by other established methods, and reduces drastically the number of FFs, if compared to other systems that use one-hot code. The greater advantage of the proposed asynchronous control compared to that proposed by Cao et al. [20-22] is the higher performance. The control is implemented as a feedback output Huffman machine [23]. It is not based on latches and do not need any decoding block. Fig. 1 shows the proposed architecture for the Moore FSM.

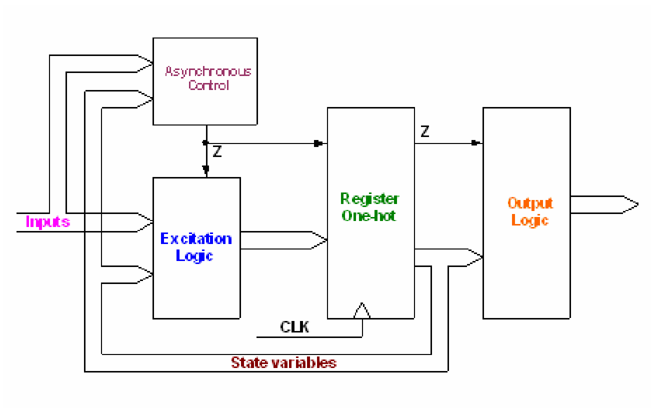


Fig. 1. Proposed target architecture with two sub-FSM.

In section II of this paper they are presented some theoretical concepts, which will be important for the understanding of the proposed method, while in section III the proposed design procedure is described. Finally, in section IV, the method is illustrated by a study of case, being analyzed in section V. In section VI some concluding remarks are done, highlighting the advantages and limitations of the proposed design method.

II. THEORETICAL CONCEPTS

A. Partitioning of FSM

Partitioning has been shown to be a very effective technique to reduce power in FSM [1,2]. The partitioning methods start always from a, so called, monolithic FSM derived from a STG, generating two or more sub-FSM [9-11,17-22].

In order to implement the partitioned sub-FSMs, different methods have been proposed in the literature. In all of them only one sub-FSM is activated at a time. There are two kind of possible classifications, based both on structural aspect and on communication mechanism between sub-FSMs. Each one of these classifications can be divided in two classes. Based on structural aspect, they can be: 1) each sub-FSM has its own state memory [9-11] (see Fig. 2); 2) all the sub-FSM share a single state memory, also called the Local State Memory [17] [20-22] (see Fig. 3). According to the communication mechanism, the methods can be divided in: 1) communication between sub-FSM is performed synchronously [9-11] [17] and 2) communication between sub-FSM is asynchronous, that is, there is an asynchronous control to activate and deactivate the sub-FSM [18-22] (see Fig. 4 and 5). Cao et al. [20-22] show that designs that use shared state memory and asynchronous control achieve higher power reduction, when compared to the other kinds of designs.

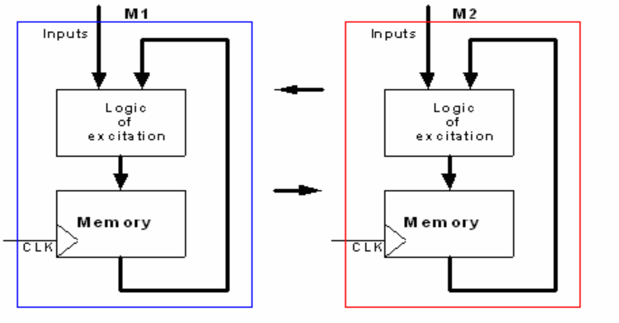


Fig. 2. Separate State Memory partitioning.

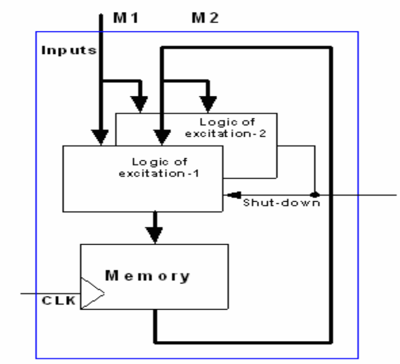


Fig. 3. Shared State Memory partitioning.

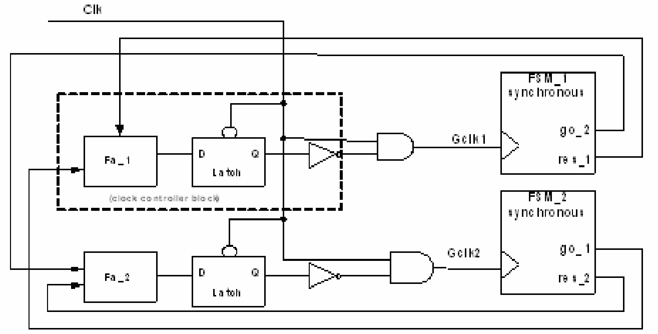


Fig. 4. Synchronous communication partitioning for two sub-FSM [9].

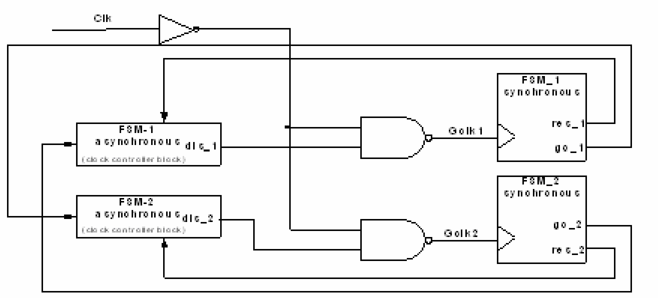


Fig. 5. Asynchronous communication partitioning for two sub-FSM [18].

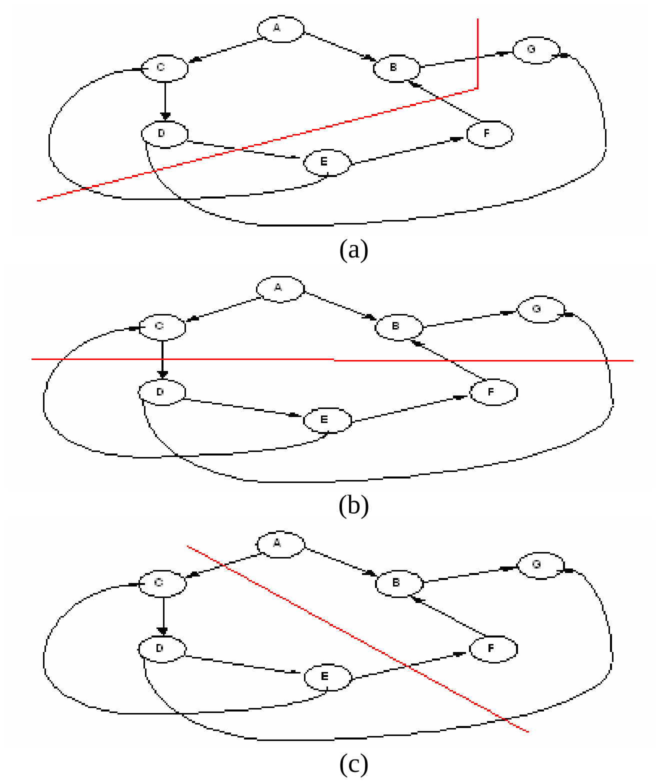
B. One-hot Code to FSM

The one-hot code is characterized by having a state variable for each one of the states of the FSM and, in every state, only one of the variables achieves a high digital value (1) at a time, remaining the other variables at a low digital value (0). The one-hot encoded FSMs have interesting features [24] [25]. For large FSMs, the one-hot code tends to present excitation and output Boolean smaller equations than those that use binary code [24]. This feature is very important when implemented in PLDs, once the logic of excitation (input) and output that present high fan-in use many cells of a PLD, thus leading to an increase in area and a particular degradation in performance (latency and cycle times).

Other important advantages of the one-hot code are its reduction of glitches in the Boolean equations, increasing in reliability and robustness to radiation, as showed by Cassel et al. [26] once the Single Event Upset (SEU) and Single Effect Transient (SET) can be more easily diagnosed. The major drawback of the code is the increasing number of FFs, once each state requires a FF. This drawback is highly decreased with the partitioning of the FSM, as proposed in this work.

C. FSM Partitioning Strategy

FSM described by a STG can be partitioned into two or more sub-FSMs. In literature, different strategies have been proposed to divide the STG [9-11] [17-22]. The most commonly used strategy is to separate state transitions with low probability of occurrence. In this paper it was implemented a strategy that reduces the number of transitions that cross the sub-FSMs, reducing also the involvement of conditional states at the border. The priority of the partitioning is the generation of sub-FSM with the same number of states. Fig. 6a, b, c show three possible different divisions of a same STG, generating two sub-FSM.



(a)

(b)

(c)

Fig. 6. STG bi-partitioned: a) five arcs; b) four arcs; c) three arcs.

D. Asynchronous control

An asynchronous control (AC) is a Mealy asynchronous FSM that serves to activate and deactivate the N existing sub-FSMs and operates in fundamental mode [23]. It is described in the flexible extended burst-mode specification (FXBM), which allows multiple input changes [27]. AC FXBM is described from the N coded STGs where the state transitions (divided arcs) that cross the sub-FSM are declared. The AC specification is obtained just in these transitions. When the division is performed in an unconditional state transition, the FXBM specification uses only the involved state variable. This variable, in FXBM, is seen as a sensitive signal to the transition and has a monotonic behavior. The example of Fig. 6a shows the division of an unconditional state transition B→G. Fig. 7a shows its description in FXBM. When the division is performed in a conditional cross-state transition, the FXBM specification uses both the state variables and the input signals. The input signals do not have a monotonic behaviour (allow glitches) and can be sensitive to the level (symbol <sign>) and sensitive to the transition [27].

The example of Fig. 6a shows the conditional state E, with a cross-state transition E→C. Fig. 7b shows its description in FXBM. There are conditional states where the transitions cross to another FSM, for example, the state D (D→E and D→G), seen in Fig. 6a. So, in this specification, FXBM it is treated as unconditional.

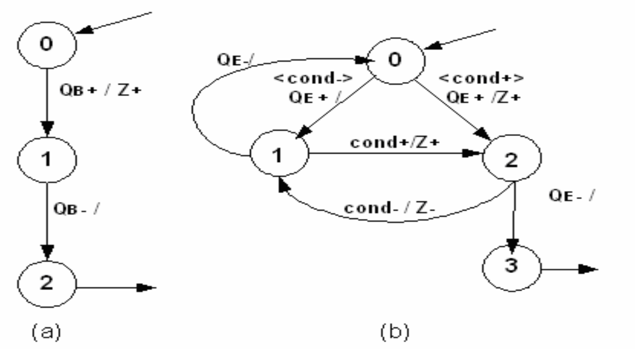


Fig. 7. Part of a FXBM specification: a) unconditional; b) conditional.

III. SYNTHESIS PROCEDURE

The methodology is composed by 6 steps and starts from the STG specification.

1: Perform state minimization of the STG, generating the STG_{MIN} (used tool: SIS [28]).
2: Perform manually the partitioning of STG_{MIN} in $\{\text{STG1}_{\text{MIN}},\text{STG2}_{\text{MIN}}....\text{STGN}_{\text{MIN}}\}$, which will refer to the sub-FSM $\{\text{FSM}_1, \text{FSM}_2 ....\text{FSM}_N\}$.
3: Encode, manually, $\{\text{STG}_{\text{1MIN}}, \text{STG}_{\text{2MIN}}...\ \text{STG}_{\text{NMIN}}\}$ in one-hot code, generating $\{\text{STG}_{\text{1MIN-COD}}, \text{STG}_{\text{2MIN-COD}}... \text{STG}_{\text{NMIN-COD}}\}$, where the used cross-states of $\{\text{MFE}_1, \text{MFE}_2 .... \text{FSM}_N\}$ must be encoded with the same variable.
4: Specify the asynchronous control $\{\text{STG}_{\text{1MIN-COD}},\text{STG}_{\text{2MIN-COD}}....\text{STG}_{\text{NMIN-COD}}\}$, using the FXBM specification (FXBM control).
5: Synthesize, in Huffman architecture with the feedback output, the FXBM control, free of logical-hazard and critical race [23] [27] [29].
6: Perform logic minimization to $\{\text{STG}_{\text{1MIN-COD}}, \text{STG}_{\text{2MIN-COD}}....\text{STG}_{\text{NMIN-COD}}\}$, obtaining the excitation equations and the output equations, with the introduction of control signal Z (used tool: SIS [28]).

The proposed method starts with the STG. The first step performs its states minimization. This step can be performed by different algorithms, for example, a computational tool called SIS. The next step performs the partitioning of the minimized STG. The number of sub-FSMs is related to the size of the STG. Based on a trade-off between cost and benefits, the designer will be able to set the memory size of the local state. The third step is the one-hot encoding for each one of the sub-FSM.

Those states that cross the sub-FSMs must to be encoded with a same code (same variable). The steps four and five are related to the synthesis of asynchronous control. Finally, the sixth step performs the logic minimization for each one of the sub-FSMs, which can be performed by the same tool SIS.

These procedures will be implemented through a study of case.

IV. STUDY OF CASE

In order to show the efficiency of the proposed method, it is presented a study of case where a detector of two sequences is analyzed (0101 and 1010). Despite being small sequences, this is an excellent example of the method application because the same procedures can be used for larger sequences as well. Fig. 8 shows the minimized Moore STG of the detector (step 1). Step 2 performs partitioning.

The division was made in the state transitions D→E and I→A of the STG, generating partitions P1 (A, B, C, D) and P2 (E, F, G, H, I) (see Fig. 9). The third step performs the one-hot encoding. The sub-FSM-2 from P2 requires five states variables. The initial states D and I of cross-transitions of the two sub-FSMs are encoded with the same code ($Q_1Q_2Q_3Q_4Q_5$=00010). The other states of both sub-FSMs may have equal one-hot codes, but different than those used to D and I (see Fig. 10 and 11). Fig. 12 and 13 show FXBM specification and the table of asynchronous control flow.

As the state D is a conditional one, it must be added the input signal X in asynchronous control. Fig. 14 to 16 show the synthesis and the control logic circuit (steps 4 and 5). Step 6 performs the logic minimization of sub-FSM-1 and sub-FSM-2. The logic circuits of both sub-FSMs can be seen in Fig. 17 to 19.

Equations of excitation of sub-FSM-1:

$DA=(X'Q_4 + XQ_3 + X'Q_2 + XQ_1)Z'$

$DB = X'Q_1 Z'$

$DC = X Q_2 Z'$

$DD = X' Q_3 Z'$

Equations of excitation of sub-FSM-2:

$DE = (X Q_4 + X Q_2 + X' Q_3 + XQ_5 + X'Q_1)Z$

$DF = X Q_1 Z$

$DG = X' Q_2 Z$

$DH = X Q_3 Z$

$DI = X'Q_5 Z$

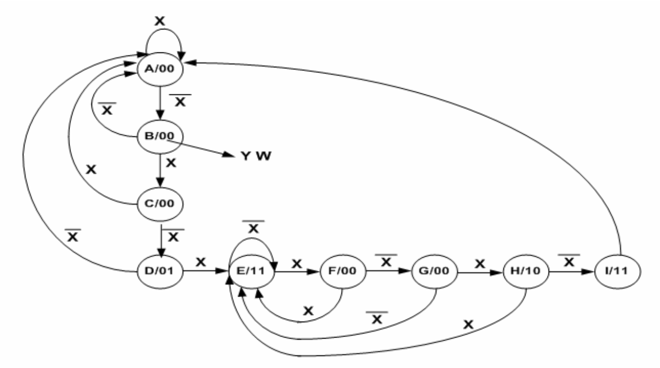


Fig. 8. STG specification: Two sequences detector.

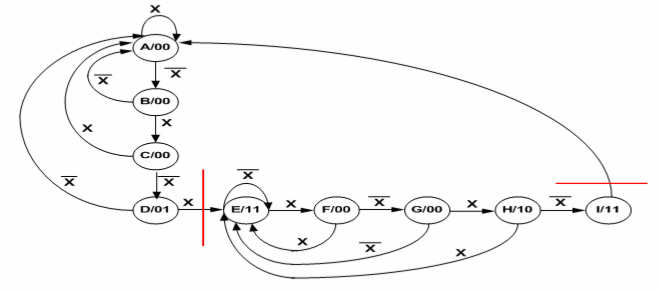


Fig.9. STG specification: division on the D→E and I→A transitions.

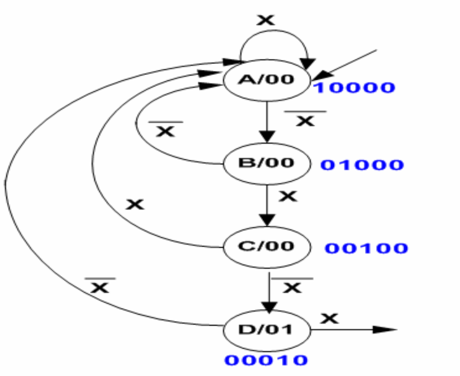


Fig. 10. Encoded STG specification: sub-FSM-1

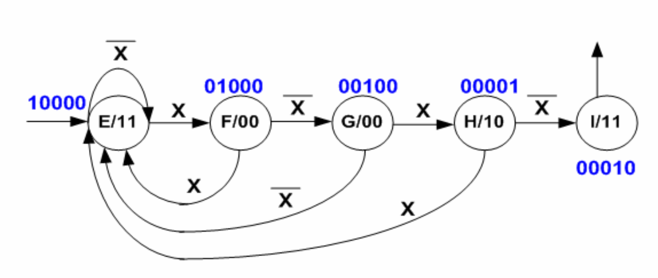


Fig. 11. Encoded STG specification: sub-FSM-2

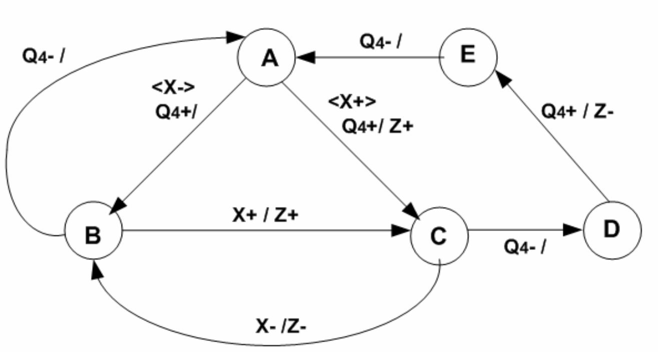
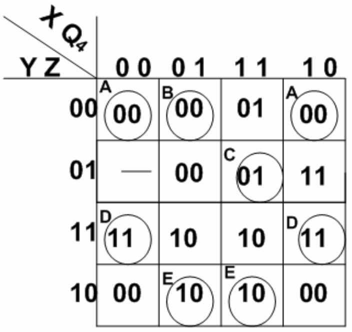
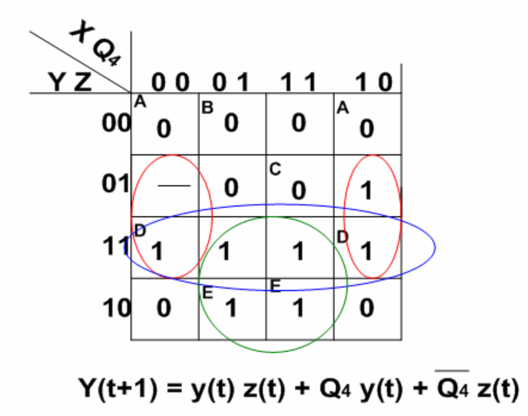


Fig.12. FXBM specification of the asynchronous control.



YZ \ XQ4	0 0	0 1	1 1	1 0
00	A 00	B 00	01	A 00
01	—	00	C 01	11
11	D 11	10	10	D 11
10	00	E 10	E 10	00

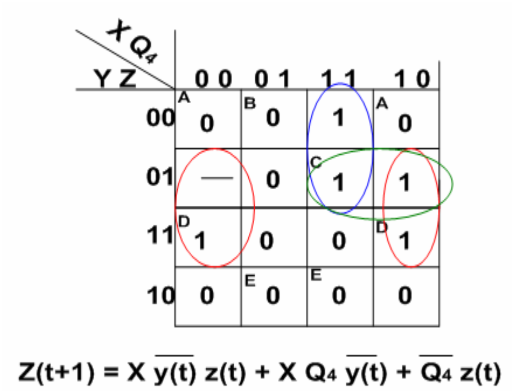
Fig. 13. Table of the asynchronous control FXBM flow



YZ \ XQ4	00	01	11	10
00	A 0	B 0	0	A 0
01	—	0	C 0	1
11	D 1	1	1	D 1
10	0	E 1	E 1	0

$$Y(t+1) = y(t)\,z(t) + Q_4\,y(t) + \overline{Q_4}\,z(t)$$

Fig. 14. Map of Karnaugh: Next state equation: Y.



YZ \ XQ4	00	01	11	10
00	A 0	B 0	1	A 0
01	—	0	C 1	1
11	D 1	0	0	D 1
10	0	E 0	E 0	0

$$Z(t+1) = X\,\overline{y(t)}\,z(t) + X\,Q_4\,\overline{y(t)} + \overline{Q_4}\,z(t)$$

Fig. 15. Map of Karnaugh: Next state equation: Z.

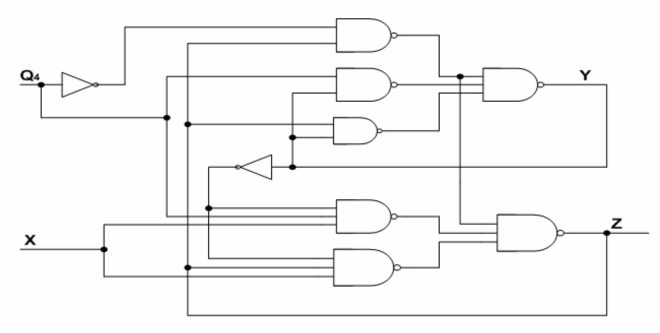


Fig. 16. Logic circuit: asynchronous control.

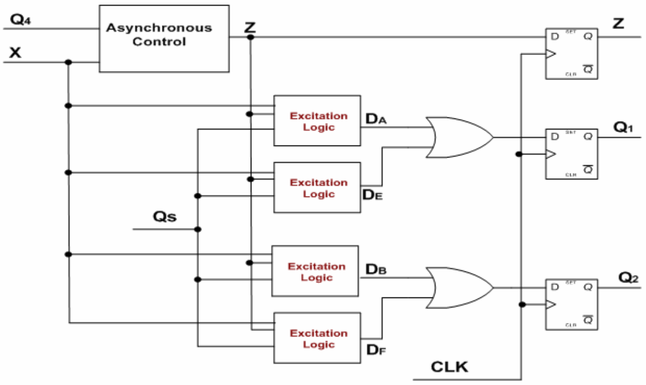


Fig. 17. Logic circuit: 1st part.

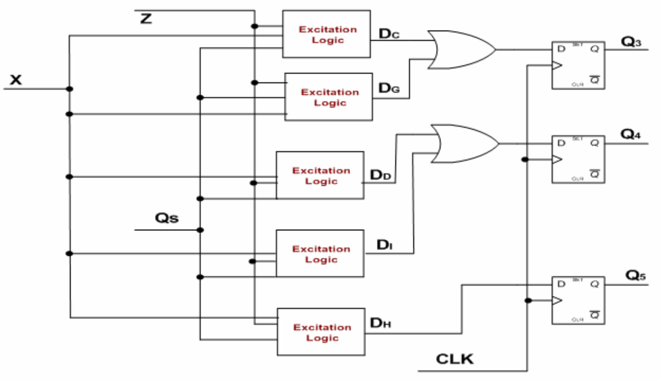


Fig. 18. Logic circuit: 2nd part.

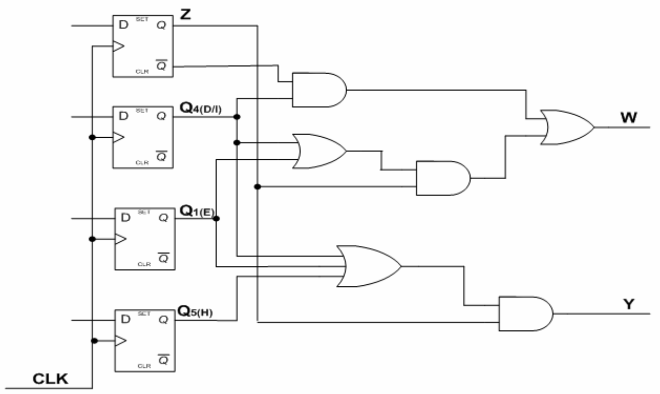


Fig. 19. Logic circuit: output equations.

V. SIMULATIONS RESULTS

In order to demonstrate the feasibility of the method and its advantages the proposed case of Section IV was simulated in the Altera QUARTUS II version 9.1 to the target family CYCLONE III in device EP3C5E14417. Fig. 20 shows a simulation of the sequences detector (0101 and 1010). Two main advantages can be seen immediately. In this method it is not needed an extra state, as implemented by the other established methods, therefore leading to an economy of one cycle. Other important features are related to the one-hot code that leads to excitation and output Boolean smaller equations, reduction of glitches in the Boolean equations, increasing in reliability and robustness to radiation.

Doing qualitative analysis in relation to the used area (number of transistors) and performance of asynchronous control (see Fig. 21), it was required 40 transistors with a latency time of 60 ns and cycle time of 90 ns. This result, compared to that of Cao et al., showed an increase of 19% of transistors but a 25% reduction in latency and 10% in cycle time, what showed to have a great potential of implementation, once the performance was highly incremented, enabling higher operation frequencies. On the other hand, it was perceived a Hamming distance (one-hot code) of two, instead of one, that is a desired level to dynamic power reduction, for example in Gray Code.

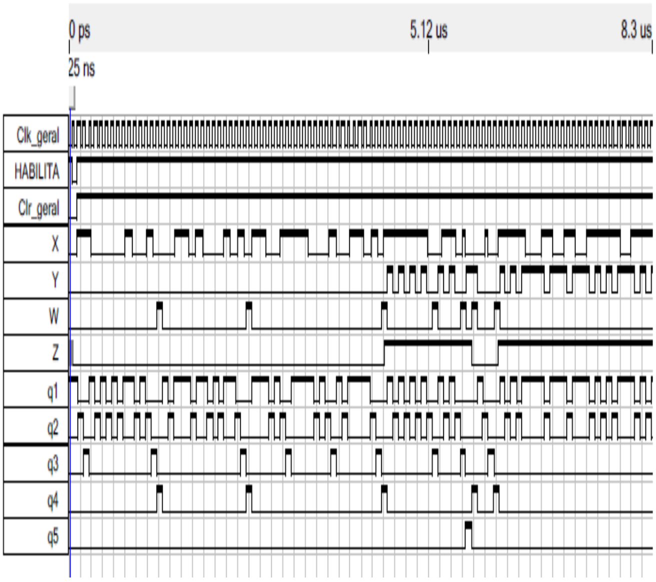


Fig. 20. Simulation: study of case (detector of two sequences).

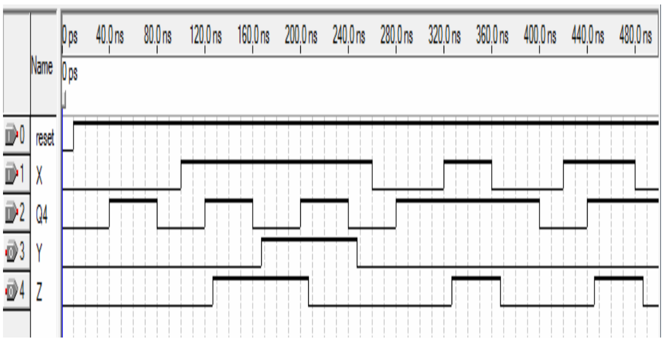


Fig. 21. Simulation: proposed asynchronous control.

VI. CONCLUDING REMARKS

It was proposed a new partitioning method for FSM using shared memory, asynchronous control and one-hot encoding. This method showed to be more efficient than the previously established methods of the literature, once it presented a 25% reduction in latency and 10% in cycle time, enabling higher operation frequencies, although through an increase of 19% of transistors.

The use of one-hot code in this implementation allows smaller excitation and output Boolean equations than those that use binary code, what is desirable when implementing devices in PLDs, once the logic of excitation (input) and output that present high fan-in use many cells of a PLD, leading to an increase in area and a particular degradation in performance (latency and cycle times). It achieves, as well, a reduction of glitches in the Boolean equations, increasing the reliability and robustness to radiation of the system. The major drawback of this code showed to be the increasing number of FFs, what can be minimized through the partitioning of the FSM.

The increasing in performance and the reduction in power dissipation leads to a high potential of its implementation in Low Power Devices, mainly in aeronautical and aerospace devices.

As future work, it is proposed a specific tool based on genetic algorithm for the partitioning of FSM, once in this work it was done manually, as well as an estimation of dissipation power for a large set of benchmarks. These features will provide a better evaluation of this method [30], allowing the dimensioning of its advantages over other methods in bigger circuits.

REFERENCES

[1] S. Devadas and S. Malik, “A Survey of Optimization Techniques Targeting Low Power VLSI Circuits”, Proc. 32nd ACM/IEEE DAC, pp.242-247, 1995. L. Benini and G. De Micheli, “Systems-Level Power Optimization: Techniques and Tools,” *ACM Trans. on Design Automation of Electronic System*, 5(2), pp. 115-192, April 2000.

[2] Li-Chuan Weng, X. J. Wang, and Bin Liu, “A Survey of Dynamic Power Optimization Techniques,” Proc. Of the 3rd IEEE Int. Workshop on System-on-Chip for Real-Time Applications, pp. 48-52, 2003.

[3] H. Hsieh, F. Balarin et. al. “Synchronous approach to the Functional Equivalence of Embedded System Implementations,“ IEEE Trans. On CAD of Int. Circuits and Systems, vol.20, no.8, pp.1016-1033, August 2001.

[4] J. J. Rodriguez, at. al., “Features, Design Tools, and Applications Domains of FPGAs”, IEEE Trans. On Industrial Electronics, vol. 54, no 4, pp. 1810-1823, August, 2007.

[5] F. N. Najm, “A Survey of Power Estimation Techniques in VLSI Circuits,” *IEEE Trans. on VLSI Systems*, vol. 2, no. 4, pp.446-455, December 1994.

[6] L. Benini and G. De Micheli, “Automatic Synthesis of Low-Power Gated-Clock Finite-State Machines,” *IEEE Trans. on CAD of Integrated Circuits and Systems*, Vol.15, No.6, pp.630-643, June 1996.

[7] Q. Wu, M. Pedram, and X. Wu, “Clock-Gating and Its Application to Low Power Design of Sequential Circuits”, IEEE Trans. on Circuits and Systems-I: Fundamental Theory and Applications, vol. 47, no.103, pp.415-420, March 2001.

[8] P. Zhao, J. McNeely, et al., “Low-Power Clock Branch Sharing Double-Edge Triggered Flip-Flops,” *IEEE Trans. on VLSI Systems*, vol. 15, no.3, pp.338-345, March 2007.

[9] L. Benini and G. De Micheli, “Synthesis of Low-Power Selectively-Clocked Systems from High-Level Specification,” *ACM Trans. on Design Automation of Electronic System*, 5(3), pp. 311-321, Julyl 2000.

[10] J. C. Monteiro and A. L. Oliveira, “Implicit FSM Decomposition Applied to Low-Power Design,” *IEEE Trans. on VLSI Systems*, Vol. 10, No. 5, pp.560-565, October 2002.

[11] B. Liu, et al., “FSM Decomposition for Power Gating Design Automation in Sequential Circuits”, 76th Int. Conf. on ASIC, ASICON, pp.944-947, 2005.

[12] M. Koegst et al., Multi-Criterial State Assignment for Low Power FSM Design. Proc. 24th EUROMICRO Conference, pp.261-268, 1998.

[13] P. Baccheletta et al. “Low-Power State assignment Techniques for Finite State Machines,” IEEE Int. Symposium on Circuits and systems, Geneva, Switzeriand, pp.641-644, May 2000.

[14] S. Iman and M. Pedram, “Two-level Logic Minimization for Low Power”, IEEE/ACM Conf. Int. on CAD Digest of Technical Papers, pp.433-438, 1995.

[15] J.-Mou Tseng and J.-Yang Jou, “A Power-Driven Two-Level Logic Optimizer,” Proc. Of the ASP-DAC, pp.113-116, 1997.

[16] R. I. Bahar and F. Somenzi, Boolean Techniques for Low Power Driven Re-Synthesis, IEEE/ACM Conf. Int. on CAD Digest of technical Papers, pp.428-432 1995.

[17] S. H. Chow, et al., “Low Power Realization of Finite State Machines – A Decomposition Approach ,” *ACM Trans. on Design Automation of Electronic System*, 1(3), pp. 315-340, July, 1996.

[18] B. Oelmann, et al., “Asynchronous Control fo Low-Power Gated-Clock Finite State Machines,” Proc. IEEE Int. Conf. on Electronics, Circuits and Systems, pp. 915-918, 1999.

[19] B. Oelmann, et al., “Automatic FSM Synthesis for Low-Power Mixed Synchronous/Asynchronous Implementation,” *Journal of VLSI Design*, Special Issue on Low-Power Design, vol. 12, no.2, pp. 167-186, 2001.

[20] C. Cao and B. Oelmann, “Mixed Synchronous/Asynchronous State Memory for Low Power FSM Design,” Proc. of the EUROMICRO Systems on Digital System Design,Rennes, France, pp.363-370, 2004.

[21] C. Cao, et al., “Synthesis Tool for Low-Power Finite-State Machines with Mixed Synchronous/Asynchronous State Memory,” IEE Proc. Comput. Digit. Tech. vol. 153, no. 4, pp.243-248, July 2006.

[22] C. Cao and B. Oelmann, "Low-Power State Encoding for Partitioned FSMs with Mixed Synchronous/Asynchronous State Memory," *Integration the VLSI Journal*, vol. 41, pp.123-134, 2008.

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

[24] G. Sutter, et al. "Low-power FSMs in FPGA: Encoding Alternatives," Lect. Notes Comput. Sci., pp. 459-467, 2451, 2002.

[25] L. Mengibar, et al. "Partitioned State Encoding for Low Power in FPGAs," ELECTRONICS LETTERS, vol. 41 , nro. 17, pp.xx , 2005.

[26] M. Cassel and F. L. Kastensmidt, "Evaluating One-Hot Enconding Finite State Machine for SEU Reliability in SRAM-based FPGAs," Proc. 12th IEEE Int. On-line Testing Symposium, pp. xx, 2006.

[27] 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.

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

[29] R. M. Fuhrer, et al., "Minimalist: An environment for the Synthesis, verification and testability of burst-mode machines," Technical Report, Columbia University, TR-CUCS-020-99, 1999.

[30] J. H. Anderson and F. N. Najm, "Power Estimation Techniques for FPGAs", IEEE Trans. On VLSI Systems, vol. 12, no. 10, pp.1015-1027, October, 2004.