

Low-Power FSMs in FPGA: Encoding Alternatives

G. Sutter¹, E. Todorovich¹, S. Lopez-Buedo², and E. Boemo²

1. INCA, Universidad Nacional del Centro, Tandil, Argentina
<http://www.exa.unicen.edu.ar/inca/>
{gsutter, etodorov}@exa.unicen.edu.ar

2. Computer Engineering School, Universidad Autónoma de Madrid, Spain
<http://www.ii.uam.es>
{sergio-lopez.buedo, eduardo.boemo}@ii.uam.es

Abstract. In this paper, the problem of state encoding of FPGA-based synchronous finite state machines (FSMs) for low-power is addressed. Four codification schemes have been studied: First, the usual binary encoding and the One-Hot approach suggested by the FPGA vendor; then, a code that minimizes the output logic; finally, the so-called Two-Hot code strategy. FSMs of the MCNC and PREP benchmark suites have been analyzed. Main results show that binary state encoding fit well with small machines (up to 8 states), meanwhile One-Hot is better for large FSMs (over 16 states). A power saving of up to the 57 % can be achieved selecting the appropriate encoding. An area-power correlation has been observed in spite of the circuit or encoding scheme. Thus, FSMs that make use of fewer resources are good candidates to consume less power.

Keywords: Low-Power, Finite State Machine, FPGA, One-Hot, State Encoding.

1. Introduction

Low-power design is nowadays a central point in the construction of integrated systems. It allows expensive packaging to be avoided, chip reliability to be increased, cooling to be simplified, and the autonomy of batteries be extended (or their weight to be reduced). The dynamic power dissipated in a CMOS circuits can be expressed by the well-known formula:

$$P = \sum_{all\ nodes} c_n f_n V_{DD}^2 \quad (1)$$

where, c_n is the load capacitance at the output of the node n , f_n the frequency of switching and V_{DD} supply voltage. The dominant source of power dissipation in CMOS circuits is the dynamic power: the energy required in each cycle to charge and discharge each node capacitance. It is also referred as the capacitive power dissipation.

Main idea in the design of low-power FSMs is minimize Hamming distance of the most probable state transitions. However, this solution usually increases the required logic to decode the next state. Then, a tradeoff between switching reduction and extra capacitance exists. This paper addresses the state encoding problem in LUT based programmable logic, using Xilinx 4K-series FPGAs as technological framework. In Section II, the basic definitions are summarized, and a review of the traditional approaches is presented. In the next section, the characteristics of the benchmark circuits are highlighted. Finally, the main experimental results are summarized.

2. Preliminaries

A finite state machines is defined by a 6-tuple $M = (\Sigma, \sigma, Q, q_0, \delta, \lambda)$, where Σ is a finite set of input symbols, $\sigma \neq \emptyset$ is a finite set of output symbols, $Q \neq \emptyset$ is a finite set of states, $q_0 \in Q$ is the “reset” state, $\delta(q, a) : Q \times \Sigma \rightarrow Q$ is the transition function, and $\lambda(q, a) : Q \times \Sigma \rightarrow \sigma$ is the output function.

The 6-tuple M can be described by a state transition graph (STG), where nodes represent the states, and directed edges, labeled with the input and output values, describe the transition relation between states. In hardware materializations, each state corresponds to a binary vector stored in the state register. From the current state and input values, the combinational logic computes the next state and the output function. The binary values of the inputs and outputs of the FSM are usually fixed by the particular application, while the state encoding can be defined by the designer.

2.1 Traditional approaches for State Encoding

The traditional methods used to generate state machines result in highly-encoded states. This type of machines typically has a minimum number of flip-flops but require implementing wide combinatorial functions.

Early research on FSM state encoding intended to minimize area or delay. For example, the NOVA tool implements an optimal two level state encoding [3], while the MUSTANG state assignment system [4] is targeted to multilevel networks. The JEDI tool [5] is a general symbolic encoding program (i.e., for encoding inputs, outputs, and states) targeted for multi-level implementations. This tool is included in the SIS system [6].

2.2 Approaches for Low Power State Encoding

Main works in low-power FSMs compute first the switching activity and transition probabilities [7]. The key idea is the reduction of the average activity by minimizing the bit changes during state transitions. In [8], a probabilistic description of the state machines is used. Then, the state assignment minimizes the Hamming distance between states with high transition probability. To obtain the probabilistic behavior of a general FSM, the STG is modeled as a Markov Chain, and the state algorithm

problem is solved using $\lceil \log_2 n \rceil$ bits, where n is the number of states. A spanning tree based state encoding algorithm is implemented in [9]. The most important characteristic is that the representation is not limited to $\lceil \log_2 n \rceil$. The resulting encoding can be ranging from $\lceil \log_2 n \rceil$ to n bits. Other interesting contribution are in [2], [21], [22].

circuits	Original Machine				Minimized Mach.			
	inp	outp	rul	#st	inp	outp	rul	#st
bbara	4	2	60	10	4	2	42	7
bbsse	7	7	56	16	7	7	208	13
bbtas	2	2	24	6	2	2	24	6
beecount	3	4	28	7	3	4	20	4
cse	7	7	91	16	7	7	91	16
dk14	3	5	56	7	3	5	56	7
dk15	3	5	32	4	3	5	32	4
dk16	2	3	108	27	2	3	108	27
dk17	2	3	32	8	2	3	32	8
dk27	1	2	14	7	1	2	14	7
dk512	1	3	30	15	1	3	30	15
donfile	2	1	96	24	2	1	4	1
ex1	9	19	138	20	9	19	233	18
ex2	2	2	72	19	2	2	56	14
ex3	2	2	36	10	2	2	20	5
ex4	6	9	21	14	6	9	21	14
ex5	2	2	32	9	2	2	16	4
ex6	5	8	34	8	5	8	34	8
ex7	2	2	36	10	2	2	16	4
keyb	7	2	170	19	7	2	170	19
kirkman	12	6	370	16	12	6	370	16
lion9	2	1	25	9	2	1	16	4
mark1	5	16	22	15	5	16	180	12
opus	5	6	22	10	5	6	29	9
planet	7	19	115	48	7	19	115	48
prep3	8	8	29	8	8	8	29	8
prep4	8	8	78	16	8	8	78	16

Table 1. Original and state minimized benchmark circuits.

2.3 FPGA State Encoding

The research line described above was targeted to gate arrays or cell-based integrated circuits. FPGA manufacturers and synthesis tools use One-Hot as default state encoding [10], [11]. This assignment allows the designer to create state machine implementations that are more efficient for FPGA architectures in terms of area and logic depth (speed). FPGAs are plenty of registers but the LUTs are limited to few bits wide. One-Hot increases the flip flop usage (one per state) and decreases the

width of combinatorial logic. In addition, the Hamming distance of One-Hot encoding is always two in spite of the machine size. It make easy to decode the next state, resulting attractive in large FSMs. However, a better implementation of small machines can be obtained using binary encoding.

4. Experiments

In this paper, each circuit was encoded in four ways: binary, One-Hot, Two-Hot, and a style proposed by JEDI [5], named “out-oriented” in this paper. This last algorithm uses a binary state encoding that minimizes the output logic. Two-Hot reduces flip-flop usage maintaining at the same time easy-decoding characteristic of One-Hot. Binary and “out-oriented” are highly encoded techniques, whereas One-Hot and Two-Hot can be considered sparse encodings.

All the experiments use the MCNC91 benchmark set [12] together with two FSMs extracted from the former PREP consortium [13]. The original MCNC FSMs are defined using the KISS2 format [6]. So, the first step has been to write a KISS format translator into VHDL. It takes the KISS file, infers a Mealy or Moore machine, and finally writes the corresponding code. The program also generates a file containing an entity with the machine, and another with a top-level VHDL code with tri-states buffers in the pads to measure the off-chip current separately.

The benchmark FSMs were first minimized with STAMINA [14]. The number of inputs, outputs, next state rules and states (for both, the original circuit and the minimized one) are presented in Table 1. Then, each description was translated into VHDL. The resulting code was compiled using FPGA Express [15] and Xilinx Foundation tools [16] into a XC4010EPC84-1 FPGA sample. All circuits have been implemented and tested under identical conditions. That is, all the electrical measurements are related to the same FPGA sample, output pins, tool settings, printed circuit board, input vectors, clock frequency, and logic analyzer probes. Random vectors were utilized to stimulate the circuit. At the output, each pad supported the load of the logic analyzer, lower than 3pf [17].

The circuits were measured at 100 Hz, 2MHz, and 4 MHz to extrapolate the static power. All prototypes include a tri-state buffer at the output pads to measure the off-chip power [18]. Other alternatives to measure power are reviewed in [19][20].

5. Experimental Results

Table 2 shows the area, delay and power obtained for each benchmark circuit. Area is expressed in CLBs, but the number FF utilized is also indicated. The delay, expressed in ns, corresponds to the critical path. Finally, the dynamic power is shown in mW/MHz.

Power Saving: Fig. 1 points out the power saving comparison: (a) OH (One-Hot) vs. binary encoding and (b) OH vs. “out-oriented”. Positive values indicate power reduction obtained using OH encoding. The x axis represents the number of states for the FSM. The figure can be separated in three zones. For machines with up to eight

states, binary encoding must be utilized to reduce power. For machines with more than 16 states always OH is the best choice. Finally, between 8 and 16 states, there is not clear the relation, but “out-oriented” is better than pure binary. On the other hand, TH (Two-Hot) encoding consume more than OH in almost all cases, but it is better than “out-oriented” and Binary for big FSMs.

Circuits	FSM characterist.				Area Bin		Area OH		Area Out-O		Area T-H		Delay (ns)				Power mW/MHz			
	inputs	outputs	rules	states	CLBs	FF	CLBs	FF	CLBs	FF	CLBs	FF	Bin	OH	Out-O	T-H	Bin	OH	Out-O	T-H
bbara	4	2	42	7	11	3	8	7	10	3	15	4	30.0	25.6	29.4	31.2	1.39	1.38	1.87	1.46
bbsse	7	7	208	13	36	4	26	13	27	4	36	5	43.1	36.2	34.6	40.1	4.02	3.37	3.14	3.43
bbtas	2	2	24	6	4	3	4	6	3	3	4	3	16.8	12.7	16.7	15.5	1.08	0.95	0.77	0.97
beecoun	3	4	20	4	7	2	10	4	7	2	12	4	21.1	18.6	16.4	28.9	1.33	1.62	1.33	2.36
cse	7	7	91	16	52	4	42	16	48	4	53	5	54.9	39.1	47.4	47.9	3.73	3.50	2.99	3.83
dk14	3	5	56	7	27	3	26	7	25	3	27	4	34.1	32.5	31.7	37.8	4.15	3.88	4.08	3.92
dk15	3	5	32	4	18	2	20	4	20	2	20	4	29.2	28.2	25.6	32.8	3.32	3.02	3.28	3.85
dk16	2	3	108	27	59	5	31	27	50	5	57	7	52.1	35.0	43.3	44.0	8.09	3.73	6.67	6.64
dk17	2	3	32	8	12	3	10	8	13	3	14	4	24.2	27.8	27.3	24.5	2.30	1.94	2.27	2.28
dk27	1	2	14	7	3	3	4	7	3	3	4	4	12.6	20.2	18.6	18.8	0.88	1.08	0.95	1.36
dk512	1	3	30	15	14	4	10	14	9	4	16	5	20.8	20.4	26.0	23.9	2.46	1.54	1.85	2.48
ex2	2	2	56	14	21	4	17	11	12	4	22	5	31.0	21.3	24.4	27.4	3.60	2.03	1.88	3.23
ex3	2	2	20	5	6	3	8	5	7	3	7	3	19.2	18.1	16.7	13.7	1.38	1.52	1.51	1.44
ex4	6	9	21	14	22	4	15	14	19	4	18	5	31.2	29.4	27.0	27.2	2.51	1.66	2.10	2.11
ex5	2	2	16	4	1	2	5	4	4	2	7	4	8.8	20.1	17.7	25.8	0.55	1.26	0.98	1.39
ex6	5	8	34	8	34	3	28	8	29	3	35	4	40.0	31.4	33.6	47.6	4.25	3.59	3.71	4.86
ex7	2	2	16	4	2	2	5	4	2	2	7	4	10.2	14.5	9.5	18.3	0.62	1.16	0.64	1.49
keyb	7	2	170	19	57	5	42	19	50	5	53	6	58.1	41.7	54.9	62.3	6.55	5.05	4.43	6.02
kirkman	12	6	370	16	45	4	43	16	45	4	57	5	38.3	36.2	38.9	36.6	4.14	4.00	3.73	5.21
lion9	2	1	16	4	2	2	2	4	2	2	5	4	8.8	15.1	8.8	25.5	0.44	0.54	0.43	1.04
mark1	5	16	180	12	19	4	15	12	17	4	17	5	30.2	24.6	24.1	30.5	2.50	1.79	2.11	2.41
opus	5	6	29	9	23	4	15	9	20	4	18	4	31.1	33.0	27.8	28.1	2.95	1.74	2.16	2.45
planet	7	19	115	48	113	6	65	48	106	6	99	10	60.6	41.3	54.3	61.1	14.4	6.23	13.2	11.7
prep3	8	8	29	8	13	3	14	8	12	3	18	4	33.3	26.9	26.5	30.9	1.66	2.04	1.42	1.99
prep4	8	8	78	16	39	4	37	16	35	4	41	5	45.9	31.4	41.5	37.7	5.47	5.29	4.37	4.92

Table 2. Area, Time and Power for the benchmark set.

States-Power relationship: For any state encoding, the power is linearly correlated with the number of states. The coefficient R^2 for the different regression analysis is over 0.85 (Fig. 2). Power is even more correlated ($R^2 \cong 0.87$) respect to $n+i$ (number of states plus number of inputs).

States-Area relationship: In this case, the correlation is similar to the previous analysis, with a $R^2 \cong 0.80$.

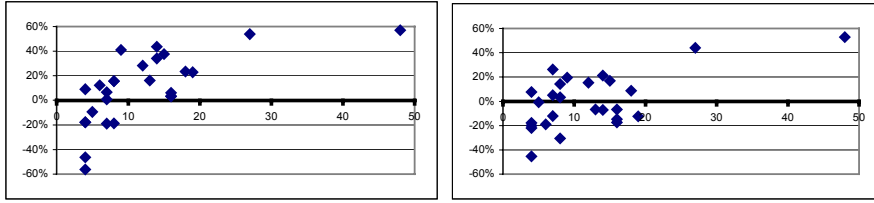


Fig. 1. Power Saving on account of state encoding. a) One-Hot versus Binary. b) One-Hot versus Out-oriented

Time-Power: The relationship is shown graphically in Fig. 4. The linear correlation is $R^2 \cong 0.7$. The experiments do not follow the FPGA rule-of-thumb that indicates that faster circuits consume less power .

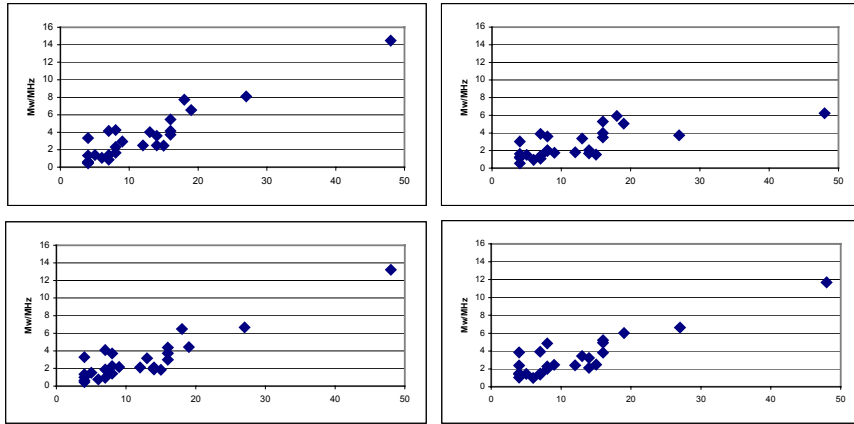


Fig. 2. Power per FSM states: a) Binary; b) One-Hot; c) Out-oriented; and d) Two-Hot.

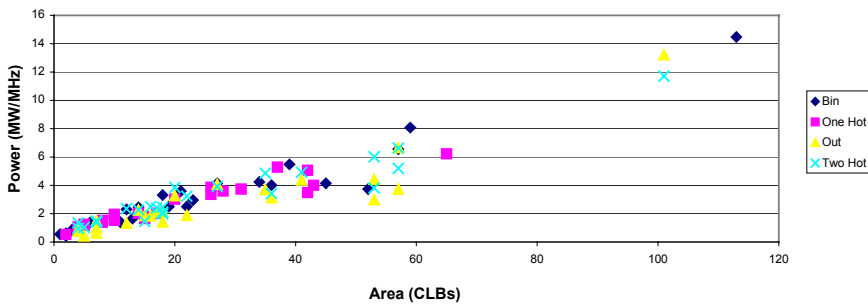


Fig. 3. Area-power relationship

Area-Power: The correlation is important ($R^2 \cong 0.91$) and it can be used as a primary approach to decide for a state assignment. The Fig. 3 represents this distribution. A comparison between area and power shows that the 77% of the benchmark circuits, the smaller circuit consume lower power.

Other correlation like States-Delay are not visible (R^2 lower than 0.6). Area, time and power correlation with the others FSM parameters (inputs, outputs, rules) and combinations of this parameters, neither produce significant results.

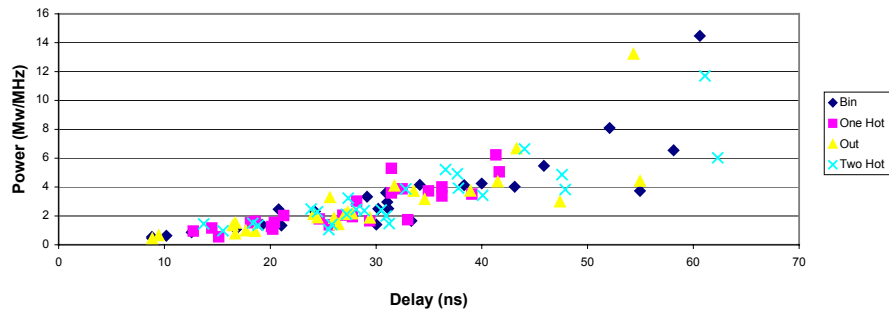


Fig. 4. Delay-Power relationships.

6. CONCLUSION

This paper has presented an analysis of the state encoding alternatives for FSMs. The main conclusions are that in small state machines (up to 8 states), area, speed and power is minimized using binary state encoding. On the contrary, One-Hot state encoding is better for large machines (over 16 states). A comparison between 26 test circuits shows important differences in power consumption. Depending on the state encoding, reaching up to 57% of power saving can be obtained. The Two-Hot approach do not offer advantages over One-Hot, nevertheless it is better than binary for big FSMs. The Out-oriented is a binary encoding that's minimize the decode logic and its in average better than pure binary. Finally, a clear area-power relationship exists. It can be used to estimate power during the design cycle using the information provided for the synthesis tool.

Acknowledgments

Ministry of Science of Spain, under Contract TIC2001-2688-C03-03, has supported this work. Additional funds have been obtained from Projects 658001 and 658004 of the *Fundación General de la Universidad Autónoma de Madrid*.

References

- [1] S. Lopez-Buedo, J. Garrido and E. Boemo, *Thermal Testing on Reconfigurable Computers*, *IEEE Design & Test of Computers*, pp.84-90, January-March 2000.
- [2] X. Wu, M. Pedram, and L. Wang, Multi-code state assignment for low power design, *IEEE Proceedings-Circuits, Devices and Systems*, Vol.147, No.5, pp.271-275, Oct. 2000.
- [3] T.Villa, A.Sangiovanni-Vincentelli, "NOVA: State assignment for finite state machines for optimal two-level logic implementation", *IEEE TCAD*, Vol.9-9, pp.905, Sept. 1990.
- [4] Devadas, S., Ma, H., Newton, A., and Sangiovanni-Vincentelli, A. 1988. MUSTANG: State assignment of finite state machines targeting multilevel logic implementations. *IEEE Trans. Computer-Aided Design* 7, 12 (December), 1290-1300.
- [5] B. Lin and A.R. Newton. Synthesis of Multiple Level Logic from Symbolic High-Level Description Languages. In Proc. of Internat. *Conf.on VLSI*, pages 187–196, August 1989.
- [6] E. Sentovich, K. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, P. Stephan, R. Brayton, and A. Sangiovanni-Vincentelli. SIS: A System for Sequential Circuit Synthesis. *Tech. Report Mem. No. UCB/ERL M92/41*, Univ. of California, Berkeley, 1992.
- [7] C-Y Tsui, M. Pedram, A. M. Despain, Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs, *31st Design Aut. Conf.*, pp. 18-23, 1994.
- [8] L.Benini and G. De Micheli. State Assignment for Low Power Dissipation. *IEEE Journ. of Solid State Circuits*, Vol. 30, No. 3, pp. 258-268, March 1995.
- [9] Winfried Nöth and Reiner Kolla. Spanning Tree Based State Encoding for Low Power Dissipation. In *Proc of Date99*, pp 168-174, Munich, Germany, March 1999.
- [10] Xilinx software manual, Synthesis and Simulation Design Guide: Encoding State. Xilinx inc, 2000
- [11] FPGA Compiler II / FPGA Express VHDL Reference Manual, Version 1999.05, Synopsys, Inc., May 1999
- [12] Bob Lisanke. "Logic synthesis and optimization benchmarks". *Technical report, MCNC, Research Triangle Park*, North Carolina, December 1988.
- [13] PREP Benchmarks (Programmable Electronics Performance Company), see: <http://www.prep.org>.
- [14] G.D. Hachtel, J.-K. Rho, F. Somenzi, and R. Jacoby. Exact and Heuristic Algorithms for the Minimization of Incompletely Specified State Machines. In *Proc. of the European Conference on Design Automation*, pages 184–191, Amsterdam, Holland, Feb 1991.
- [15] FPGA Express home page. Synopsys, inc.; http://www.synopsys.com/products/fpga/fpga_express.htm
- [16] Xilinx Foundation Tools F3.1i, information available at www.xilinx.com/support/library.htm
- [17] Tektronix inc., "TLA 700 Series Logic Analyzer User Manual, available at <http://www.tektronix.com>.
- [18] E. Todorovich, G. Sutter, N. Acosta, E. Boemo and S. López-Buedo, End-user low-power alternatives at topological and physical levels. Some examples on FPGAs, *Proc. DCIS'2000*, Montpellier, France, November 2000.
- [19] J. Alcalde, J. Rius and J. Figueras, Experimental techniques to measure current, power and energy in CMOS integrated circuits, *Proc. DCIS'00*, Montpellier, France, Nov. 2000.
- [20] L. Mengibar, M. García, D. Martín, and L. Entrena, Experiments in FPGA Characterization for Low-power Design, *Proc. DCIS'99*, Palma de Mallorca, 1999.
- [21] Chi-Ying Tsui, Massoud Pedram, Chih-Ang Chen, and Alvin Despain, Low Power State Assignment Targeting Two- and Multi-level Logic Implementations, *Proceedings of ACM/IEEE International Conf. of Computer-Aided Design*, pp. 82-87, November 1994
- [22] M. Martínez, M. J. Avedillo, J. M. Quintana, M. Koegst, ST. Rulke, and H. Susse: Low Power State Assignment Algorithm, *Proc. Design of Circuits and Integrated Systems Conf. (DCIS'00)*, pp. 181-187, 2000.