Synthesis of Low Power Output Direct Synchronous Finite State Machines

Duarte Lopes de Oliveira[1]   *Leonardo Romano*[,2]
duarte@ita.br   leoroma@uol.com.br
[1]Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – IEEA – ITA
Praça Marechal Eduardo Gomes, 50 – CEP 12228-900 – São José dos Campos – São Paulo– Brazil.
[2]Departamento de Engenharia Elétrica do Centro Universitário da FEI
São Bernardo do Campo – São Paulo – Brazil.

Abstract

The reduction of the energy consume is one of the most important tasks in the contemporary project of digital circuits. The methods that are proposed in the literature for the synthesis of low power synchronous finite state machines (LP-SFSM) damage the area and most of all, the cycle time.
In this article we propose a method for LP-SFSM synthesis with high performance in cycle time and low penalty in area. Our method eliminates all the glitches in the output and state signals, reduces the glitches generated by non-monotonic behavior input signals and eliminates all the dynamic power consume in the state transitions where there is no change of value in the output and state signals. Our method uses the output signals as state signals, proposes the gRS flip-flops and one algorithm of logic minimization related to low power.

Keywords: low power finite state machines, low power logic synthesis, non-conventional flip-flops and low power logic minimization.

1. Introduction

The reduction of the energy consumption is one of the most important tasks in the projects of digital circuits. It is due to the increasing of portable applications, so as: laptops, notebooks and communication (cellular, pager and so on) [2]. These portable appliances require long life batteries with a low consume power. Traditionally the digital circuits are introduced with components that are constructed with the CMOS technology. The power sources dissipated in the CMOS components are briefly provided in the following expression:

$$P_{total\text{-}average}=1/2.C.V^2_{DD}.f.N +Q_{sc}.V_{DD}.f.N + I_{leakage}.V_{DD} \quad (1)$$

Where: $P_{total\text{-}average}$ denotes a total average power, V_{DD} is the supply voltage, and f is the operation frequency. The first term represents the dynamic dissipated power. The second term represents the dissipated power related to the short current. (the current flow from the source to the ground when there is an output transition). The third term represents the static dissipated power related to the leakage current. C represents the capacitances. The Q_{SC} factor represents the quantity of load carried by the short circuit current by transition. The N factor is the switching

activity, that is, the number of transitions in the output gate through clock cycle.

In the independent level of the technology the most representative techniques of reduction of dynamic power in the synthesis of finite states machines (FSM) are being proposed in the logic level, which are: clock logic control (*gate-clocked*), decomposition, states assignment and logic minimization [3,4,5,7,11].
The different strategies for the reduction of the dissipated power start from the models of Moore or Mealy machine and from traditional target architecture that uses a logic block of excitation and flip-flops (FF) memory elements. We believe in a general pattern, that the models of Moore and Mealy machine and the usage of the conventional FFs as memory elements are not the most convenient for the low power FSMs.

1.1. Direct output FSMs

Forrest [8] and Koegst [9,10] describe a new type of FSM known as direct output FSM where the output signals are used as state signals. For this type of machines, the models are called direct Moore and direct Mealy models. There are four advantages in using the output signals as state signals:
1) Reduction or elimination of the state variables so we can have an area reduction; **2**) At the classical execution of the Moore model machines there are three blocks (excitation logic, flip-flops and output logic), but in the direct Moore model machines there are only two blocks (excitation logic and flip-flops), therefore there is a reduction of the cycle time (increasing of the clock rate); **3**) The output signals are free of glitches, then there is a reduction in the switching activity and they can be used to activate counters and registers (*Datapath*); **4**) Reduction or removal of the state variables increase the observability and the controllability of the circuit, then it makes the testability easy.
The methods for the synthesis of direct output FSMs are interested only in the stage of the state assignment, which is to find the least number of state variables that must be inserted.

In this article we are proposing a new method of synthesis of direct output FSMs. That besides eliminating the

consumption of dynamic power where the outputs and the state signals don't change their values, eliminates the glitches produced in those signals and also reduces the glitches produced by the input signals. Our method starts from the state transition graph specification (STG) and generates the optimized logic circuit in dissipated dynamic power, area and cycle time.

Different from all the methods presented before the new method acts in several stages of the low power logic synthesis (architecture, machine model and logic minimization) and not in a single stage. It generates direct output FSMs with a superior performance in the consumption of dynamic power and clock rate when compared to the FSMs generated by the traditional methods [1].

What remains in this article is organized in the following way. In the section 2 we present our architecture of low power; in section 3 a new notation for the STG specification; in section 4 the algorithm of logic minimization of low power; in section 5 the procedure of synthesis of low power FSM; in section 6 we analyze the performance of our circuits.

2. Target architecture

The direct Moore model is developed in our target architecture. To obtain some optimization in dynamic power consume and cycle time we propose the target architecture called generalized RS flip-flop (gRS FF - see figure 1) and the algorithm of logic minimization concerning the reduction of the dissipated dynamic power. The direct output FSMs produced by our method has two important characteristics: 1) The excitation logic and the memory element are fused in the gRS FF; 2) The two feedback of the gRS FF stop (dissipated dynamic power turned to zero) either the block of the F_{SET} function or the block of the F_{RESET} function (see figure 1).

The gRS FF is structured as a master-slave. The master is the gRS latch with two feedback triggered in the high level. The slave is SR latch triggered in the low level. The project of two latches was elaborated in order to eliminate all the dissipation of dynamic power caused by the clock signal where there is no change of output or state signal. Figure 2 shows the full custom of SR latch. In [13] the gRS FF triggered in two edge of the clock signal that is used for the increasing of the circuit speed or the reduction of the clock rate to decrease the power consume, but with equivalent speed.
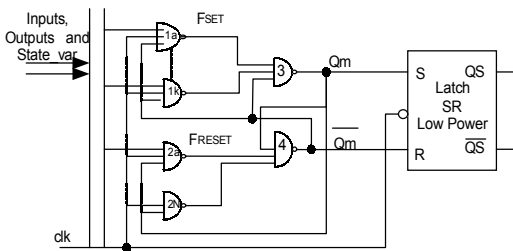

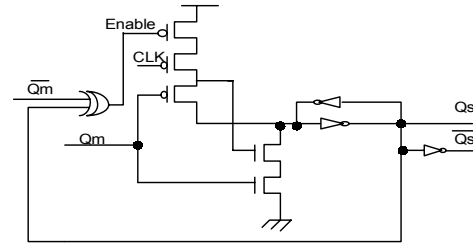
Figure 1 – Master-slave FF gRS.

Figure 2 – Complex LP-SR Latch.

2.1. Timing analysis

The times of *setup, hold, latency and cycle* are four important temporal parameters interacting with memory elements: setup time is the minimum interval between the stabilization of the input signals and of states and the transition of the clock signal. Latency time is the minimum interval between the falling edge transition of the clock signal and the change of the output value of the Q_S output. Cycle time is the minimum interval between the rising transition of the clock signal and the change of the values of the input signals in the next state transition of the STG. The temporal conditions of the interaction of external environment with FF gRS (see figures 1 and 3) are supplied through the following temporal equations, noticing that T is the delay of the logic latch[1] or the logic gate.
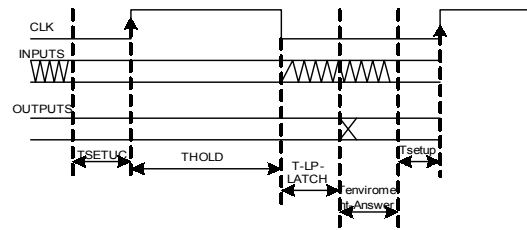

Figure 3 – gRS FF temporal conditions.

The setup time is:
$$T_{SETUP} \geq T_{MAX-NAND-1a} + T_{MAX-NAND-2} + T_{MAX-NAND-3} \qquad (2)$$
The minimum width of the high pulse of the clock is:
$$T_{HIGH-PULSE-WIDTH} \geq T_{SETUP} \qquad (3)$$
The hold time is:
$$T_{HOLD} \geq T_{HIGH-PULSE-WIDTH} - T_{SETUP} \qquad (4)$$
The minimal width of the clock low pulse is:
$$T_{LOW-PULSE-WIDTH} \geq T_{MAX-LATCH-D} + T_{ENVIRONMENT-ANSWER} \qquad (5)$$
Where: $T_{ENVIRONMENT-ANSWER}$ is the maximum time for the input signals activation in a state transition.
The minimum latency time is:
$$T_{LATÊNCIA} \geq T_{SETUP} + T_{MIN-LATCH-D} \qquad (6)$$
The minimum cycle time is:
$$T_{CYCLE} \geq T_{HIGH-PULSE-WIDTH} +$$

---

[1] To simplify the inequations we assume a zero delay in the architecture wires.

$$T_{LOW\text{-}PULSE\text{-}WIDTH} \qquad\qquad (7)$$

The maximum clock frequency is:

$$F_{MAX} \le 1 / T_{CYCLE} \qquad\qquad (8)$$

3. State Transition Graph Specification

We propose a new notation for the STG and state transition table (STT – [13]) to facilitate the specification of direct outputs FSM model direct Moore. In the direct Moore STG, the node represents the state, the number of occurrence of the code and the output code. In figure 4, in the node *A/2/00*, *A* is the state label, *2* means the second occurrence of the output code *XY=00*.
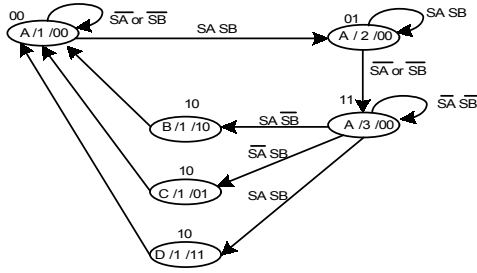

Figure 4 – STG for the direct Moore model.

4. Logic minimization for low power

The problem of minimization of a two level function for a low power must to satisfy the following cost[2] function: For an *f* function with the $E=(e_1,...,e_N)$ input combination find a two level implementation (F_{SET} and F_{RESET} do FF gRS) of the *f* function (f_{SET} and f_{RESET}) so that in each STT state transition either no product-cube is activated or only a number minimum of product cube are activated. Our Min_BP algorithm follows the steps of the *Quine-Mccluskey* algorithm in order to extract for each non-input signal (output signals or if there is any state signals) the F_{SET} and F_{RESET} functions of two levels sum-of-products FF gRS [12,13].

Covering Condition:
Each product that belongs to F_{SET} or F_{RESET} function must satisfy the lemma 4.1 or 4.2 and these functions must satisfy theorem 4.1

Lemma 4.1 (without proof). Considering the *fx* Boolean function of the non-input signal *x* and $F_{SET\text{-}X}$ of the FF gRS that corresponds to the implementation in form of sum-of-products of *fx*. Considering the state transition $t \in$ STT where the x signal is activated 0→1 and covered by the required cube CR_t. $F_{SET\text{-}X}$ dissipates minimum dynamic power if and only if there is a single product-cube $p_i \in F_{SET\text{-}x}$ so that $p_j \in F_{SET\text{-}x}$ that completely covers CR_t, and if there is a product-cube $p_i \cap p_j \ne \varnothing$ then the sharing of the states is don't care.

Lemma 4.2 of the function F_{RESET} is similar to lemma 4.1

Theorem 4.1 (proof [13]): The covering of the $F_{SET\text{-}x}$ ($F_{RESET\text{-}x}$) dissipates minimum dynamic power if and only if:

- Each required cube from *fx* is completely covers in a single $F_{SET\text{-}x}$ ($F_{RESET\text{-}x}$) product.
- All the $p_i \in F_{SET\text{-}x}$ ($F_{RESET\text{-}x}$) products satisfies Lemma 4.1 (4.2).
- The literals of each $p_i \in F_{SET\text{-}x}$ ($F_{RESET\text{-}x}$) product have a minimum probability of activated.

5. Output direct FSM Synthesis

Our method follows three steps:

1. To Generate STG for direct Moore model. Let K the largest number of occurrences of an output code in STG. If K>1 then step 2, otherwise {Generating without conflicts STT and to go step 3}
2. To codify Y state variable, where Y=⌈log2 K⌋, generating coded STT [13].
3. For each output and state signal of STT to obtain the minimized excitation equations for the FF gRS [13].

5.1. Study case

The example of the figure 4 will be used to illustrate our method. The first step verifies if the STT has any conflict. Since the state *A* occur three times maximum in the STT (*XY=00*), then, it is necessary to insert two state variables (*Y1,Y2*) to eliminate conflicts (two or more states with the same output code).

Step 2 generated codified STT, and then it is free of conflicts as it is shown in figure 5. The rows in the STT of figure 5 describe the state code (outputs plus state variables). Step 2 also extracts the required cubes. In the figure 5 the required cubes for the Y_{1SET} function are $SASBY_1Y_2XY=[210100,100100]$, where 2 signifies don't-care.

Step 3 corresponds to the logic minimization.

$$Y_{1SET} = SA' Y_2 + SA\ SB' Y_2$$

$$Y_{1RESET} = Y_2'$$

$$Y_{2SET} = SA\ SB\ Y_1'$$

$$Y_{2RESET} = SB\ Y_1 + SA\ SB' Y_1$$

$$X_{SET} = SA\ Y_1\ Y_2$$

$$X_{RESET} = Y_2'$$

$$Y_{SET} = SB\ Y_1\ Y_2$$

$$Y_{RESET} = Y_2'$$

Y1 Y2 X Y \ SA SB	00	01	11	10
0 0 0 0	0000	0000	0100	0000
0 1 0 0	1100	1100	0100	1100
1 1 0 0	1100	1001	1011	1010
1 0 1 0	0000	0000	0000	0000
1 0 0 1	0000	0000	0000	0000
1 0 1 1	0000	0000	0000	0000

Figure 5 – Conflicts free coded STT for the direct Moore model.

6. Results and discussion

First of all we have discussed the several advantages that the FF gRS have when we decided in implementing low power FSM. These advantages are obtained due to the three characteristic of this architecture, which are: a) non-conventional FF; b) latch type; c) feedback.

Table 1 show small-sized controllers obtained in the literature where they were executed by our method and by the traditional method [1]. The resultant circuits were mapped at the IMEC-96 standard cell library of 0,7 μm, where the propagation time of our RS full custom latch was estimated in 1,25ns. In these 11 examples our method obtained a medium reduction of 27% (without feedback) and 14% (with feedback) in the cycle time when compared to the classical method. Table 2 shows for the same examples the result in area (number of transistors). Our method generated circuits with a medium penalty of 15% (without feedback) and 25% (with feedback) when compared to the classical method. From 11 examples our method obtained a reduction in area in 3 examples without feedback and 2 examples with feedback. The reason was the common technological mapping.

	Input/ Output	State / transitions	Our Method		Traditional Method
			Without feedback	With feedback	
			Time of cycle (ns)	Time of cycle (ns)	Time of cycle (ns)
Auto alarm	3/1	3/7	3,21	3,21	4,30
RailRoad/Highway	2/1	4/9	3,21	3,21	4,68
Traffic-Light-1	2/2	4/6	3,21	3,21	4,58
Pulsos Train	2/2	3/6	2,56	3,08	4,41
Display	3/1	4/7	2,69	3,21	4,64
TrafficTalker	2/2	4/8	2,84	2,84	4,82
PLL	2/2	5/10	3,21	3,21	5,10
Traffic-Light-2	2/3	6/12	2,97	2,97	5,10
Sealt belt Alarm	5/3	3/7	3,83	3,83	5,82
Traffic Light-3	3/2	4/8	2,69	3,21	4,54
Figure 4	2/2	6/11	3,15	3,51	5,15
Total	——	——	38,70	46,30	53,15

Table 1 – Results in cycle time.

	Input/ Output	State/ transitions	Our Method		Traditional Method
			Without feedback	With feedback	
			Nr. of transistor	Nr. of transistor	Nr. of transistor
Auto alarm	3/1	3/7	90	104	74
RailRoad/Highway	2/1	4/9	131	153	98
Traffic-Light-1	2/2	4/6	165	187	90
Pulsos Train	2/2	3/6	80	92	90
Display	3/1	4/7	96	116	80
TrafficTalker	2/2	4/8	121	133	102
PLL	2/2	5/10	206	234	178
Traffic-Light-2	2/3	6/12	133	159	166
Sealt belt Alarm	5/3	3/7	245	253	146
Traffic Light-3	3/2	4/8	92	110	146
Figure 4	2/2	6/11	200	210	156
Total	——	——	1559	1751	1326

Table 2 – Results in area.

7. Conclusion

In this article we have discussed several techniques used in the FSM logic synthesis. We believe that the Moore and Mealy model machines and architectures based on conventional FF are not the most designate for low power FSMs. In this article we have presented a method for direct output FSMs that are implemented in the target architecture based on non-conventional FFs (gRS FF). Our method synthesizes synchronous machines that reduce the generation of glitches in the state transitions where the outputs and state signals don't change their value and don't consumption dynamic power. This result is achieved through the two contributions, the FF gRS and the logic minimization. Our FSMs have better performance in the low power consumption and cycle time when compared to the FSMs generated by the traditional methods. For future works insert in the logic minimization algorithm the selection of the literals with the last switching probability. Adapt for the direct output FSMs one states assignment algorithm that codifies the states with the least switching cost, and accomplish an estimation of the power consumed by our controller and the low power controllers of the literature.

References

[1] R. H. Katz, Contemporary Logic Design, The Benjamin/ Cummings Publishing Company, Inc., 2ª edition 2003.

[2] S. Devadas and S. Malik, A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. Proc. 32nd ACM/IEEE DAC, pp.242-247, 1995.

[3] 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.

[4] Luca 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.

[5] M. Koegst, et al. Low Power Design of FSMs by State Assignment and Disabling Self-Loops, Proc. 23rd EUROMICRO "new Frontiers of Information Technology", pp. 323-330, 1997

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

[7] S. Chattopadhyay, et al. State Assignment and Selection of Types and Polarities of Flip-Flops, for Finite State Machine Synthesis, IEEE India Conf. (INDICON), pp.27-30, 2004.

[8] Forrest, ODE: Output Direct State Machine Encoding. Proc. Euro-DAC with Euro-VHDL, pp.600-605, 1995.

[9] 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.

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

[11] S. Roy and P. Bonerjee, Resynthesis of Sequential Circuits for Low Power. Proc. IEEE Int. Symposium on Circuits and Systems – ISCAS, pp.VI-57-61, 1998.

[12] 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.

[13] D. L. Oliveira et al. Synthesis of Low Power Synchronous Finite State Machine using Latches, Relatório Técnico – Instituto Tecnológico de Aeronáutica – Divisão de Engenharia Eletrônica – ITA – 2006.