AUTOMATIC SYNTHESIS OF LOW POWER OUTPUT DIRECT SYNCHRONOUS FINITE STATE MACHINES

Leonardo Romano
Departamento de Engenharia Elétrica do Centro Universitário da FEI
São Bernardo do Campo – São Paulo – Brasil
leoroma@uol.com.br

Duarte Lopes de Oliveira
Divisão de Engenharia Eletrônica do Instituto Tecnológico de Aeronáutica – IEEA – ITA
duarte@ita.br

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 and latency times. Our method eliminates all the glitches in the output 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 low power SR flip-flops and one algorithm of logic minimization related to low power.

Keywords. Low power finite state machines, low power logic synthesis, low power flip-flops, 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) (Devadas, 1995). These portable appliances require long life batteries with a low consume power. Two strategies are used in the low power project (Devadas, 1995; Monteiro, 1998). One strategy tries to reduce the power consume through refinement of the project process. Other strategy is to modify the processing technology. Several methods to decrease the power consume have been proposed on the three levels of the project process; architecture (Monteiro, 2002; Liu, 2005), logic (Koegst, 1997; Cao, 2004; Iman, 1995) and physic (Strollo, 1999). 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:

\[ P_{\text{total-average}} = \frac{1}{2}C.V_{DD}^2f.N + Q_{ac}.V_{DD}.f.N + I_{\text{leakage}}.V_{DD} \]  \hspace{1cm} (1)

Where: \( P_{\text{total-average}} \) 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 \( Q_{ac} \) 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 static CMOS technologic the largest fraction of the dissipated power occurs during the switching activity of the events (dynamic power) (Strollo, 1999). The dissipation of the medium dynamic power in a gate \( g \) can be simplified to the first term of the equation (1):
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: gated-clock control (GCC) (Benini, 1996), decomposition (Monteiro, 2002; Liu, 2005), states assignment (Koegst, 1997, 1998; Baccheletta, 2000; Chattopadhyay, 2004; Lemberski, 2002) and logic minimization (Iman, 1995; Bahar, 1995; Tseng, 1997; Choi, 1997; Roy, 1998). Now we will discuss the proposals.

- The GCC techniques cause two problems: a) only applied in Moore model FSM that possess many events of the self-loops type; b) The CLC demands some extra circuit which besides increasing the area, consumes power and affects directly the latency time.

- The decomposition techniques causes three problems: a) only applied to large FSM; b) not all FSM have good decomposition; c) decomposition always leads to an increasing of area that consumes power and increases the latency time.

- The low power states assignment techniques causes two problems: a) to create the codification of a FSM with the least number of transitions using the least number of state variables is a NP-hard problem, then the problem demands an heuristic solution; b) the reduction of the number of transitions usually demands a larger number of state variables\(^1\), which leads to an increasing of area that consumes power.

- The low power logic minimization techniques causes two problems: a) the task is NP-complete, then it needs a heuristic solution (sub-optimum) b) reduction of activity of the implicants switching usually leads to an area increasing.

The methods previously presented are related to some stage of logic synthesis of the low power FSM. These methods have some drawback, in the majority of cases they don’t eliminate the consumed of needless dynamic power and produce some circuits with increasing not only in the cycle time but also in the area. 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

Pomeranz (1993), Forrest (1995) and Koegst (1997, 1998) 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 (Pomerantz, 1993). When we compare them to the traditional Moore and Mealy models we have: a) Pomeranz (1993) shows the potential of the FSMs of direct output in achieving reductions in the latency time and area, as the elimination of glitches in the output signals; b) In (Valeri, 2001) show that the direct output FSMs occupy a much less number of PLDs macrocells.

In the direct Moore machines the code of the output signals coincides with the state code. In the direct Mealy machines the code of output signals are the same of the next state code. 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 presented 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. These methods are more interested in the area reduction, and the reduction of the dissipated dynamic power is poor.

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 times of cycle and latency.

\[ P_{\text{average}} = \frac{1}{2} C_g V^2 D_{\text{FF}} f N_{\text{average}} \]
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 (McCluskey, 1986; Katz, 2003).

What remains in this article is organized in the following way. In section 2 we present our architecture of low power; in section 3 the state transition graph specification for output direct FSM; in section 4 the procedure of synthesis of low power FSM; in section 5 the algorithm of logic minimization of low power; in section 6 we illustrate our method with an example; in section 7 we analyze the performance of our FSMs.

2. Target architecture

The direct Moore model is developed in our target architecture. To obtain some optimization in dynamic power consumes and cycle time we propose the target architecture called feedback standard SR (see fig. 1) that use low power SR Flip-Flop (LP-SR FF). The direct output FSMs produced by our method has one important characteristic: The two feedback of the standard SR architecture stop (dissipated dynamic power turned to zero) either the block of the \( F_{SET} \) function or the block of the \( F_{RESET} \) function (see fig. 1).

The LP-SR FF is structured as a master-slave. The master is the SR latch 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. Figures 2 and 3 respectively show the state transition graph and full custom of LP-SR FF.

![Figure 1 – Target architecture: feedback standard SR.](image1)

![Figure 2 – State transition graph: Master-Slave SR Flip-Flop.](image2)

![Figure 3 – Full custom LP-SR Flip-Flop.](image3)

3. State transition graph specification

We propose a new notation for the state transition graph (STG) and state transition table (STT) 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 fig. 4, in the node \( A/2/00 \), \( A \) is the state label, 2 means the second occurrence of the output code \( XY=00 \).

The STT can be generated either directly or from STG where the rows represent the combination of the values of the input signals and the lines represent the states with the respective outputs. Figure 5 shows the STT of the STG of fig. 4.
4. Output direct FSM synthesis

Our method follows three steps:

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}

To codify Y state variable, where \( Y = \lceil \log_2 K \rceil \), generating coded STT (Forrest, 1995).

For each output and state signal of STT to obtain the low power minimized excitation equations for the feedback SR standard architecture (see section 5).

In section 6 we illustrate our method using the STG of fig 4.

5. 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 \( (F_{\text{SET}} \text{ and } F_{\text{RESET}} \text{ of the LP-SR FF}) \) of the \( f \) function \( (f_{\text{SET}} \text{ and } f_{\text{RESET}}) \) so that in each STT state transition either no product-cube is activated or only a number minimum of product cube are activated.

To obtain the minimal switching activity of the \( F_{\text{SET}} \) and \( F_{\text{RESET}} \) excitation functions from the SR standard architecture, the concept of regions will be used at the stage of logical minimization.

5.1. STG Regions

In the STG, state transitions that share some common characteristics can be bundled together to inform regions. According to the region-based theory there is a link between the functionality of these regions and physical gates implementing it (the circuit). The instability or stability of non-input signals characterizes two important regions. They are called unstable and stable regions, respectively, in the STG.

\(^2\) The purpose is to reduce the dissipated dynamic power, but we assume that the probability of activation of each input signal in each state transition is the same (Devadas, 1995; Monteiro, 1998).
Definition 5.1 An Unstable Region of a non-input signal $X$ in a STG is a set of unstable intermediary states, generated by a state transition in which the signal $X$ is excited ($0 \rightarrow 1$ or $1 \rightarrow 0$).

An unstable region will be denoted as $UR[*X,Nr]$, where $X$ is the transition value + or – and Nr is the number of the unstable region. The region $UR[*X,Nr]$ is covered by a cube or more cubes, here denominated required cube.

Definition 5.2 A Stable Region of a non-input signal $X$ is the largest set of connected state transitions (arcs) which are directly reached from $UR[*X,Nr]$ and $X$ is not excited, i.e., $X$ does not change its value ($1 \rightarrow 1$ or $0 \rightarrow 0$).

A stable region will be denoted as $SR[*X,Nr]$, where $*X$ has the constant value 0 or 1 and Nr is the number of the stable region. Figure 6 shows all regions of the output signal $Y$ of the STG of fig. 4.

Figure 6 – STG regions: $UR[+Y,1]$, $UR[-Y,1]$ and $SR[-Y,1]$

Covering Condition:
Each product that belongs to $F_{SET}$ or $F_{RESET}$ function must satisfy the lemma 5.1 and these functions must satisfy theorem 5.1

Lemma 5.1 (without proof) A low power cube or low power prime implicant of an unstable region, $UR[*X,i]$, is a cube with minimal switching activity if and only if:

- It completely covers all the states of the required cube $\supseteq UR[*X,i]$;
- It changes its value at most once in $UR[*X,i] \cup SR[*X,i]$;
- If it covers some state of the required cube $\supseteq UR[*X,j\neq i]$, then it must completely cover all states of the required cube $\in UR[*X,j]$; and
- If there is a product-low-power-cube $p_i \cap p_j \neq \emptyset$ then the sharing of the states either $\in UR[*X,i]$ or are non reachable states.

From each unstable STG region one or more required cubes is extracted. One required cube is a cube that covers the maximum of states that belong to an unstable STG region. Figures 7a,b shows in the STT the required cubes of the $UR[+Y,1]$: $SASBY_1Y_2=212100$ and $SASBY_1Y_2=122100$ (where 2 is don’t-care).

Figure 7 – State transition $A_2 \rightarrow A_3$: a)STG; b) STT

Theorem 5.1 (without proof): The covering of the $F_{SET-x}$ ($F_{RESET-x}$) dissipates minimum dynamic power if and only if:

- Each required cube from $f_x$ is completely covers in a single $F_{SET-x}$ ($F_{RESET-x}$) product.
- All the $p_i \in F_{SET-x}$ ($F_{RESET-x}$) products satisfies Lemma 5.1.
- The literals of each $p_i \in F_{SET-x}$ ($F_{RESET-x}$) product have a minimum probability of activated.
Our LP_MIN 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 $F_{\text{SET}}$ and $F_{\text{RESET}}$ functions of two levels sum-of-products LP-SR FF (Unger, 1995; Nowick, 1995).

**Procedure of the LP_MIN algorithm:** The minimum function of the two-level $F_{\text{SET-X}}$ or $F_{\text{RESET-X}}$ of $f_X$ function is obtained as it follows:

1. Extract from each state transition where $x:0 \rightarrow 1$ ($F_{\text{SET-X}}$) or $x:1 \rightarrow 0$ ($F_{\text{RESET-X}}$) the respective required cubes.
2. Extract the $f_X = \Sigma(M_1, M_2, ..., M_N) + d(M_k, ..., M_y)$ function from the coded direct output STT to $F_{\text{SET-X}}$ or $F_{\text{RESET-X}}$.
3. Extract all the prime implicants that satisfy lemma 5.1 ($F_{\text{SET-X}}$ or $F_{\text{RESET-X}}$), generating the set of low power implicants (S_LPPI).
4. Create the minimum covering that satisfies the theorem 5.1: obtain from S_LPPI in LPPI minimum set that cover all the $M_i \in F_X$ minterms.

6. Study case

The example of the fig. 4 will be used to illustrate our method. The first step verifies if the STT (see fig. 5) 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 ($Y_1, Y_2$) 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 fig. 8. The rows in the STT of fig. 8 describe the state code (outputs plus state variables). Step 3 initially extracts the required cubes. In the fig. 8 the required cubes for the $Y_{\text{SET}}$ function are $\text{SASBY}_1Y_2XY = [210100, 100100]$.

Step 3 corresponds to the logic minimization. Fig. 9 and 10 respectively show the generation of the STTs for the output signal $Y$. The STT cells occupied by the “−” value signify no reachable states. The STT cells occupied by “x” signify states with don’t care value. The minimum low power functions obtained by our algorithm, where the covering satisfying the theorem 5.1 are presented below, and fig. 11 shows the logic circuit.
7. Results and discussion

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

a) LP-SR FF versus conventional FF:
When the conventional FF output doesn’t have a change of value the clock signal transitions dynamically activate several transistors, then dissipates dynamic power. For the same situation the LP-SR FF doesn’t dissipate dynamic power. The usage of LP-SR FF eliminates the necessity of the logic control of the clock signal.

b) Latch type:
The advantages of the master SR latch in the place of the master D latch are the elimination of the covering (two product-cubes) from the non-input signals of the transitions 1 ⇒ 1 and reduction of cubes in the transition 1 ⇒ 0.

c) Feedback:
Due to the non-monotonic performance of the input signals, the two feedbacks (target architecture) are used to stop (propagation of the glitches) one of the two-combinatory blocks that correspond to the $F_{SET}$ and $F_{RESET}$ excitation functions. During the FSM processing on of blocks will always be deactivated, then there will be no dissipation of the dynamic power.

Table 1 shows small-sized controllers obtained in the literature where they were executed by our method and by the traditional method (McCluskey, 1986; Katz, 2003). The resultant circuits were mapped at the IMEC-96\(^3\) standard cell library of 0.7 μm, where the propagation time of our full custom SR latch was estimated in $t_p=1.44$ns and $t_{SETUP}=0.48$ns. In these 11 examples our method obtained a medium reduction of 36% (without feedback) and 32% (with feedback) in the cycle time when compared to the traditional method.

Table 2 shows for the same examples the result in latency time of the traditional method. The latency time of the output directed FSMs is the propagation time of the LP-SR FF ($t_p=1.44$ns). Our method generated circuits with a medium reduction of 37% when compared to the traditional method.

Table 3 shows for the same examples the result in area (number of transistors). Our method generated circuits with a medium reduction of 5% (without feedback) and a penalty of 8% (with feedback) when compared to the traditional method. From 11 examples our method obtained a reduction in area in 8 examples without feedback and 3 examples with feedback.

\(^3\) The traditional method used D FF (26 transistors), where $t_{SETUP}=1.50$ns and propagation time $t_p=1.98$ns.
Table 1 – Results in cycle time

<table>
<thead>
<tr>
<th>Input/Output</th>
<th>Moore state transitions</th>
<th>Mealy state transitions</th>
<th>Traditional Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto alarm</td>
<td>31</td>
<td>34</td>
<td>36</td>
</tr>
<tr>
<td>Traffic light 1</td>
<td>40</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>Traffic light 2</td>
<td>38</td>
<td>40</td>
<td>42</td>
</tr>
<tr>
<td>Traffic light 3</td>
<td>36</td>
<td>38</td>
<td>40</td>
</tr>
<tr>
<td>Traffic light 4</td>
<td>34</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>Total</td>
<td>---</td>
<td>---</td>
<td>32,220</td>
</tr>
</tbody>
</table>

Table 2 – Results in latency time

<table>
<thead>
<tr>
<th>Input/Output</th>
<th>Moore state transitions</th>
<th>Mealy state transitions</th>
<th>Traditional Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traffic light 1</td>
<td>40</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>Traffic light 2</td>
<td>38</td>
<td>40</td>
<td>42</td>
</tr>
<tr>
<td>Traffic light 3</td>
<td>36</td>
<td>38</td>
<td>40</td>
</tr>
<tr>
<td>Traffic light 4</td>
<td>34</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>Total</td>
<td>---</td>
<td>---</td>
<td>23,682</td>
</tr>
</tbody>
</table>

Table 3 – Results in area

<table>
<thead>
<tr>
<th>Input/Output</th>
<th>Moore state transitions</th>
<th>Mealy state transitions</th>
<th>Traditional Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Auto alarm</td>
<td>29</td>
<td>32</td>
<td>36</td>
</tr>
<tr>
<td>Traffic light 1</td>
<td>40</td>
<td>44</td>
<td>46</td>
</tr>
<tr>
<td>Traffic light 2</td>
<td>38</td>
<td>40</td>
<td>42</td>
</tr>
<tr>
<td>Traffic light 3</td>
<td>36</td>
<td>38</td>
<td>40</td>
</tr>
<tr>
<td>Traffic light 4</td>
<td>34</td>
<td>36</td>
<td>38</td>
</tr>
<tr>
<td>Total</td>
<td>---</td>
<td>---</td>
<td>19,19</td>
</tr>
</tbody>
</table>

8. 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 low power FFs (LP-SR 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 LP-SR FF 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 FSMs and the low power FSMs of the literature.

9. Acknowledgment

We thanked to the prof. Alexis of the department of applied electronics of ITA that aid us in the simulation of the circuits.

10. References


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


