



Contents lists available at ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi



Genetic algorithm-based FSM synthesis with area-power trade-offs

Saurabh Chaudhury^a, Krishna Teja Sistla^b, Santanu Chattopadhyay^{c,*}

^a Department of Electrical Engineering, National Institute of Technology, Silchar 788010, India

^b Mentor Graphics Noida Pvt Ltd., Logix Techno Park, Building – A, Plot # 5, Sector – 127, Noida 201301, UP, India

^c Department of Electronics & Electrical Communication Engineering, Indian Institute of Technology, Kharagpur 721302, India

ARTICLE INFO

Article history:

Received 6 September 2007

Received in revised form

29 September 2008

Accepted 22 November 2008

Keywords:

Genetic algorithm

Low leakage

FSM synthesis

Area-power trade-offs

ABSTRACT

Traditionally, state-encoding strategies targeting minimization of area, dynamic power or a combination of them have been utilized in finite state machine (FSM) synthesis. With drastic scaling down of devices at recent technology level, leakage power has also become an important design parameter to be considered during synthesis. A genetic algorithm-based state encoding, targeting area and power minimized FSM, has been proposed in this paper. A unified technique to reduce both static power (leakage) and dynamic power along with area trade-off has been carried out for FSM synthesis, targeting static CMOS NAND–NAND PLA, dynamic CMOS NOR–NOR PLA and pseudo-NMOS NOR–NOR PLA implementations. Suitable weights for area, leakage power and dynamic power to minimize power density have also been explored. Simulation with MCNC benchmarks shows an average improvement of 31%, 26% and 29% in leakage power consumption, dynamic power consumption and area requirement respectively, over NOVA-based state assignment technique in case of dynamic CMOS PLA implementation. Improvements of 30% in leakage power and 15% in area have been obtained for pseudo-NMOS PLA implementation. For the static CMOS case, the improvements are about 29% in leakage power consumption, 14% in dynamic power consumption and 18% in area requirement.

© 2008 Elsevier B.V. All rights reserved.

1. Introduction

Higher packing density and smaller device dimensions are the characteristics of modern VLSI chips containing millions of devices. Continuous thrive for achieving better circuit performance along with reduced power consumption demands for lower threshold voltage, ultra-thin dielectric and shorter channel length. Although scaling down of supply voltage reduces power consumption significantly, yet leakage power is emerging as a prime concern today because of its exponential dependence on threshold voltage. Gate oxide tunneling leakage has become a non-negligible component of leakage in modern ultra-thin gate oxides, and hence leakage power is quite comparable to dynamic power with current technology. These constraints demand for introduction of a new cost function that will include static and dynamic power in early phases of design optimization, especially at system level in order to reduce total power consumption. Partitioning, state encoding, simultaneous partitioning and state encoding, finite state machine (FSM) decomposition are some of the well-known techniques for reducing power consumption in sequential circuits realized as FSMs. Quite a large number of

contributions can be found in the literature [1–6,18,19,21,23], some of which use genetic algorithm-based state encoding for area and dynamic power minimization, but none of these target the issue of static power minimization. State encoding along with type of flip–flop selection for low power FSM synthesis has been addressed in [7,8]. Low power FSM synthesis with testability has been proposed in [11–13]. Low power state encoding for two- and multi-level implementation of FSM has been proposed in [9]. A method of non-uniform code length and Huffman style state encoding has been proposed in [10] for low power FSM synthesis. Physical partitioning approaches have been proposed in [1,24–28], whereby a finite state machine is decomposed into a number of coupled submachines. Most of the time, only one submachine is active. This can lead to substantial saving in power consumption. However, decomposition of the FSM into two or more disjoint parts often results in a considerably increased area for the circuit. In [22], reduction of power dissipation in FSM circuits has been achieved by minimizing the switching activity in the state register. It introduced a novel state assignment method that utilizes dynamic loop information extracted from FSM profiling data. In [29–31] clock gated FSM synthesis approaches have been presented which dynamically stop the clock to unused parts at certain input states and thus save power. A reengineering approach has been proposed in [32,33] that works on a given FSM to create functionally equivalent but topologically different machines. It uses the concept of *state-splitting* to split states with

* Corresponding author. Tel.: +91 3222 283564; fax: +91 3222 255303.

E-mail addresses: saurabh@nits.ac.in, saurabh1971@gmail.com (S. Chaudhury), s11.teja@gmail.com (K.T. Sistla), santanu@ece.iitkgp.ernet.in (S. Chattopadhyay).

large number of input edges into two or more states and distribute the edges between the newly created states. Now, standard encoding tools can be used to obtain the final circuit. It shows on an average 2.7% area reduction and 5.5% dynamic power reduction. However, the works do not report any leakage power results.

It is only in [14] that both static and dynamic power have been addressed for the first time and a technique developed to minimize them. A cost function that can be used as a metric for static power consumption in the sequential part of an FSM has been introduced. But, the power consumed in the combinational part has not been accounted for. In this paper, we implement the combinational logic part of FSMs in three different styles—static CMOS NAND–NAND PLA, dynamic CMOS NOR–NOR PLA, as well as pseudo-NMOS PLA, and formulate a cost function extending the work done in [14]. Main contribution of our work is that we have formulated a new cost metric, which combines leakage power contributions both from combinational and sequential part. Hence it can be used as cost metric for static power in FSMs. Using this new cost metric, a genetic algorithm-based technique for state encoding targeting low static and dynamic power consumption is proposed. Moreover, a trade-off is also sought for combined area and power minimized FSM synthesis. Combined area, dynamic power and static power consumption-based cost functions have been used for the first time in this particular work. We also present a preliminary work on thermal-aware state encoding by attempting to minimize the power density of the synthesized circuit.

Rest of the paper is organized as follows. In Section 2, we briefly describe the cost functions used to estimate the leakage power consumed in the sequential part of the FSM as suggested in [14] and propose the procedure to calculate the leakage power in the combinational part implemented as a PLA. In Section 3, we describe a procedure to calculate the dynamic power consumed in the FSM. In Section 4, the cost function used for calculating the combinational area for the PLA-based implementation is briefed. This has been used for rough estimation of power density, which is responsible for heat generation and is a concern for thermal limit of the chip. Section 5 presents the genetic algorithm-based formulation to achieve area-power trade-off. Experimental results on benchmark FSMs follow in Section 6. Finally some concluding remarks for the work have been reported.

2. Modeling leakage power

In this section we detail our model used to estimate the leakage power consumed by a circuit realizing an FSM with a

particular state assignment. As the circuit consists of two separate partitions, namely the sequential part and the combinational part, our model handles these two separately.

2.1. Leakage estimation of sequential components

The sequential part of an FSM essentially consists of a set of flip-flops. Thus, the leakage can be estimated by determining the leakage corresponding to each of the flip-flops and summing them up. Consider a transition from state S_i to state S_j in the FSM with k number of flip-flops (representing the state register). Let c_{il} and c_{jl} be the l th bit of codes c_i and c_j assigned to the states S_i and S_j , respectively. The leakage in the corresponding flip-flop can be estimated as [14]

$$X \text{LeakTable}(\text{clk} = 1, \text{output} = c_{jl}, \text{input} = c_{il}) + (1 - X) \text{LeakTable}(\text{clk} = 0, \text{output} = c_{jl}, \text{input} = c_{il}) \quad (1)$$

where X is the fractional duty cycle, LeakTable contains the leakage current values of a flip-flop for different input–output combinations and clock. The duty cycle of the clock has been assumed to be 50%. The LeakTable values have been found via *Spice* simulation as discussed later. Thus the total leakage for a transition from state S_i to state S_j can be written as

$$P_{ij} \sum_l (X \text{LeakTable}(\text{clk} = 1, \text{output} = c_{jl}, \text{input} = c_{il}) + (1 - X) \text{LeakTable}(\text{clk} = 0, \text{output} = c_{jl}, \text{input} = c_{il}))$$

where $0 \leq l \leq k$. P_{ij} is the total state transition probability between the states S_i and S_j .

To compute the P_{ij} values, we need to consider the FSM as a Markov Chain [15]. Let the FSM consist of n states, with the steady-state probabilities P_1, P_2, \dots, P_n . If the conditional state transition probability from state S_i to state S_j is p_{ij} , we can write a set of Chapman–Kolmogorov equations [15]:

$$P_i = \sum_j (P_j p_{ji}), \quad 1 \leq i \leq n$$

with the normality condition, $\sum_i P_i = 1, 1 \leq i \leq n$.

Solving these equations we obtain the values for the steady-state probabilities of the individual FSM states. Now, the total transition probability from state S_i to state S_j can be expressed as

$$P_{ij} = P_i p_{ij}$$

Next, to obtain the LeakTable values, we have to simulate the flip-flop for various input combinations. Since we are targeting a low power realization, we have used the HLFF [16] structure as shown in Fig. 1. It has been implemented in 0.18 μm technology in *Cadence* simulator. The corresponding leakage values are shown in Table 1. Here Q_p is the current value of Q .

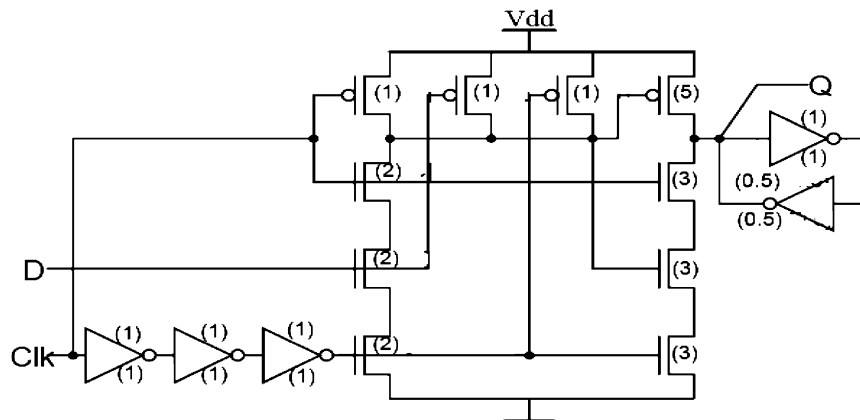


Fig. 1. The HLFF circuit.

Download English Version:

<https://daneshyari.com/en/article/540198>

Download Persian Version:

<https://daneshyari.com/article/540198>

[Daneshyari.com](https://daneshyari.com)