

# State Assignment for FSM Low Power Design

M. Koegst, G. Franke, K. Feske

FhG IIS Erlangen - Department EAS Dresden

D-01069 Dresden, Zeunerstr. 38, Germany,

e-mail: koegst@eas.iis.fhg.de

## Abstract

*The paper concerns low power design of synchronous FSM and power estimation regarding a given input sequence. A novel and practical approach for state assignment is suggested by means of which the average rate of register switching is reduced. We achieved more realistic power estimates in comparison with the probabilistic approach. Experimental results demonstrate the effectiveness of the proposed approach.*

## 1 Introduction

Power dissipation is becoming a critical parameter in the design of microelectronic circuits, especially in portable computing and personal communication applications. Therefore a low power design of FSMs implies the necessity of adapted minimization approaches at both technology-independent and technology-dependent level of specification and moreover the possibility of an adequate power estimation for a synthesized circuit.

In the literature different approaches and synthesis methods are published (overview e.g. in [6, 16]). At the logic level there are three main viewpoints for power minimization: factorization including restructuring [1, 4, 9, 17, 18], state assignment [2, 8, 12, 15] and logic minimization [19]. After mapping at a library further reduction of power dissipation is possible by retiming at gate level ([10], e.g. by critical path reduction [5]).

Methods for power estimation of combinational circuits based on primary input signal probability have been presented previously (e.g. [3, 7]). Power estimation of sequential circuits [14] is significantly more difficult, because state line probabilities have to be taken into account. In [11, 20, 21] a static and a simulative approach is presented on the assumption that the inputs are uncorrelated. In the static procedure state line probabilities can be explicitly computed for circuits up to 15 flip-flops [11] and approximately

otherwise. The simulative approach uses internally created, random input sequences. Both approaches are implemented in SIS [11,19] version 1.2.

In [13] a technique for the power estimation under user-specified input sequences is proposed. The advantage of this approach is that both, temporal and spatial correlations between the inputs are considered.

The goal of our paper is at first to present a new state assignment procedure for low power, at second to discuss disadvantages of the probabilistic power estimation approaches and at third to demonstrate our strategy for low power design and power estimation with regard to a given input sequence.

Our paper is structured as follows. In Section 2 we formulate preliminaries and basic definitions for the behavior description of a FSM with regard to an input sequence and describe an adapted power dissipation model. Based on a definition for the register switching rate  $\delta$  we outline in Section 3 our state assignment procedure for low power. In Section 4 we dispute the probabilistic approach for power estimation. The main steps of our strategy for FSM design and power estimation are outlined in Section 5. Experimental results are presented in Section 6.

## 2 Preliminaries and Definitions

### 2.1 Finite State Machine

Let a initial finite state machine (FSM) be denoted by  $M=(X,Y,S,f,g,S_0)$  where  $X$ ,  $Y$ ,  $S$ ,  $S_0$  are the sets of inputs, outputs, states and initial states, respectively.  $f:X \times S \rightarrow S$  is the transition function and  $g:X \times S \rightarrow Y$  the output function. State transition graph (STG) and state transition table (STT) are two equivalent representations of a FSM. A FSM is called completely-specified if for each state  $s_i$  from  $S$  and each input  $x$  from  $X$  the function  $f(x,s_i)$  is defined, otherwise it is incompletely-defined.

A STG is defined by the vertex set  $S$  and a related set of edges with elements  $(s_i,s_j)$  weighted by a condition  $x$  for the transition from  $s_i$  to  $s_j$  and the corresponding output  $y=g(x,s_i)$ . In our example FSM, Figure 1,  $s_1$  is the initial state.

This work has been partially supported by DFG Project SFB 358 and the JESSI Applications Project AC-8.

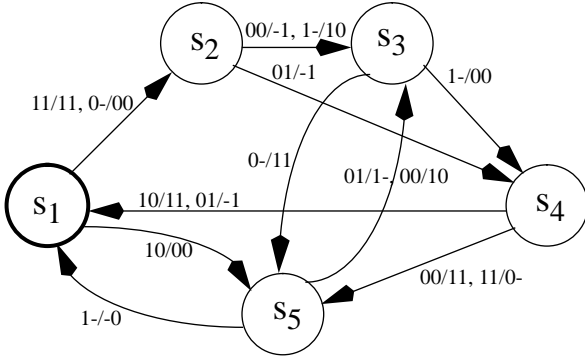


Fig. 1: STG of the example FSM

A STT is defined by a set of 4-tuples  $(x, s_i, s_j, y)$  of input  $x$ , present state  $s_i$ , next state  $s_j=f(x,s_i)$  and output  $y=g(x,s_i)$ .

## 2.2 Relative Frequency and Switching Density

Due to low power design the behavior of the FSM is investigated with regard to a given user-specified input sequence of inputs  $x$  from  $X$  where each  $x$  is given as a  $m$ -tuple of values for all  $m$  primary input signals  $x_1, x_2, \dots, x_m$ .

For our example Figure 1 with two primary input signals  $x_1$  and  $x_2$   $X=\{00,01,10,11\}$  is the input set and e.g.

$$\begin{aligned} &00, 01, 11, 10, 01, 10, 01, 00, 10, 01, \\ &00, 01, 11, 10, 00, 01, 11, 10, 10, 11 \end{aligned} \quad (1)$$

a sequence of  $N=20$  input tuples. For simulation the index  $t$  of an input tuple corresponds with the clock cycle  $t$ .

With regard to such a sequence (1) we define:

- $N$  length of the sequence,
- $n(x_i)/N$  relative frequency of input signal  $x_i$  where  $n(x_i)$  is the number of sign 1 in the  $i$ -th column of the input sequence,
- $d(x_i)/(N-1)$  switching density of input signal  $x_i$  where  $d(x_i)$  is the number of clocks  $t$  for which signal  $x_i$  has switched, i.e.  $x_i(t) \neq x_i(t-1)$ ,
- $n(c_i)/N$  relative frequency of state line  $c_i$  where  $n(c_i)$  is the number of clocks  $t$  for which sign 1 exists in the  $i$ -th column of the corresponding present state code,
- $d(c_i)/(N-1)$  switching density of state line  $c_i$  where  $d(c_i)$  is the number of clocks  $t$  for which signal  $c_i$  has switched, i.e.  $c_i(t) \neq c_i(t-1)$ ,
- $n(s_i, s_j)/N$  relative frequency of state transition  $s_i$  to  $s_j$  where  $n(s_i, s_j)$  is the number of clocks  $t$  for which there is a transitions from state  $s_i$  to  $s_j$ ,
- $d(g_i)/(N-1)$  switching density of gate  $g_i$  where  $d(g_i)$  is the number of clocks  $t$  for which the output of gate  $g_i$  has switched.

In contrast to our density based concept in the literature (e.g. [20]) the probabilistic model is used. Between the relative frequency and the related probability there is the fol-

lowing relation:

The signal probability  $p(x_i)$ , total state transition probability  $P(s_i, s_j)$  and state line probability  $p(c_i)$ , is the limiting value approached by the relative frequency of input signal, state transition and state line, respectively, as the length of the input sequence is increased infinitely. In each input sequence the inputs are correlated. Therefore replacing the relative frequency by the corresponding probability is an approximation because this case the independence of the inputs is assumed.

Applying the above given input sequence (1), starting with state  $s_1$  and the state code in Figure 2 we achieve for the example STG Figure 1 the two further sequences: a sequence of state transitions and a sequence of present state codes.

We get the following relative frequencies and switching densities for the signals  $x_i$  and the state lines  $c_i$ , respectively, by counting sign '1' in the given input sequence and in the related sequences of present state codes:

$$\begin{aligned} n(x_1)/N &= 0.50 & d(x_1)/(N-1) &= 0,47 \\ n(x_2)/N &= 0.50 & d(x_2)/(N-1) &= 0,63 \\ n(c_1)/N &= 0.90 & d(c_1)/(N-1) &= 0,21 \\ n(c_2)/N &= 0.45 & d(c_2)/(N-1) &= 0,47 \\ n(c_3)/N &= 0.50 & d(c_3)/(N-1) &= 0,53 \end{aligned}$$

Already this small example shows that the relative frequencies can differ enormously from the corresponding switching densities. This relation is discussed in Section 4.

## 2.3 Conditional Transition Probability

In the literature (e.g. [2]) the state transition probability  $p(s_i, s_j)$  will be approximated by the so-called conditional transition probability  $Q(s_i, s_j)$ .  $Q$  is there a function of the probabilities  $p(x_k)$  of all  $m$  primary inputs and the conditions  $b(s_i, s_j)$  for the transition from state  $s_i$  to  $s_j$ . In preparation of Section 4.3 we define now  $Q$ .

Let be  $B(s_i, s_j)$  the set of all tuples  $e=(e_1, \dots, e_m)$  of  $X=\{0,1\}^m$  for which the condition  $b(s_i, s_j)$  is fulfilled then  $Q$  is defined (analogously to [15, 17]) by

$$\begin{aligned} Q(s_i, s_j) &= \sum_{e \in B(s_i, s_j)} \left( \prod_{k=1}^m q(e_k) \right) \text{ with} \quad (2) \\ q(e_k) &= \begin{cases} p(x_k) & \text{if } e_k = 1 \\ 1 - p(x_k) & \text{if } e_k = 0 \end{cases} \end{aligned}$$

## 2.4 Model of Power Dissipation

The average total power dissipation  $P$  is a function of the average power dissipated by each gate  $g_i$  during one clock cycle  $T_{\text{cycle}}$  as follows [2]

$$P = \left( V_{dd}^2 / 2T_{\text{cycle}} \right) \sum_{g_i \in G} C(g_i) \frac{d(g_i)}{N-1} \quad (3)$$

where  $V_{dd}$  is the supply voltage,  $C(g_i)$  the capacitive load at the output of gate  $g_i$ ,  $d(g_i)/(N-1)$  the switching density of

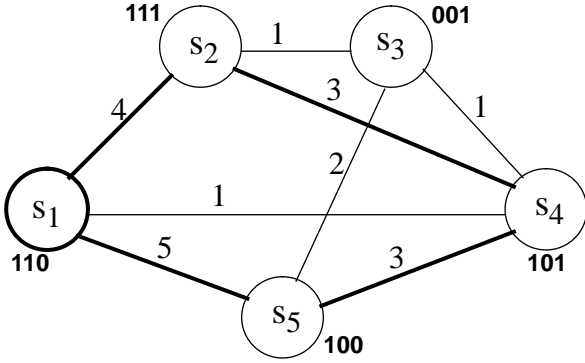
gate  $g_i$  and  $G$  the set of all gates in the circuit.

Voltage  $V_{dd}$  and clock cycle  $T_{cycle}$  are assumed to be fixed. But the sum in (3) can be decreased by resynthesis of the combinational logic of the FSM. We do it by an adapted state assignment procedure. Then the resulting code determines the register configuration and acts upon size and structure on the combinational circuit.

### 3 State Assignment for Low Power

#### 3.1 Register Switching Rate

Our approach for state assignment aims at diminishing the switching activity of state transitions. We do it by minimizing the number of state variables that switch when the FSM changes between two adjacent states. For that we modify the STG to a state graph (Figure 2). The edges  $(s_i, s_j)$  of the state graph are weighted by the number  $n(s_i, s_j) + n(s_j, s_i)$  of transitions between the states  $s_i$  and  $s_j$ . These numbers originate from simulation using the input sequence. The resulting graph is the basic model for the next steps.



**Fig. 2: State graph for STG (Figure 1) encoded and weighted with numbers of transitions for the input sequence (1), starting with state  $s_1$**

In contrast to our approach the state graph in [17] is weighted from a probabilistic point of view. There a measure for a state code  $C$  is defined by

$$\gamma(C) = \sum_{i,j} p(s_i, s_j) \cdot H(C(s_i), C(s_j)) \quad (4)$$

where  $p(s_i, s_j)$  represents the state transition probability and  $H$  the Hamming distance between the codes of state  $s_i$  and  $s_j$ . The task is to find a state code  $C$  with a minimal measure  $\gamma(C)$ . For that simulated annealing [17] was used. In [2, 12] state assignment for low power is outlined as an integer linear programming problem.

Our aim for state assignment is at first to find a state code which cause a minimal rate of register switching with respect to the weighted graph in Figure 2 and at second to improve this code reducing area and power. The ideal case would be if only one state variable changes in each possible state transition but usually this cannot be guaranteed.

For a given FSM we define the register switching rate  $\delta(C)$  for its state assignment  $C$  with regard to a given input sequence of the length  $N$  by

$$\delta(C) = \sum_{i,j} \frac{n(s_i, s_j)}{N} \cdot H(C(s_i), C(s_j)) \quad (5)$$

$\delta(C)$  holds the relation  $1 \leq \delta(C)$  for all state codes  $C$ . The Gray code is characterized by  $\delta(C) = 1$  and the one-hot code by  $\delta(C) = 2$ .

The register switching rate  $\delta$  is a generalization of weight  $\gamma$  in [17]. The relation between the state transition probability  $p(s_i, s_j)$  and the total state transition probability  $P(s_i, s_j)$  is described by

$$P(s_i, s_j) = p(s_i, s_j) \cdot P(s_i), \quad (6)$$

where  $P(s_i)$  is the state probability. Weight  $\gamma$  corresponds to rate  $\delta$  because

$$P(s_i, s_j) = \lim_{N \rightarrow \infty} \frac{n(s_i, s_j)}{N} \quad (7)$$

and moreover, if  $P(s_i) = 1$  is assumed for all states in (5).

#### 3.2 Encoding Procedure

Our encoding procedure operates as follows:

1. Determination of the relative frequency for all state transitions by simulation:  
Applying the user-specified primary input sequence to the FSM we determine the relative frequency  $n(s_i, s_j)/N$  for all state transitions  $(s_i, s_j)$ .
2. Computation of a code  $C_k$  of width  $k = \lceil \text{Id } |S| \rceil$ :  
Using the register switching rate  $\delta$  as a measure for the state code quality we solve the state assignment problem by simulated annealing, analogous to [17].
3. Computation of a code  $C_{k+1}$  of width  $k+1$ :  
In the case  $3 \cdot 2^{k-2} \leq |S| \leq 2^k$  we compute additionally a code  $C_{k+1}$  of width  $k+1$  with a minimal register switching rate  $\delta(C_{k+1})$ . Our experiments have namely shown, the smaller the difference of  $2^k - |S|$  the greater is the probability that  $\delta(C_{k+1}) < \delta(C_k)$  and the circuit synthesized with code  $C_{k+1}$  has a smaller power dissipation.
4. Code modification:  
The computed code can be refined by negation of selected state lines and using don't cares in the state code, i.e. by replacing one or more sign „1“ by sign „-“.

### 4 Power Estimation Using the Input Signal Probability

#### 4.1 Probabilistic Approach

In these sections we discuss the probabilistic model of power estimation and show that assumptions for it imply a deviation from the real power dissipation.

The probabilistic approach is characterized by the following features:

1. It considers neither the temporal correlation between the input signals, i.e. the relationship between consecutive input vectors, nor the spatial, i.e. relationship between bits within a vector.
2. The difference between the switching density and the probability results in deviations of the computed power dissipation, i.e. the estimated power differs from the more realistic value using the switching density (Section 4.2).
3. The state transition probability  $p(s_i, s_j)$  necessary for the estimation is approximated e.g. in (4) by the conditional transition probability  $Q(s_i, s_j)$  (Section 4.3).

## 4.2 Relation between the Relative Frequency and the Switching Density

Under a simplified model, the power dissipation of a circuit is directly related to the switching density. In the probabilistic case the probability is used instead of the corresponding switching density (e.g. [20]).

frequency $n(x_i)$	1	2	3	4	5	6	7	8	9	10
Input sequences for different frequencies $n(x_i)$	0	0	0	0	0	1	1	1	1	1
	0	0	0	0	1	0	1	1	1	1
	0	0	0	0	0	1	1	1	1	1
	0	0	0	1	1	0	0	1	1	1
	0	0	0	0	0	1	1	1	1	1
	0	0	1	1	1	0	0	0	0	1
	0	0	0	0	0	1	1	1	1	1
	0	1	1	1	1	0	0	0	0	1
	0	0	0	0	0	1	1	1	1	1
	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	
rel. frequency $n(x_i)/N$	1/11	2/11	3/11	4/11	5/11	6/11	7/11	8/11	9/11	10/11
max. density $d(x_i)/(N-1)$	1/5	2/5	3/5	4/5	1	1	4/5	3/5	2/5	1/5

**Table 1: Input sequences for  $x_i$  with a maximal density with regard to a given frequency  $n(x_i)$**

By means of sequences of length  $N=11$  for a signal  $x_i$  we demonstrate in Table 1 the relation between the relative frequency and the switching density. As an example we construct for  $x_i$  ( $i=1, \dots, 10$ ) for each possible frequency  $n(x_i)=1, \dots, 10$  a sequence with the maximal switching density  $d(x_i)/(N-1)$ . The minimal density is in all cases  $1/10$ .

The range of the density between upper and lower bound is depicted in Figure 3. In the worst case  $n(x_i)=5$  and  $6$  (i.e. a relative frequency of  $5/11$  and  $6/11$ , respectively) the switching density can differ between  $1/10$  and  $1$ . Remember that the relative frequency corresponds to the relevant signal probability.

Generally the minimal and the maximal switching density is given by

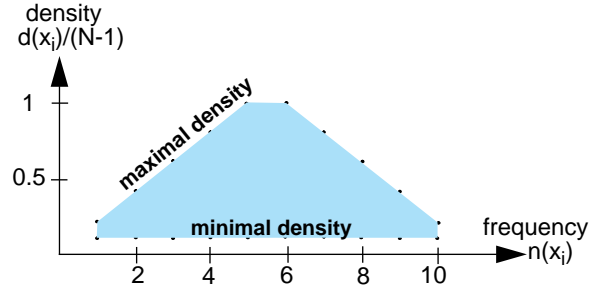
$$\frac{1}{N-1} \quad \text{and} \quad \frac{2 \cdot \max(n(x_i), N-n(x_i))}{N-1}, \quad \text{respectively.}$$

For  $p(x_i)=0,5$  with  $p(x_i) = \lim_{N \rightarrow \infty} \frac{n(x_i)}{N}$  the switching den-

sity is bounded by the limes of the minimal and the maximal density:

$$0 < \lim_{N \rightarrow \infty} \frac{d(x_i)}{N-1} \leq 1.$$

In this worst case there is no correlation between the input signal probability and the corresponding switching density. Consequently the estimate of power dissipation may differ largely if the signal probability is used instead of the corresponding switching density. The different bits of a counter are examples for the extreme deviations between density and probability



**Fig. 3: Upper and lower bound of the density depending on the frequency (s. Table 1)**

## 4.3 Approximation of the Transition Probability

For power estimation the state transition probability  $p(s_i, s_j)$  for all states  $s_i$  and  $s_j$  of the corresponding FSM is necessary. In the literature (e.g. [2]) this probability  $p(s_i, s_j)$  will be approximated by the conditional transition probability  $Q(s_i, s_j)$ , s. section 2.3.

If the FSM is completely-specified the relation

$$Q(s_i) = \sum_{s_j \in \text{SUC}(s_i)} Q(s_i, s_j) = 1$$

holds for each state  $s_i$ , where  $\text{SUC}(s_i)$  is the set of all successors of  $s_i$ .

If the FSM is incompletely-specified it is proposed in [17] to make it completely-specified by introducing a self loop at each state  $s_i$  corresponding to the don't care input to that state. But in this way the possibilities for logic minimization are reduced and moreover the assumed state transition probabilities are changed.

This section shows that replacing  $p(s_i, s_j)$  by  $Q(s_i, s_j)$  which follows from the input conditions cannot be justified.

Because of the above discussed disadvantages of the probabilistic model we apply for our approach an user-specified primary input sequence.

## 5 Strategy for Design and Power Estimation with regard to an Input Sequence

In comparison with the probabilistic approach we get a more realistic estimation utilizing an user-specified input sequence starting from an initial state. Our approach is

based on such a given input sequence. The same sequence is used by both, the encoding method aiming at a minimized register switching rate and the power estimation via simulation with the tool SIS. For the simulation we extended SIS such, that power estimation is possible using a given input sequence.

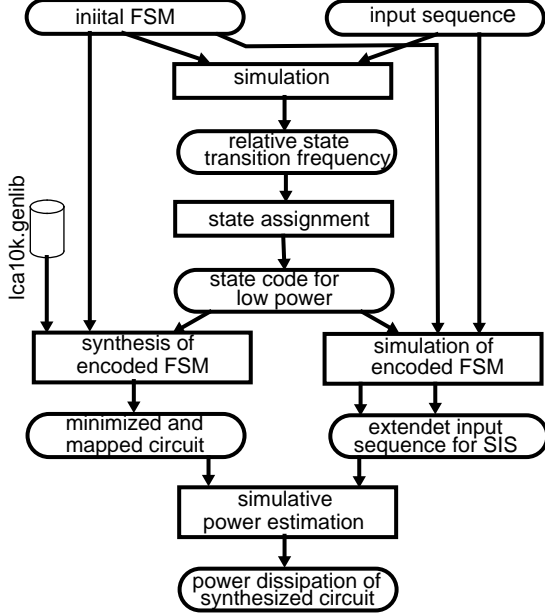


Fig. 4: Strategy for design and power estimation

We propose a strategy for low power design and power estimation composed of four main steps (Figure 4):

1. State encoding for low power with regard to an input sequence.
2. Minimization of the encoded FSM and mapping at the given library (e.g. lca10k.genlib).
3. Extending the given sequence of the primary inputs to a sequence consisting of 3-tuples <input, next input, present state> [7].
4. Power estimation of the synthesized circuit by simulation with regard to the extended sequence utilizing a modified procedure of SIS1.2 [11, 19].

## 6 Experimental Results

To illustrate our encoding approach for low power we carried out two groups of experiments applying input sequences of about 10.000 pattern.

The goal of the first group is to demonstrate the relation between the state code and the power dissipation. For the FSM „cse“ of the MCNC benchmarks [22] we computed codes and estimated the power dissipation for seven state codes with regard to a defined input sequence. The results are depicted in Table 2. The codes C4\_1, C4\_2, C4\_j and C4\_n have a minimal width of four. Code C4\_j and C4\_n

are computed by JEDI and NOVA implemented in SIS [19], respectively. (Remember the state encoding procedures in JEDI and NOVA are especially oriented on area minimization.) Codes C4\_1 and C4\_2 are computed by applying the second respectively the second and the fourth encoding steps (Section 3.2). C5\_1, C5\_2 and C5\_3 are codes of width five which are achieved analogously by using our procedure. E.g., for state  $s_{13}$  we found by simulated annealing (third step) the code  $C5_1(s_{13})=11100$ , by negation of both first two and last two lines code  $C5_2(s_{13})=00111$  and finally by replacing the last sign „1“ by sign „-“ (fourth step) code  $C5_3(s_{13})=0011-$ .

	state code						
	C4_1	C4_2	C5_1	C5_2	C5_3	C4_j	C4_n
switching rate $\delta$	1.163	1.163	1.156	1.156	-	1.613	2.005
P without min. ( $\mu W$ )	1336	1302	1517	1177	<b>1054</b>	1348	1539
P with min. ( $\mu W$ )	565	535	615	588	<b>507</b>	670	635
comparison (%)	111	106	121	114	<b>100</b>	132	125

Table 2: Power dissipation using seven different state codes for FSM „cse“ (P in  $\mu W$ )

In the first row of Table 2 the register switching rate  $\delta$  for the codes are given, except code C5\_3 for which  $\delta$  is not defined. All codes of the group C4 and especially C4\_j and C4\_n obtained by JEDI and NOVA have a greater switching rate than the codes C5 of width five.

design	#i	#o	#s	#p		C	C_j	C_n
cse	7	7	16	91	$\delta$	1.1569	1.6130	2.0051
					P	507.7	670.2	635.5
pma	8	8	24	73	$\delta$	1.2503	1.9060	2.4389
					P	904.3	1267.5	2269.5
dk16	2	3	27	108	$\delta$	1.6021	2.5691	2.6289
					P	1696.2	1933.0	2027.7
keyb	7	2	19	170	$\delta$	1.1679	1.7379	1.6808
					P	734.5	1034.0	1172.6
modulo12	1	1	12	24	$\delta$	1.0000	1.1666	3.1668
					P	109.1	142.3	254.9
sand	11	9	32	184	$\delta$	1.2458	1.7273	3.33644
					P	1376.8	1585.3	2431.4

Table 3: Power estimates for different designs (minimization: source script.rugged)

In the second and third row the power dissipation P of the circuits are entered without logic minimization and after using SIS, script.rugged, respectively. For the power estimation the circuits are mapped at the library lca10k.genlib. Our optimized code C5\_3 requires the lowest dissipation (507  $\mu W$ ) which is the basis (100%) in comparison with the power of the other codes.

The second group of experiments is collected in Table 3. For six circuits of the MCNC FSM benchmarks we calcu-

lated the register switching rate  $\delta$  and estimate the power dissipation  $P$  ( $\mu W$ ) in three cases: For our optimized code  $C$ , for the JEDI code  $C_j$ , and for the NOVA code  $C_n$ . In the second column the parameters of the designs are depicted: number of primary inputs, primary outputs, states, and product terms.

The results of the experiments can be summarized as follows:

- In all cases the design with the optimized code  $C$  requires the lowest value of power dissipation.
- Usually the design encoded by NOVA needs more power in comparison with JEDI.
- For design „modulo12“ our encoding procedure finds the Gray code with the minimal switching rate  $\delta=1$ .
- Comparing two different codes  $C$  and  $C'$  of a design the following trend can be recognized: The difference of the register switching rates  $|\delta(C)-\delta(C')|$  is in proportion to the difference of the power dissipation  $|P(C)-P(C')|$  weighted by the factor  $\frac{v}{m+v}$ , where  $m$  and  $v$  are the numbers of primary input variables and the state code width, respectively.

## 7 Conclusions and Future Works

In our paper we presented a FSM state assignment procedure for reducing the switching activity in the synthesized circuit. For measuring the power dissipation of a circuit we proposed an adapted approach for power estimation with regard to a given (user-specified) input sequence. Such a sequence can be got from recording the inputs of the FSM in the real environment or in the simulation process. In our experiments the lowest power dissipation is obtained if the same input sequence was used both, by our encoding procedure and by the power simulation.

Our future work is focused on two directions: on improving this assignment procedure with regard to the code quality and on reducing the necessary CPU time and moreover in simultaneous consideration of several low power aspects at different design levels.

## References

- [1] Alidina,M.; Monteiro,J.; Devadas,S.; Ghosh,A.; Papaefthymiou,M.: Precomputation-Based Sequential Logic Optimization for Low Power. IEEE Transactions on VLSI Systems, Dec. 1994, pp.426-436.
- [2] Benini,L.; de Micheli,G.: State Assignment for Low Power Dissipation. IEEE Journal of Solid-State Circuits, Vol. 30, No. 3, March 95, pp. 32-40.
- [3] Burch,R.; Najm,F.; Yang,P.; Trick,T.: Monte Carlo Approach to Power Estimation. International Conference on Computer-Aided Design, 1992, pp. 90-97.
- [4] Benini,L.; Siegel,P.; de Micheli,G.: Saving Power by Synthesizing Gate Clocks for Sequential Circuits. IEEE Design & Test of Computers, 1994, pp. 32-94.
- [5] Chandrakasan,A.; Potkonjak,M.; Rabaey,J.; Brodersen,R.: HYPER-LP: A System for Power Minimization Using Architectural Transformations. International Conference on Computer-Aided Design, 1992, pp. 300-303.
- [6] Devadas,S.; Malik,S.: A Survey of Optimization Techniques Targeting Low Power VLSI Circuits. 32th DAC, 1995, pp. 242-247.
- [7] Ghosh,A.; Devadas,S.; Keutzer,K. White,J.: Estimation of Average Switching Activity in Combinational and Sequential Circuits. 29th DAC, 1992, pp.253-259.
- [8] Hachtel,G. et al. : Re-encoding sequential circuits to reduce power dissipation. Int. Workshop on Low-Power Design, Napa, April 1994, pp.69-73.
- [9] Iman,S.; Pedram,M.; Logic Extraction and Factorization for Low Power. 32th DAC, 1995, pp. 248-253.
- [10] Monteiro,J.; Devadas,S.,Ghosh,A.: Retiming Sequential Circuits for Low Power. Internat. Conference on Computer-Aided Design, 1993, pp. 398-402.
- [11] Monteiro,J.; Devadas,S., Lin,B.: A Methodology for Efficient Estimation of Switching Activity in Sequential Logic Circuits. 31th DAC, 1994, pp. 12-17.
- [12] Monteiro,J.; Kukula,J.; Devadas,S.; Neto,H.: Bitwise Encoding of Finite State Machines. International Conference on VLSI Design, Calcutta, India, January 1994, pp.379-382.
- [13] Monteiro,J.; Devadas,S.: Techniques for the Power Estimation of Sequential Logic Circuits Under User-Specified Input Sequences and Programs. Int. Symp. on Low Power Design, Laguna Beach, California, Apr. 1995.
- [14] Najm, F.N.; Goel, S.; Hajj, I.N.: Power Estimation in Sequential Circuits. 32th DAC 1995, pp. 635-640.
- [15] Olson, E.; Kang,S.: State Assignment for Low-Power FSM Synthesis Using Genetic Local Search. Proc. of IEEE Custom Integrated Circuits Conf., May 1994, pp.140-143.
- [16] Rabaey,J.M.; Pedram,M.: Low Power Design Methodologies. Kluwer Academic Publishers, Boston/Dordrecht/London, 1996.
- [17] Roy, K.; Prasad, S. C.: Circuit Activity Based Logic Synthesis for Low Power Reliable Operations. IEEE Transact. on VLSI Systems, vol.1,No.4,Dec. 1993.
- [18] Schneider,P.H.; Schlichtmann,U.; Antreich,K.J.: A New Power Estimation Technique with Application to Decomposition of Boolean Functions for Low Power. EURO-DAC 1994, pp.388-393.
- [19] Sentovich,E.; et.al.: „SIS: A System for Sequential Circuit Synthesis“, Memo UCB/ERL M92/41, University of Berkeley, CA, 1992.
- [20] Tsui,C.-Y.; Monteiro,J.; Pedram,M.; Devadas,S.; Despain,A.; Lin,B.: Power Estimation for Sequential Logic Circuits. IEEE Transactions on VLSI Systems, Vol.3, No.3, September 1995, pp.404-416.
- [21] Tsui,C.; Pedram,M.; Despain,A.: Exact and Approximate Methods for Calculating Signal and Transition Probabilities in FSMs. 31th DAC 1994, pp.18-23.
- [22] Yang, S.: Logic Synthesis and Optimization Benchmarks, User Guide Version 3.0, 1991.