FSM Non-Minimal State Encoding for Low Power

I. Lemberski, M. Koegst, S. Cotofana and B. Juurlink

Abstract— **In this paper, we focus our attention on the problem of FSM state encoding for low power. In contrast to many publications where probabilistic approach to power estimation is offered, we consider power measurement based on given user-specified input sequence. Although the power dissipation depends on several parameters (register and output switching activity, complexity of combinational part, capacitance load on the gate), switching is the most important source of power dissipation. Our goal is to develop an encoding procedure which minimizes register switching activity. We start with a highly redundant (seed) encoding and minimize its length while minimizing the register switching activity. Unlike previous works, we don't restrict encoding final length (only register switching activity is considered). Therefore, final encoding length may differ from the minimal one. We tested our encoding procedure on several benchmarks from the MCNC set. The experiments show that in many cases, power dissipation obtained using our encoding (generally, of non-minimal length) is less than one achieved when encoding of minimal length is generated.**

I. Introduction

In CMOS, power dissipation depends on capacitance load at the gates, switching activity and number of gates [8]. However, switching is the most important source of power dissipation. For switching activity estimation, in the literature two approaches are described: probabilistic and deterministic. Within probabilistic one, transition probability is generated. Based on this information, switching activity is computed[8], [10]. In sequential logic (FSM), switching activity strongly depends on state encoding. It is assumed [1], that power dissipation is proportional to the register switching activity. In [4], [5], deterministic approach to FSM register switching activity is offered. It calculates switching activity under user-specified input sequence and therefore, gives more accurate estimation of switching activity than probabilistic one. Within it, switching activity is described as follows:

$$E_{sw} = \sum_{s_i, s_j \in S} \frac{n(s_i, s_j)}{N} H(s_i, s_j) \qquad (1)$$

I. Lemberski is with the School of Computing, Information Systems and Mathematics, South Bank University, 103 Borough Rd., London SE1 0AA, United Kingdom, E-mail: lemberi@sbu.ac.uk

M. Koegst is with the Branch Lab Design Automation, Fraunhofer Institute for Integrated Circuits, Zeunerstr. 38, 01069, Dresden, Germany, E-mail: koegst@eas.iis.fhg.de

S. Cotofana and B. Juurlink are with Delft University for Technology, Mekelweg 4, 2628CD, Delft, The Netherlands, E-mail: {S.D.Cotofana,B.H.H.Juurlink}@cardit.et.tudelft.nl

where $n(s_i, s_j)$ - number of transitions between states s_i, s_j within the given input sequence, $H(s_i, s_j)$ - Hamming distance between encoding of two states, N - user-specified input sequence length. To minimize power dissipation , in [14] weighting switching activity (weighted by the capacitance load) is considered as a cost function, In [9], a linear combination of switching activity and the number of literals is used as a cost function. In [3], [15], the strategy of re-encoding an existing sequential circuit to minimize switching activity is offered. It is shown [15] that encodings obtained by JEDI [7] and NOVA [12] may be improved significantly in terms of power dissipation. In many papers, simulated annealing is considered to find encoding of minimal or fixed length [4], [5], [11], [14]. The encoding length implies on the switching activity. Sometimes, non-minimal encoding gives better solution in terms of switching activity. In most papers, minimal encoding is supposed. In case of non-minimal one, the procedure of searching for the best encoding is based on the incremental approach [5]. Within it, encoding length is increased by 1 starting from the minimal one and for each length, encoding targeting switching activity minimization is created and the results are compared. Encoding that generates minimal switching activity is accepted as a final solution.

In our work, we don' t restrict encoding length and start with highly redundant (so-called seed) encoding with many don't cares and reduce it while minimizing switching activity [2].

II. FSM Description and Motivation of the Approach

FSM F is described as: $F = (X, Y, S, f, g, s_o)$, where X, Y, S - sets of inputs, outputs and states respectively, s_o - initial state, f: $X \times S \rightarrow S$, g: $X \times S \rightarrow Y$ - transition and output functions respectively. In Fig. 1, two inputs and one output FSM is represented by the state transition graph (STG). The encoding of length 3 (non-minimal) gives minimal Hamming distance (equal to 1) between each pair of states involved in a transition. It results in minimal switching activity for any user-specified input sequence. However, no encoding of length 2 (minimal) with the same properties can be found.

The basic idea of our encoding procedure is to construct the state encoding by an iterative improvement

0-7803-7235-2/02/$10.00 © 2002 IEEE

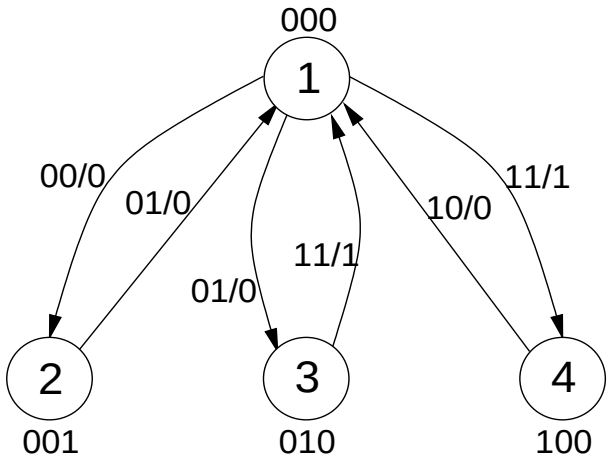

Fig. 1. FSM with non-minimal encoding that results in the best switching activity.

of a generic and highly redundant seed state encoding [2]. Within this approach, we start with the encoding which is described as follows. Let dichotomy $A : B$ be a disjoint block partition $A \bigcap B = \emptyset$ of state set S:$A \subset S$, $B \subset S$. Each dichotomy is associated with a state variable in encoding. It takes the same binary value for all states in the left block and opposite value for all the states in the right block. We arbitrarily assign 1 to the left block and 0 to the right block. To construct a seed encoding, we have to generate all possible pairs of states and treat them as dichotomies with exactly one state in each block: $\{s_i\} : \{s_j\}, s_i, s_j \in S$. Let n be the number of states. Therefore, we have $\frac{n(n-1)}{2}$ dichotomies (or, what is the same, $\frac{n(n-1)}{2}$ rows) in the seed encoding. For FSM (Fig. 2) with 6 states, the seed encoding contains 15 rows (Table I).

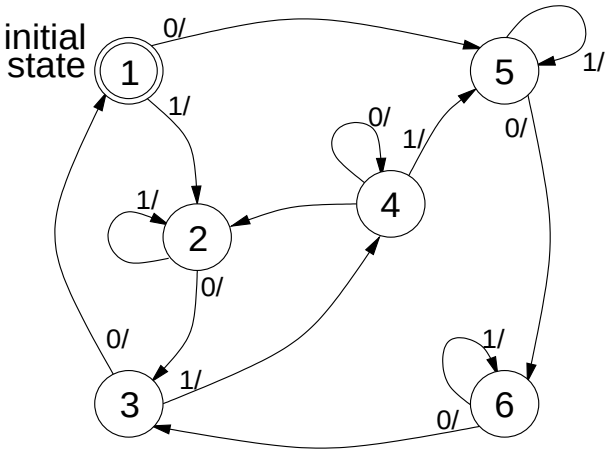

Fig. 2. FSM.

Although the number of state variables grows with n^2 and is therefore, high for practical application, this encoding is a good starting point for the construction

TABLE I

WEIGHTED SEED ENCODING: r - ROW NUMBER, $\{s_i\} : \{s_j\}$-SEED DICHOTOMY,1,2,...,6 - STATE NUMBER,w_{ij}-ROW WEIGHT

r	$\{s_i\} : \{s_j\}$	1	2	3	4	5	6	w_{ij}
1	$\{1\} : \{3\}$	1	-	0	-	-	-	6
2	$\{2\} : \{3\}$	-	1	0	-	-	-	6
3	$\{1\} : \{2\}$	1	0	-	-	-	-	5
4	$\{5\} : \{6\}$	-	-	-	-	1	0	4
5	$\{3\} : \{4\}$	-	-	1	0	-	-	4
6	$\{3\} : \{6\}$	-	-	1	-	-	0	4
7	$\{4\} : \{5\}$	-	-	-	1	0	-	3
8	$\{1\} : \{5\}$	1	-	-	-	0	-	1
9	$\{2\} : \{4\}$	-	1	-	0	-	-	1
10	$\{1\} : \{4\}$	1	-	-	0	-	-	0
11	$\{1\} : \{6\}$	1	-	-	-	-	0	0
12	$\{2\} : \{5\}$	-	1	-	-	0	-	0
13	$\{2\} : \{6\}$	-	1	-	-	-	0	0
14	$\{3\} : \{5\}$	-	-	1	-	0	-	0
15	$\{4\} : \{6\}$	-	-	-	1	-	0	0

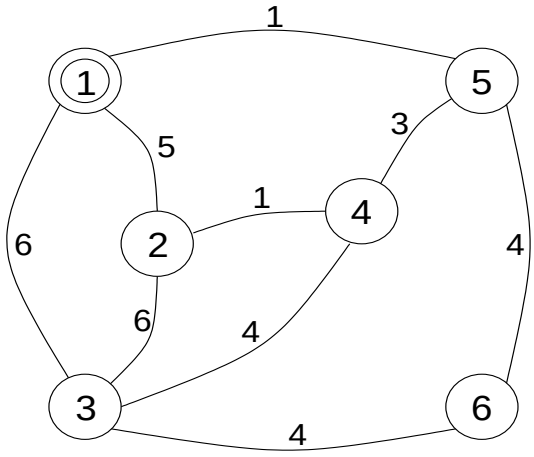

Fig. 3. State graph weighted by the number of transitions under user-specified sequence of length N=34:10010010010 010000011010001000010000.

of the low power state encoding. This comes from the fact that encoding can be constructed by extending and covering dichotomies associated with the variables. Two dichotomies $d_1 = A : B$ and $d_2 = C : D$ are compatible if : 1) $(A \bigcup C) \bigcap (B \bigcup D) = \emptyset$ or 2)$(A \bigcup D) \bigcap (B \bigcup C) = \emptyset$. A dichotomy d_1 is covered by a dichotomy d_2 if $A \subset C$ and $B \subset D$ or $B \subset C$ and $A \subset D$.

A compatible dichotomy d_1 (or d_2) can be extended to a dichotomy 1) $d_{12} = (A \bigcup D) : (B \bigcup C)$ or 2) $d_{12} = (A \bigcup C) : (B \bigcup D)$ to cover dichotomy d_2 (or d_1). As a resul, the number of dichotomies decreases. Such a dichotomy extension is complete if $A \bigcup B \bigcup C \bigcup D = S$.

To construct minimized encoding we sequentially extend and cover dichotomies which are selected according to the value of the cost function (switching activity). There is equivalence between extending (covering) dichotomies and variables. Our strategy looks more clear if we consider extending and covering variables rather than dichotomies.

III. Encoding Procedure

A. Preliminaries

Given a FSM (without outputs) described by the STG (Fig. 2). Let weighted state graph (WSG) [4] be a non - directed graph where edges between vertices (s_i, s_j) are weighted by the number of transitions between the states s_i, s_j under a user-specified input sequence (Fig. 3). The starting point for our method is weighted seed encoding (WSE) instead of WSG (Fig. 3), however there is equivalence between them: in WSG, each transition is weighted but in WSE, we weight a row switching its value within a transition. Each encoding row accepting zero and one values for the states s_i, s_j, is weighted by the number of transitions between mentioned states within the given user-specified sequence. In the Table I, WSE ordered by decreasing row weights is showed.

Let n be the minimized encoding length. The best encoding (in terms of switching activity) has only one switching within each transition. If there is a row r that violates this condition with respect to rows $1, 2, ..., t, t < r, r \leq n$ constructed previously then r produces one more switching between a pair of states (s_i, s_j). We say that r produces extra-switching $Extra$: $(r \rightarrow Extra)$ which is determined as follows:

$$Extra_t(r) = \sum_{\forall (s_i, s_j): r \rightarrow Extra} w_{ij} \qquad (2)$$

where w_{ij} - appropriate seed encoding row weight.

Let $r_1 = \{1\} : \{2, 3, 4, 5, 6\}$, $r_2 = \{2\} : \{1, 3, 4, 5, 6\}$ be minimized encoding rows of FSM (Fig. 2). One can see, that row r_2 produces extra-switching for state pair (1,2) with respect to row r_1, where $w_{1,2} = 5$ and $Extra_1(2) = 5$. For our procedure, (2) is more suitable for switching activity evaluation than (1) because encoding is performed row-by-row. Although only one row is created each time, its structure implies significantly on the rows that will be created further (rows $r + 1, r + 2, ..., n$). Therefore, when row r is produced the cost function is as follows:

$$Extra(sum) = Extra_{r-1}(r) + \sum_{i=r+1}^{n} Extra_r(i) \qquad (3)$$

B. Procedure Description

Given a FSM described by STG and a user-specified input sequence $input\_seq$. Based on this data weighted seed encoding row set $seed$ is generated and ordered producing set $seedo$. Let $seedc$ be a subset of seed encoding rows covered by the rows $1, 2, ..., r - 1$ (included into minimized encoding) and row r (being currently created), $seednc$ - subset of non-covered rows: $seednc = seedo \backslash seedc$.

We propose a two step encoding procedure $encoding()$. In the first step, we reduce the encoding length with the goal to minimize (3). We start with non-covered seed encoding row $r \in seednc, r = A : B, A = \{a\}, B = \{b\}, a, b \in S$, of highest weight and extend it (procedure $extend(r)$) state - by - state assigning 1 or 0 value (rules may be found in [6]) to cover compatible rows. Once row is extended completely (no don't cares) it is included into the reduced encoding. The rows covered are avoided from the subset $seednc$ (they are determined by procedure $cover()$). We continue until all the rows are covered. In Table I, we start with the row 1 being described by the dichotomy $\{1\} : \{3\}$ and extend the row (applying rules from [6]) as follows: $\{1\} : \{2, 3\}$, $\{1\} : \{2, 3, 4\}$, $\{1\} : \{2, 3, 4, 6\}$, $\{1\} : \{2, 3, 4, 5, 6\}$. The row $\{1\} : \{2, 3, 4, 5, 6\}$ (extended completely) is included in the reduced encoding. As our procedure has no backtracks, we improve the solution (procedure $reencoding()$) in the second step. Procedure $reencoding()$ will be explained in the next subsection. Below we give the formal description of procedure $encoding()$.

```
encoding()
 {seed := generate_seed_encoding(STG, input_seq);
  seedo := order_seed_encoding_rows(seed);
  seednc := seedo
  until seednc ≠ ∅
    {for the first row r , r ∈ seednc
      {until r extended completely
        {r = extend(r);
        }
       minimized_encoding := r;
     seednc := seednc\cover(minimized_encoding);
    } }
   encoding = reencoding(minimized_encoding);
  }
```

C. Reencoding

Let n be the minimized encoding length. Let C be a set of 2^n binary vectors that may be assigned to FSM states.

Suppose, $c(i)$ is a binary vector assigned to state i in minimized encoding but $C(s)$ - set of binary vectors assigned to FSM states from S.

In procedure $reencoding()$, each transition is considered. Let (i, j) be a state pair, involved in the transition. If more than one bit is switched within transition between states i and $j, i, j \in S$, procedure reencoding() replaces $c(i)$ and $c(j)$ by other vectors $c_{test}(i), c_{test}(j)$:

$c_{test}(i) \in (C \backslash C(s)) \cup \{c_j\}$ $c_{test}(j) \in (C \backslash C(s)) \cup \{c_i\}$ and calculates $E_{sw}(new)$ using (1). It value (1) is decreased, vectors $c_{test}(i), c_{test}(j)$ are accepted.

In our example (Table I), minimized encoding is obtained in the first step with $E_{sw} = 1,24$: c(1)=1011, c(2)=0111, c(3)=0011, c(4)=0001, c(5)=0000, c(6)=0010. One can see, that extra-switching is produced within transition between states $1,2$. Therefore, we replace vector $c(1)$ by vector $c_{test}(1) = 1001$ and vector $c(2)$ by vector $c_{test}(2) = 1011$. It results in less $E_{sw}(new) = 1,12$. After avoiding a trivial row 2 (containing all zeroes) we obtain an encoding of less length: c(1)=101, c(2)=111, c(3)=011, c(4)=001, c(5)=000, c(6)=010.

IV. Experimental Results

Within the experiment, we compare power dissipation using procedure DET and procedure SA of simulated annealing [4].

To evaluate power dissipation for encodings obtained using procedures DET and SA an extended SYNOPSYS simulation tool [5] is applied. Power dissipation results ($10^{(-5)}W$) are reflected in Table II. For procedure SA, encoding of minimal length ENL_M is generated. One can see (Table II), that for the last 4 examples and in total, encoding ENL_DET (in most cases, non-minimal) obtained by procedure DET results in less power dissipation (column DET) than one obtained by procedure SA (column SA). It should be pointed out that sometimes, encoding length optimal for low power may be quite far from the minimum (for example "sand", minimal encoding length is 5 but the optimal one is 10).

TABLE II
Power dissipation ($10^{(-5)}W$) for DET and SA

Examples	ENL_M	ENL_DET	DET	SA
cse	4	5	8,51	7,77
dk16	5	7	19,77	17,98
ex1	5	6	14,74	13,27
keyb	5	5	11,38	13,79
pma	5	7	10,49	10,63
sand	5	10	19,79	22,29
kirkman	4	5	8,01	9,31
Total:			92,69	95,04

V. Conclusion

In this work we developed approach to FSM state encoding for low power. In contrast to many papers where probabilistic approach to power estimation is offered, we consider power measurement based on a given user-specified input sequence. Instead of encoding of minimal or fixed length proposed in some papers, we don't restrict the encoding length and start with the highly redundant seed encoding. We reduce encoding length using register switching activity as a cost function. As a result the final encoding length may be non-minimal. However, in some examples, non-minimal encoding results in less power dissipation than minimal ones generated by the procedure of simulated annealing [4].

Acknowledgment

This paper reflects the results obtained during the first author's visit of Delft University for Technology (TU Delft). The first author takes this opportunity to thank NWO (Netherlands) for sponsoring his research with TU Delft. Many thanks to Prof. S. Vassiliadis for the permanent attention to this work and excellent working conditions in the laboratory.

References

[1] Benini L., De Micheli G.: State Assignment for Low Power Dissipation. IEEE Journal of Solid - State Circuits, vol. 30, N 3,1995, 258-268

[2] Grass W., Lemberski I.: Support-Based State Encoding Targeting FSM Optimal LUT FPGA Implementation. Proceedings of the International Workshop on Logic and Architecture Synthesis, Dec, 1997, 97-104

[3] Hachtel G.D., Hermida M., Pardo A., Poncino M., Somenz F.: Re-Encoding Sequential Circuits to Reduce Power Dissipation, International Workshop on Low Power Design, April, 1994, 70-73

[4] Koegst M., Franke G., Feske K.: State Assignment for FSM Low Power Design, EURO-DAC96, September, 1996, 28-33

[5] Koegst M., Rulke St., Susse H., Martinez M., Avedillo M.J., Quitana J.M.: Combination of Two Approaches for Low Power Design of Controllers. Proceedings of MIXDES2000, June, 2000

[6] Lemberski I, Cotofana S., Juurlink B.: Efficient FSM State Encoding for Low Power. Technical Report, Delft University for Technology, 2001

[7] Lin B., Newton A.R.: Synthesis of Multi-level Logic from Symbolic High-level Description Language. Proc. IFIP Int. Conf. VLSI, August, 1989, 187-196

[8] Monteiro J., Devadas S.: Computer-Aided Design for Low Power Sequential Logic Circuits. Kluwer Academic Publisher, 1997

[9] Olson E., Kang S.M.: Low-power state Assignment for Finite State Machine Search. Proc. Int. Workshop Low Power Design, April, 1994, 63-68

[10] Rabaev J.M., Pedram M.: Low Power Design Methodologies. Kluwer Academic Publisher, Boston/Dortrecht/London, 1996

[11] Roy K., Prasad S.C.: Syclop: Synthesis of CMOS Logic for Low Power Application. Proc. Int. Conf. Computer Design, October, 1992, 464-467

[12] Sangiovanni-Vincentelli A., Villa T.: NOVA: State Assignment for Optimal Two-level Logic Implementation. IEEE Trans. CAD, September, 1990, vol. 9, N 9, 905-924

[13] Sentovich E., et.al: SIS: A System for Sequential Circuit Synthesis. Memo UCB/ERL M92/41, University of Berkeley, CA, 1992

[14] Tsui C.-Y., Pedram M., Despain A.M.: Low-Power State Assignment Targeting Two- and Multilevel Logic Implementations. IEEE Trans. CAD, vol.17, N 12, December, 1998, 1281-1291

[15] Veeramachaneni V., Tyagi A., Rajgopal S.: Re-encoding for Low Power State Assignment of FSMs. Int. Symp. on Low Power Design, April, 1995