STATE ENCODING TECHNIQUE FOR LOW POWER FSM

Elena Fomina

Abstract: *The problem of minimizing the power consumption in synchronoussequential circuits is explored in this paper. The new technique to solve the state encoding problem targeting for low power Finite State Machnime (FSM) is proposed. The technique use the probabilistic model of an FSM to obtain state encoding that minimize the average number of signal transitions on the state lines for a general State Transition Graph (STG). The method constructs a set of weakly crossed edge cuts to form a set of encoding partitions. The technique has been applied to the MCNC benchmark circuits and has also been compared to other approaches aimed at reducing the switching activity of the state variables.*

Key words: *FSM, STG, State Encoding, Dynamic Power Dissipation, Low Power.*

INTRODUCTION

The synthesis of circuits with reduced power consumption has grown more and more over the last years. Present research focuses on minimizing the power consumption of synchronous FSM, which is an important part of many VLSI products [6]. Since the circuit, realization of an FSM is mostly determined by the state encoding, the encoding can justly be assumed to have a great influence on power dissipation. The goal of proposed approach to the state encoding for a low power FSM is to find a state encoding that minimizes the number of state variables that change their value when an FSM moves between two adjacent states.

Recent power-oriented state encoding works are targeted at reducing the switching activity during the state encoding step of the low-power design of the combinational circuits. The approach presented in [2] is a column based approach that tries to assign the binary value to each state in the STG of an FSM so that the switching activity will be minimal for the complete assignment is presented. Authors developed algorithms trading off accuracy vs. computational complexity. Cost functions are integrated for state register and transition logic activity in [10]. The problem of state encoding for an FSM is separated in two logical steps in [1]. First, the heuristic technique to visit the STG to assign a priory to the symbolic state is presented. Second, the encoding techniques to assign the binary codes to the symbolic states to reduce the switching activity of state register are developed. In [8] the state encoding problem is formulated as an embedding of the spanning tree into a Boolean hypercube of unknown dimension. Several other low power state assignment approaches have been proposed in [3], [5], [7].

All the above-mentioned methods [1], [2], [8] use the probabilistic description of an FSM modeled as a Markov chain [6] and a STG of an FSM as a starting point. Moreover, common to these approaches is also using of edge-weighted graphs. The algorithms try to give states that are linked by high-weight edges the same value for most state variables, while ensuring that each state has a unique code. To achieve the optimal solution, the well-known Gray Codes are used in [1] and [8]. State assignments [1] and [2] give the minimum possible number of state variables to keep the size of the combinational part small, while in [8] the limitation of predetermined number of bits is removed.

The paper is organized as follows. The forthcoming section consists of preliminaries. Then follows the description proposed state encoding technique. Then we present we present the “Weakly Crossed Edge Cuts” state encoding algorithm. How the developed algorithm works is illustrated in detail in the example section. Finally, the conclusions and future directions are reported in the last section.

PRELIMINARIES

An FSM is a model for representation of sequential Boolean functions. An FSM is conveniently described by an STG where nodes represent the states, and directed edges, which are labelled with inputs and outputs, describe the transition relation between states. Each state corresponds to a binary vector stored in the state register. The combinational logic computes the next state and output function, which is based on the current state and input values.

For sequential circuit modelled by an FSM, power dissipation is proportional to the average switching activity [2]:

$$P = 0.5V_{dd}^2 \times f \times \sum_i C_i E_i \tag{1}$$

where V_{dd} is the supply voltage, *f* is the clock frequency of the state machine, C_i is the capacitance of the latch storing state bit *i* and E_i is switching activity of the latch.

For the purpose of minimizing (1), we try to reduce the parameter E_i by concentrating on the state encoding problem whose solution determines the register configuration.

The state encoding problem is an optimization problem whose solution can me measured in terms of a cost function and such that the cost function attains a minimum value [6]. The power-oriented cost function consider the sum of the Hamming distances $H(c_i, c_j)$ between the codes c_i, c_j being assigned to all pairs of states s_i, $s_j \in S$ among which a transition can occur [1], [8]:

$$F_{\cos t} = \sum_{i,j \in S} p_{ij} \cdot H\left(c_i, c_j\right) \tag{2}$$

where p_{ij} is the probability of transition between states *i* and *j*. These probabilities (2) can be determined stochastically by assuming equiprobability of all input patterns of the FSM and solving the Chapman Kolmogorov equations [2], [6]. Alternatively, they can be obtained statistically by applying a sufficiently long series of input patterns until the state occurrence and transition probabilities converge towards discrete values [8].

PROPOSED STATE ENCODING TECHNIQUE

In this section, we introduce the greedy strategy to solve the state encoding problem of an FSM. Synthesis of an FSM is a process of producing an implementation starting from a behavioural specification of a sequential function. We assume that the starting point is an STG. We use the probabilistic model of the FSM described in [1]-[6]. Given the FSM description and the input probabilities, we compute the total state transition probabilities for each edge in the STG, by modelling the FSM as a Markov chain. Then all the unreachable states and self-loops are eliminated from the graph. The STG is transformed into an undirected graph by converting all multiple-directed edges into a single undirected edge.

We weight the received undirected graph twice. The first graph is a vertex-weighed graph (the weight of the vertex $p(v_i)$ is the steady state probability, P_i) while the second graph is an edge-weighted graph (the weight of the edge $p(e_i)$ is the total transition probabilities, $P_{ij}=P_i \cdot p_{ij}+P_j \cdot p_{ji}$) [1], [2].

Further, the proposed state encoding technique is described in terms of the adjacency matrix and the incidence matrix [4]. These matrixes are formed by considering the weighs of vertices and edges. The adjacency matrix is reorganized by sorting $p(v_i)$ in decreasing order and the incidence matrix is reorganized by sorting $p(e_i)$ in decreasing order. This will be the starting point for the state encoding algorithm.

The basic idea of our approach is an economical covering of the set of transitions by weakly crossed edge cuts. To form such edge cuts we construct a set of two blocks partitions called encoding partitions on the set of states of an FSM.

An *encoding partition* π on *S* is a collection $\pi=\{B_1(s), B_0(s)\}$ of two disjoint subsets $B_1 \cap B_0=\varnothing$ of *S* whose set union is *S*: $B_1 \cup B_0=S$ (B_1 is the unit block and B_0 is the zero block).

The number of encoding partitions corresponds to the code length. We consider the minimum number of state variable to find a set of distinct codes.

The goal is to assign codes with minimum Hamming distances to states with higher total transition probabilities.

When the set of encoding partitions is formed, and the set of corresponding edge cuts is constructed, we analyze the effectiveness of the encodings. The number of complicated transitions corresponds to the number of all transitions, which have Hamming distance more than 1. The number of redundant switches is calculated as difference between the total number of switches during proposed encoding and the number of unavoidable switching. Finally, the defect of the encoding is a relationship between power-oriented cost function of the proposed encoding and the optimal encoding.

"WEAKLY CROSSED EDGE CUTS" ECODING ALGORITHM

The STG representation of an initial FSM is given: ($G(V, E)$), $V=(v_1, v_2, \ldots, v_n)$ – set of vertices and $E=(e_1, e_2, \ldots, e_m)$ – set of edges.

The preliminary step of our algorithm is forming of the adjacency matrix and the incidence matrix. To the adjacency matrix the column of degree of vertices $d(v_i)$ is added.

The algorithm starts with a construction of a set of encoding partitions. The number of encoding partitions is equal to $k=\lceil log_2 n \rceil$, where n is a number of vertices in STG and k is a code length. Each encoding partition π_r consists of two blocks: the unit and the zero blocks that are represented by variable sets B^1_r and B^0_r respectively, $1 \le r \le k$. A product partition π_s (the variable parameter that reflects the current state of the decision) is equal to unit partition $\pi_s=\pi_1$,

On the first step of the algorithm, sets B^1_r and B^0_r are empty at the beginning; the first edge (v_i, v_j) from the weighted incidence matrix is selected as a starting point. Vertices of the selected edge (v_i, v_j) are placed in the set B^1_r, so the latter takes value $\{v_i, v_j\}$.

The procedure of calculation the edge cut weight increment by adding vertex v_c to the set B^1_r. For each undistributed vertices v_c this increment is calculated by the following way:

$$\gamma^1\left(B^1_r, v_c\right) = d\left(v_c\right) - 2\left|N'\left(B^1_r, v_c\right)\right| \tag{3}$$

where $d(v_c)$ is the degree of the vertex v_c and $N'(v_c)$ is the set of neighbors for the vertex v_c, consists of all vertices from the set S adjacent to v_c but not including v_c, with respect to the set B^1_r. By using the weighted adjacency matrix the first vertex v_c with the minimal increment (3) is selected. The selected vertex v_c is added to the set B^1_r, so it becomes equal to $\{v_i, v_j, v_c\}$.

Normally, the procedure of calculation of the increment by adding vertex v_c to the set B^1_r, choosing the best vertex and extending the set B^1_r repeats until a cardinality of B^1_r reaches approximately the half of a cardinality of V. The last vertex v_l in the set B^1_r is checked on closeness to both sets B^1_r and B^0_r.

Then the sum of the increment $\sum_l \gamma^{1/0}\left(B^{1/0}_r, v_l\right)$ of vertices v_l to the sets B^1_r or B^0_r is calculated. If $\sum_l \gamma^1\left(B^1_r, v_l\right) \le \sum_l \gamma^0\left(B^0_r, v_l\right)$ then vertex v_l is added to the set B^1_r, otherwise v_l is added to the set B^0_r. All rest undistributed vertices are placed in the set B^0_r. An encoding partition π_r has the unit block equal to the set B^1_r and the zero block equal to the set B^1_r. From the weighted incidence matrix, all rows that correspond to the vertices of the set B^0_r are deleted. The rest rows (rows that correspond to the vertices of the set B^1_r) are summed component-wise by modulo two. The set of edges marked with 1 in the resulting Boolean vector form the edge cut of an encoding partition π_r.

On the next step of the algorithm $r=r+1$; $\pi_s=\prod_r \pi_r$. First $B^1_k=\{v_i, v_j\}$, where (v_i, v_j) is the first edge from the previous edge cut. Then both procedures are called one by another. It is repeats while $r \le k$.

The concluding step of the algorithm is checking the condition $\pi_s=\pi_0$; and computation of the defect of proposed encoding, defining as: $\sum_i p(e_i) \cdot H(e_i) / \sum_i p(e_i)$, where

$p(e_i)$ is the probability of edge e_i, and $H(e_i)$ is the Hamming distance between the codes of the vertices of the edge e_i.

EXAMPLE

Let us illustrate the proposed approach generating a 4-bit code for an 11-state FSM *train11*, which has been selected from the MCNC benchmark suite [9].

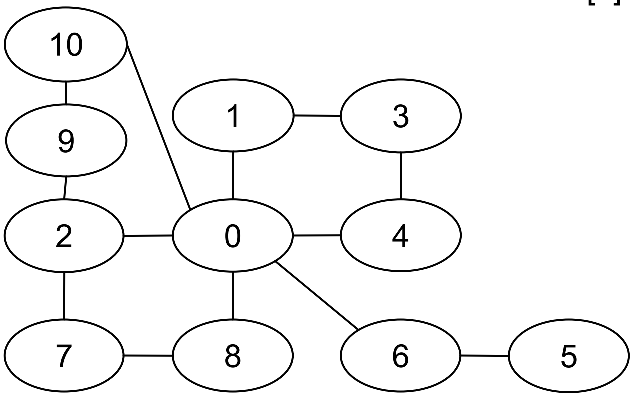

Figure 1. Undirected graph for *train11* FSM

$G(V, E)$: $V=\{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$, $E=\{e_{0,1}, e_{0,2}, e_{0,4}, e_{0,6}, e_{0,8}, e_{0,10}, e_{1,3}, e_{3,4}, e_{5,6}, e_{7,8}, e_{2,7}, e_{2,9}, e_{9,10}\}$

We start from the edge $e_{0,1}$ because this edge has the maximal weight in the weighted incidence matrix $p(e_{0,1})=\max p(e_{i,j})$, $B^1{}_1=\{v_0, v_1\}$. Then, we test all undistributed vertices on the increment by adding the next vertex to the set $B^1{}_1$. Thus, for the vertex v_3 we have $\gamma^1(B^1{}_1,v_3)=\mathbf{2}-2\cdot\mathbf{1}=0$ (3), where **2** is the degree of vertex v_3 and **1** (edge $e_{1,3}$) is the number of neighbours for the vertex v_3 with respect to the set $B^1{}_1$; $B^1{}_1=\{v_0, v_1, v_3\}$ (Figure 1). Several other vertices (v_4, v_6, v_8, v_{10}) have the same value of the increment, 0, but we select the first vertex from the reorganized adjacency matrix, vertex v_3. The next selected vertex is vertex v_4: $\gamma^1(B^1{}_1,v_4)=\mathbf{2}-2\cdot\mathbf{2}=-2$ (3). In that way, we form two sets $B^1{}_1=\{v_0, v_1, v_3, v_4, v_6, v_5, v_8\}$, and $B^0{}_1=\{v_2, v_7, v_9, v_{10}\}$, which are the blocks of the first encoding partition π_1. The first edge cut is $\{e_{0,2}, e_{7,8}, e_{0,10}\}$; the edge cut was constructed by using the reorganized incidence matrix.

On the second step we start with $B^1{}_2=\{v_0, v_1, v_2\}$, because the edge $e_{0,2}$ is the first edge from the first edge cut. Then, $B^1{}_2=\{v_0, v_1, v_2, v_3, v_4, v_9, v_{10}\}$, and $B^0{}_2=\{v_5, v_6, v_8, v_7\}$, which are the blocks of the second encoding partition. The second edge cut is $\{e_{0,8}, e_{0,6}, e_{2,7}\}$.

On the third step, we start with $B^1{}_3=\{v_0, v_1, v_2, v_8\}$. Then, $B^1{}_3=\{v_0, v_1, v_2, v_8, v_7, v_9\}$, and $B^0{}_3=\{v_5, v_6, v_3, v_4, v_{10}\}$, which are the blocks of the third encoding partition. The third edge cut is $\{e_{0,4}, e_{0,6}, e_{0,10}\}$.

On the last step of our example, we start with $B^1{}_4=\{v_0\}$ and $B^0{}_4=\{v_1\}$. Then, $B^1{}_4=\{v_0, v_2, v_4, v_6, v_7, v_8\}$, and $B^0{}_4=\{v_1, v_3, v_5, v_9, v_{10}\}$, which are the blocks of the fourth encoding partition. The fourth edge cut is $\{e_{0,1}, e_{3,4}, e_{5,6}, e_{0,10}\}$.

As a result we have four encoding partition for the FSM *train 11*:

$\pi_1=\{(0,1,3,4,5,6,8), (2,7,9,10)\}$,

$\pi_2=\{(0,1,2,3,4,9,10), (5,6,7,8)\}$,

$\pi_3=\{(0,1,2,7,8,9), (3,4,5,6,10)\}$,

$\pi_4=\{(0,2,4,6,7,8), (1,3,5,9,10)\}$.

Figure 2 demonstrates all constructed edge cuts for the FSM *train 11*. As we can see the edge $e_{0,6}$ is crossed by the set of edge cuts twice and the edge $e_{0,10}$ is crossed thrice. It is means that transitions between the states, presented by vertices v_0, v_6, and v_{10}, are complicated and the Hamming distance between these states is equal to 2 and 3 correspondingly. The state 0 has the code 0000, the state 6 has the code 0110, and the state 10 has the code 1011.

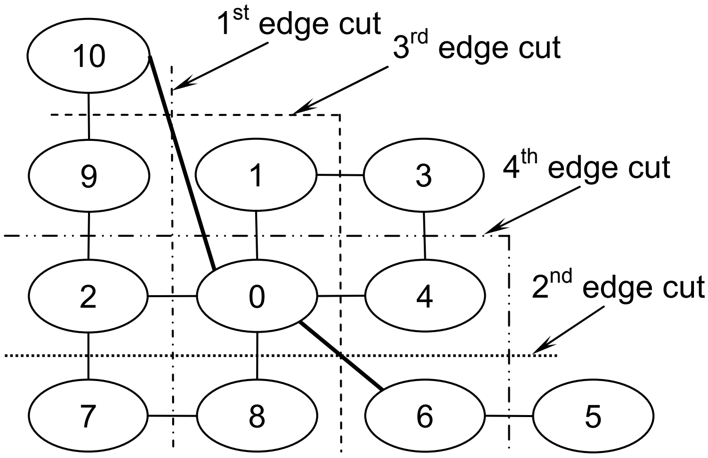

Figure 2. The set of weakly crossed edge cuts for the FSM *train11*

Total number of redundant switching is 2. Thus, the defect of the proposed encoding is equal $(\sum_i p(e_i)\cdot H(e_i)/\sum_i p(e_i)\ -1)\cdot 100\%=(1.417/1.249-1)\cdot 100\%=13.3\%$.

EXPEREMINTAL RESULTS

The proposed encoding technique was implemented and has been applied to MCNC benchmark circuits [9] with aim to compare the results with state-of-the-art encodings. To compare, we select three low power FSM encoding methods: One Level Tree (OLT) [1], POW3 [2], and Spanning Tree Based (STB) [8]. A small part of the results of the defect of encoding is summarized in Table 1.

Table 1 Comparison among several encoding methods

benchmark	WCEC	OLT	POW3	STB
bbara	**24.0**	24.6	25.0	26.0
beecount	**5.3**	5.5	5.5	5.4
dk14	**34.1**	44.8	44.8	**34.1***
dk15	**19.8**	**19.8**	**19.8**	**19.8***
dk17	**22.7**	23.5	23.5	23.4
dk27	**16.3**	17.8	19.8	19.8
ex4	**10.2**	12.1	14.2	11.4
log	**4.0**	-	4.7	**4.0***
opus	**30.8**	-	32.4	32.4
s8	**14.4**	29.3	33.7	14.5*
train11	**13.3**	16.8	15.7	13.5

The results confirm that register switching activity and power consumption are highly correlated.

For most of the benchmarks, WCEC algorithm produces circuits with less switching activity. We found that for some benchmarks (*dk14*, *dk15*, and *log*) our algorithm gives the same defect as STB approach; however, we receive such results with smaller code length. The proposed algorithm found optimal decision (the defect of encoding is equal to 0) for benchmarks *bbtas*, *kirkmann*, *lion*, *lion9*, *mc*, *modulo12*, *tav*, and *train4*, which we do not report in the summary table due to the lack of space.

Globally, the reported results show that the proposed encoding method gives an average of 22% better transition activity with respect to POW3 and OLT encodings; and an average of 5% better transition activity with respect to STB encoding. In STB encoding, the limitation of a predetermined number of bits is removed; therefore, the defect of encoding is lower than in WCEC, POW3, and OLT encodings.

CONCLUSIONS AND FUTURE WORK

A new state encoding technique for a low power FSM is presented. The technique based on the concept of weakly crossed edge cuts for an STG in order to minimize the switching activity. The algorithm was implemented and was run on standard benchmark circuits. The results were compared with existing state encoding tools. Results confirm that proposed state encoding technique can be used for synthesis of an FSM towards low power dissipation.

As one of possible future direction to improving the proposed state encoding targeting at a low power FSM can be underlined the task of finding an optimal state encoding due to insignificant increasing of code length.

ACKNOWLEDGEMENTS

This work has been supported by the Estonian Science Foundation under Grant 5643.

REFERENCES

[1] Baccheta P., L. Daldoss, D. Sciuto and C. Silvano, “Lower-Power State Assignment Techniques for Finite State Machines”, IEEE International Symposium on Circuits and Systems (ISCAS’00), 2000, pp.II-641-II-644.
[2] Benini L. and G. De Micheli, “State Assignment for Lower Power Dissipation”, IEEE Journal of Solid State Circuits, vol. 30, no 3, 1995, pp. 258-268.
[3] De Micheli G., R. Brayton, A. L. Sangiovanni-Vincentelli “Optimal State Assignment for Finite State Machines”, IEEE Transactions on Computer-Aided Design, Vol. CAD-4, no.3, pp 269-285, 1985.
[4] Douglas B. West, “Introduction to graph theory” (2ed). Upper Saddle River: Prentice Hall, 2001.
[5] Koegest M., S. Rülke, G. Franke and M. J. Avedillo, “Two Criteria Constraint-Driven FSM State Encoding for Lower Power”, IEEE, Trans. on Computer-Aided Design, pp.94-99, 2001.
[6] Macii E., “Sequential Synthesis and Optimization for Lower Power”, Lower Power Design in Deep Submicron Electronics, NATO ASI Series, Series E: Applied Sciences, vol. 337, 1997, pp. 321-353.
[7] Martinez M., M. J. Avedillo, J. M. Quintana, M. Koegest, S. Rülke, and H. Suesse, “New Lower Power State Assignment Approach”, Design of Circuits and Integrated Systems Conference (DCIS’00), pp. 181-187, 2000.
[8] Nőth W. and R. Kolla, “Spanning Tree Based State Encoding for Lower Power Dissipation”, Technical report, Dept. of Computer Science, University of Wűrzburg, 1998.
[9] McElvain K., LGSynth’93 Benchmark Set: Version 4.0,1993.
[10] Tsui C-Y, M. Pedram and A. M. Despain, “Lower-Power State Assignment Targeting Two- and Multilevel Implementation”, IEEE Trans. on Computer-Aided Design, vol. 17, no 12, pp.1281-1291, 1998.
[11] Закревский А.Д., Потосин Ю.В., и Черемисинова Л. Д., “Основы Логического проектирования”, Минск, 2004.

ABOUT THE AUTHOR

Researcher, Elena Fomina, PhD student, Department of Computer Engineering, Tallinn University of Technology, Phone: +372 620 22 55, E-mail: elfom@staff.ttu.ee.