

ILP-based Computer-aided Testing and Optimization of Embedded Core

G. Rohini

St. Joseph's College of Engineering,
Chennai – 600 119

S. Salivahanan

SSN College of Engineering,
Kalavakkam – 603 110

ABSTRACT

Low power consumption in a digital circuit can be achieved by partitioning it into sub-circuits which can be turned off at their inactive states. State encoding process can also reduce the power requirement considerably. Switching activity in a sequential logic circuit can be effectively reduced by Clock-gating techniques. In this work, the given circuit is modeled as finite-state machine and its decomposition is formulated into submachine as an Integer Linear Programming (ILP) problem. The power consumption is further reduced by a simple, but powerful state encoding method. The original circuit is decomposed into two structural sub circuits. Computer Aided Testing (CAT) environment is strategically applied in testing each of this sub circuit successively. The average power consumption resulted from the switching activity time interval is minimized by the process of partitioning the circuit and test session planning. In order to reduce the average switching activity, the small cluster of states having high stationary state probability is sought out, for creating the small sub-finite state machine.

Keywords

Computer aided testing, decomposition, finite-state machine, integer linear programming, low power, partitioning, sequential logic decomposition, system on-chip.

1. INTRODUCTION

The power dissipation is considered to be a crucial constraint in the Integrated Circuit (IC) design, especially with the ever increasing scale of integration and clock frequency. As a result, the power consumption turned out to be a major design parameter for ICs. Low power consumption and low heat dissipation are the two independent aspects which are considered [1]. One should give importance to low power consumption aspect to achieve longer autonomy for portable devices. High circuit density and clock frequencies are the factors which create heat dissipation problems that ultimately accounts for expensive packaging and reliability. Study has been made in techniques evolved in transistor and gate levels to the operating system and application levels to reduce system power consumption [1] [2].

In this work, the issue of optimizing logic-level sequential circuits for low power is also considered. Many techniques have been proposed for state assignment issue to reduce the average switching activity of the present state lines and ultimately lead to internal nodes in the combinational logic block [3]. The total amount of glitches in the sequential circuit has been minimized by distributing the registers within the logic block using retiming technique [4]. By turning off the part of the circuit

inactive, the power dissipation can be reduced effectively for the sequential circuit, which is facilitated by Finite State Machine (FSM) approach. A set of submachines, M_i , are formed by partitioning the states of the target FSM and individual submachines implement the functions of M in one state partition. Switching activities involved in the state transitions are reduced by activating one submachine and inactivating remaining submachines [5]-[8].

The study of different structures of FSM decomposition has been made. In one approach, submachines compute their own state transitions and maintain their own states. Circuit structure of this type of decomposition is shown in Figure 1. This approach is called Sequential Logic Decomposition (SLD). In the next approach, state update and state transition computation are separated. Circuit state maintenance and submachine activities scheduling are performed using dedicated logic [9]. Submachines in this approach are used to just compute the next states and outputs [2]. The target FSMs are decomposed incrementally using generic or heuristic algorithms, and are only able to explore a limited design space [5]-[8]. Study for many decades has been made in FSM encoding. Flexibility which is possible in submachine state encoding further reduces the circuit power consumption. The main aim of the encoding algorithm is to reduce the area and to achieve the low power consumption. The difference between the submachine obtained by decomposition and that of the traditional FSMs is that the former has transitions to and from other submachines. Hence, the submachines are made dependant thereby the state assignment of one submachine affects the state assignment of other submachines [10] [11].

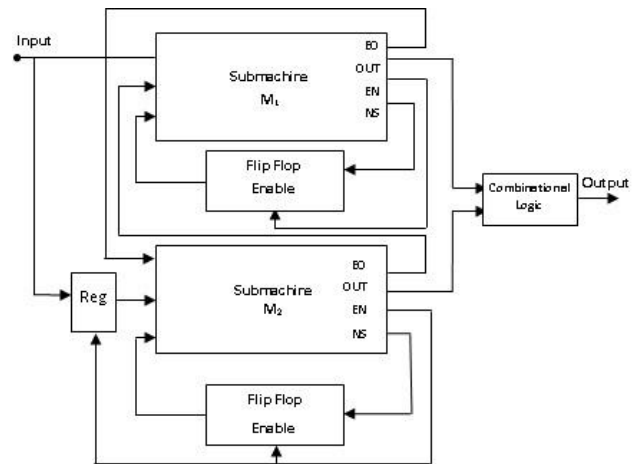


Fig. 1 Decomposition of Sequential Logic Circuit using FSM

In this paper, with power minimization as the objective, FSM decomposition is formulated as an Integer Linear Programming (ILP) [12]. The organization of the paper is as follows: Energy and Power Modeling with notations is presented in Section 2, ILP based decomposition of FSM is described in Section 3, Estimation of Power for sequential circuits is described in Section 4, Estimation Results are summarized in Section 5 and Conclusion of the proposed approach is given in Section 6.

2. ENERGY AND POWER MODELLING

Dominant source of power consumption is dynamic power for the current CMOS technology although this may be replaced in future by high scaled integration.

If the equivalent output capacitance is C_i and the power supply voltage is V_{DD} , then the average energy consumed at node i per switching is $\frac{1}{2}C_i V_{DD}^2$. If s_i is the number of transitions during the period, then good approximate of the energy consumed in that period is $\frac{1}{2}C_i s_i V_{DD}^2$. Higher parasitic capacitance nodes are those with more than one gate connection. By considering this fact, capacitance C_i is assumed approximately proportional to the fanout of the node F_i . During one clock period, the estimation of energy consumed E_i at node i is $\frac{1}{2}s_i F_i C_o V_{DD}^2$, where C_o is the minimum size parasitic capacitance of the circuit [12]. Accordingly, the energy consumed in the circuit after subjecting to a pair of input vectors (V_{k-1}, V_k) successively is expressed by

$$E_{V_k} = \frac{1}{2} C_o V_{DD}^2 \sum_i s(i, k) F_i \quad (1)$$

where i ranges all the nodes of the circuit and $s(i, k)$ is the number of transitions provoked by V_k at node i . Instead of a pair of input vectors, large number of vectors known as pseudo random test sequence can be considered to test the fault. This test sequence which has a test length of $Length_{test}$ is required to achieve the targeted fault coverage [12]. By applying sequence of test vectors, the total energy consumed in the circuit changes to

$$E_{V_k} = \frac{1}{2} C_o V_{DD}^2 \sum_{V_k} \sum_i s(i, k) F_i \quad (2)$$

3. ILP - BASED FSM DECOMPOSITION (FSMD)

The FSM partitioning is formulated as Integer Linear Programming (ILP) problem. Relatively large number of problems can easily be reduced to ILP. The main advantage of using an ILP solver is that an exact solution is found.

Here main aim is to maximize the isolation of circuit components by minimizing the communication between the partitioned FSM. The overall power dissipation is reduced by maximizing the number of components which are put to sleep [13]. Figure 3 gives the ILP-based partitioning technique proposed for the FSM component of the circuit shown in Figure 2. The state transition probabilities used in ILP formulation are difficult to obtain in practice.

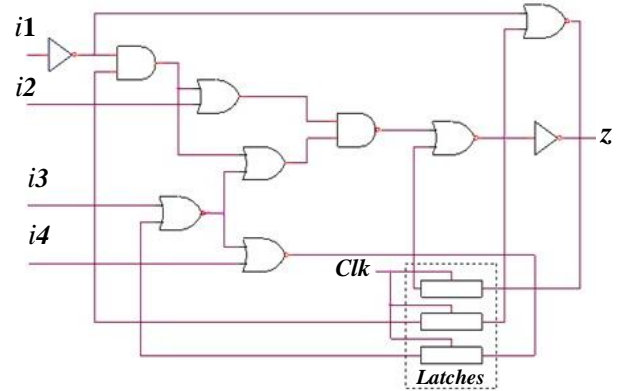


Fig. 2 RTL Schematic for S27 circuit

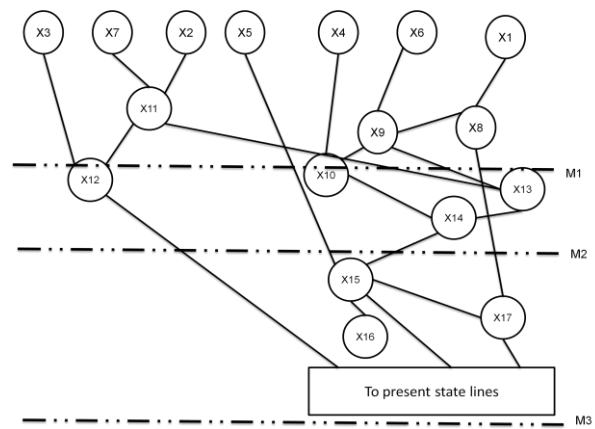


Fig. 3 FSM Decomposition of S27 circuit

3.1 Partitioning of FSMD

Prior to synthesis, FSMD partitioning technique is proposed at the behavioural level. The decomposition of the FSM subunits is made at this level. Only one of these units is powered at a time. This results in power saving in both static and dynamic levels. During the activation of each of these partitions, the registers which are the data components related to that particular partition are also activated [13] [14].

The data components are isolated to the extent that is feasible, so that they can be turned off as long as possible. The newly activated FSM should be communicated with the updated values when the data components are shared between FSM. The power dissipation should be minimized when there is a communication overhead. The proposed FSM partitioning with high level architecture is shown in Figure. 4 [14].

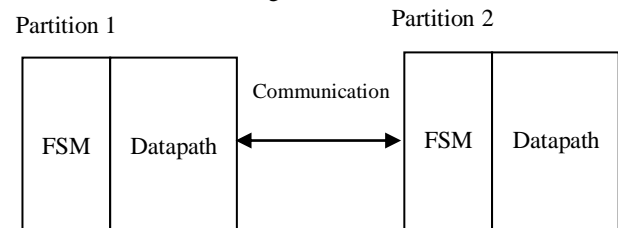


Fig. 4 Partitioned FSM

Minimizing the number of transitions between partitions is another criterion to be considered. Whenever a transition occurs, there will be a communication penalty and also a startup delay, resulting in charging of the capacitances of the newly activated FSM. The communication penalty can be reduced by a lookahead mechanism. In order to reduce the amount of shared data components, the FSM is efficiently partitioned which minimizes these effects of communication penalty. An ILP approach is used to achieve an efficient partition. The number of shared data components between the partitions and the number of possible transitions between the partitions is finally minimized [14].

3.2 Problem Formulation

Standard way of defining FSM P, is 6-tuple format which can be expressed as

$P = \{I, S, \delta, O, \lambda, S_0\}$,
 where

- I is set of inputs
- S is set of states can be considered as S_0, S_1, \dots, S_n
- δ is state transition function depends on the pattern of inputs
- O is set of outputs
- λ is output function which depends on inputs and present state in the case of Mealy machine and depends only on present state in the case of Moore machine.
- S_0 is initial state.

In general, a system can be modeled as a graph. A State Transition Graph (STG) can be used to represent a FSMD. The STG of P can be identified as a graph $G_p(V_p, E_p)$, where V_p is the set of nodes, representing the state set of P, S, and $E_p = \{(u, v), u, v \in V_p\}$ is the set of edges representing the state transition set of P [15].

The partitions of P are a subset of S, along with the transitions related to the states in S. Further, the structures for coordination and communication between the partitions are required, by maintaining the functionality. The machine P is partitioned into sub-machines P_k so as to minimize the interaction between these partitions. If the number of partitions is M, then the set of partitions can be specified as $P_k \forall k \in [1, M]$ [15].

The set of variables that are shared between the different states has to be determined after partitioning the finite state machine. A variable, $v \in VAR$, is considered to be shared between two different states s_i and s_j if the variable is read or written in these states. The bits related to these variables are referred to as transition bits. Hence the total number of transition bits between the states s_i and s_j is represented as T_{ij} [15].

The set of edges between various nodes in STG is represented as a binary variable E_{ij} . The bit 1 is assigned to E_{ij} , if there is an edge from state s_i to s_j . Similarly, the bit 1 is assigned to states s_i and s_j when they are available in same partition.

The objective function can be expressed as

$$\min \left[\sum_{i,j=1}^N T_{ij}(1 - s_{ij}) + \lambda \sum_{i,j=1}^N E_{ij}(1 - s_{ij}) \right] \quad (3)$$

where N is the total number of states in a machine, P. The transition bits between various partitions are represented by the first summation term and the edges between partitions are represented by the second summation term. The edge weight between different partitions is stated as

$$\lambda = \sum_{v \in VAR} |V| \quad (4)$$

which gives the sum of all register bits in the original partition P, because all register bits may need to be communicated from one partition to the other. There are two constraints on the ILP, namely, quality constraints and correctness constraints. The quality constraints help to guide for a useful solution and the correctness constraints ensure the variables have consistent values [15].

4. ESTIMATION OF POWER

Figure 5 shows the counter timing diagram after partitioning the sequential circuit. Partitioning is applied to S298 circuit and its counter alone is represented in the form of timing diagram [14].

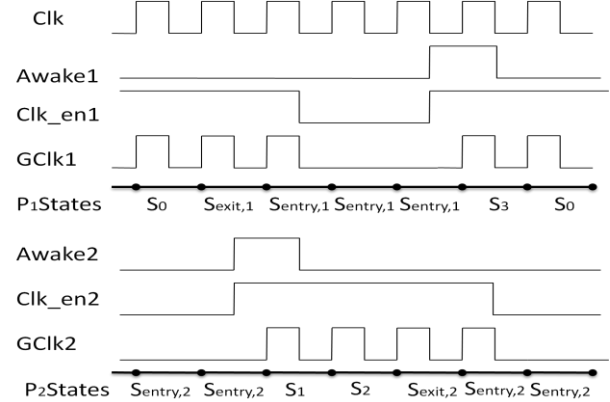


Fig. 5 Counter timing diagram after partitioning [14]

Estimation of power dissipation is necessary to avoid complicated and expensive design. The total power savings can be estimated only after partitioning the circuits. The power dissipation is more before partitioning than that of after partitioning. Power saving always depends on both Static Power (SP) and Dynamic Power (DP). This will be more in Unpartitioned Circuit (UPC) because all the parts of the circuit are participating in action. Since some parts of the Partitioned Circuit (PC) are kept inactive, the power dissipation gets reduced. SP is proportional to number of memory elements in the circuit. To achieve SP savings, a few number of memory elements is kept inactive. DP is proportional to switching activity. DP savings can be achieved by reducing the number of bits in the register after partitioning.

SP savings have been estimated between 20% and 40% and DP Savings between 10% and 30% [14].

5. IMPLEMENTATION RESULTS

The main functions of the CAT are to generate the test program, to parameterize the wrappers, to insert scan chains whenever necessary, and to generate System-on-Chip (SoC) interface to integrate the cores. The ILP formulation is used to partition the original machines into submachines. Using this formulation, fault coverage has been found and it is shown in Table 1. SP and DP have been estimated for both unpartitioned and partitioned circuits. From Table 2, it can be justified that power dissipation is less in PC compared to UPC.

Table 1. Fault Coverage (FC)

Circuit	Original Circuit		Partitioned Circuit	
	Length	FC (in %)	Length	FC (in %)
C17	20	100	16	100
S27	32	77.77	20	95
S298	16	100	12	100
S382	224	95	203	95
S386	182	97	156	97

Table 2. Estimated Power Savings

Circuit	UPC		PC		Power Savings (%)	
	SP (μW)	DP (μW)	SP (μW)	DP (μW)	SP	DP
C17	265	6.809	203	4.44	23.3	34.7
S27	428	23.422	257	18.14	39.9	22.55
S298	781.5	45.32	462.6	27.96	40.8	38.3
S382	656.32	42.37	401.41	25.25	38.8	40.40
S386	802.27	51.08	511.19	33.33	36.28	34.75

6. CONCLUSION

This paper deals with breaking down the controller and the data path into two or more partitions using techniques such as FSM partitioning method and ILP. Implementing and analyzing an ISCAS benchmark circuit in detail shows that up to 41% power saving is possible. The work is in progress on implementing the proposed FSM partitioning method for the entire set of explored circuits and the measured power results will be reported soon. The amount of power dissipation of SoCs is found to be twice in the test mode when comparing to normal mode. This is because the cores rarely operate in parallel and are tested simultaneously to minimize testing time. Thus it is essential to schedule tests that are power constrained by proper test scheduling to make sure that the power rating of the SoC is kept within safe limits. This also constricts the amount of concurrency during the testing period. The original circuit is partitioned into smaller circuits to test them successively using IEEE Std 1500. This research is targeted at the reduction of the average power consumption during the testing period and also to minimize the peak power consumption.

7. REFERENCES

- [1] E. Macii, M. Pedram, and F. Somenzi, "High-Level Power Modeling, Estimation, and Optimization", IEEE Transactions on CAD, Vol. 17, 1998, pp. 1061-1079.
- [2] M. Pedram, "Power Minimization in IC Design: Principles and Application", ACM TODAES, Vol. 1, 1996, pp. 3-56.
- [3] K. Roy and S. Prasad, "Circuit Activity Based Logic Synthesis for Low Power Reliable Operations", IEEE Transactions on VLSI Systems, Vol. 1, No.4, December 1993, pp. 503-513.
- [4] J. Monteiro, S. Devadas, and A. Ghosh, "Retiming Sequential Circuits for Low Power", Proceedings of the International Conference on Computer-Aided Design, November 1993, pp. 398-402.
- [5] L. Benini, G. De Micheli, and F. Vermulen, "Finite State Machine Partitioning for Low Power", Proceedings of International Symposium on Circuits and Systems, 1998, pp. 5-8.
- [6] S.H. Chow, Y. C. Ho, T. Hwang and C. L. Liu, "Low Power Realization of Finite State Machines-A Decomposition Approach", ACM TODAES, Vol. 1, 1996, pp. 315-340.
- [7] J.C. Monterio and A.L. Oliveria, "Finite State Machine Decomposition for Low Power", Proceedings of Design Automation Conference, 1998, pp. 758-763.
- [8] J.C. Monterio and A.L. Oliveria, "FSM Decomposition by Direct Circuit Manipulation Applied to Low Power Design", Proceedings of Asia South Pacific Design Automation Conference, 2000, pp. 351-358.
- [9] Girard P, Guiller L, Landrault C, Pravossoudovitch S, "Circuit partitioning for low power BIST design with minimized peak power consumption", Test Symposium, 1999, Proceedings, Eighth Asian Digital Object Identifier, 1999, pp. 89- 94.
- [10] E. Olson and S. Kang, "Low-Power State Assignment for Finite State Machines", Proceedings of International Symposium on Low Power Design, 1994, pp. 63-68.
- [11] C.Y. Tsui, M. Pedram, and A. Despain, "Low-Power State Assignment Targeting Two and Multilevel Implementations", IEEE Transactions on CAD, Vol. 17, 1998, pp. 1281-1291.
- [12] Feng Gao and John P. Hayes, "ILP-based optimization of sequential circuits for low power", Proceedings of international symposium on Low power electronics and design, 2003.
- [13] E. Hwang, F. Vahid, and Y.C. Hsu, "FSMD functional partitioning for low power", Proceedings of the conference on Design, Automation and Test in Europe, 1999.
- [14] Nainesh Agarwal and Nikitas Dimopoulos, "FSMD Partitioning for Low Power using ILP", IEEE Computer Society Annual Symposium on VLSI, 2008
- [15] H. R. Lewis and C. H. Papadimitriou, "Elements of the Theory of Computation", Englewood Cliffs, NJ: Prentice-Hall, 1981.