POSE: Power Optimization and Synthesis Environment

Sasan Iman, Massoud Pedram

Department of Electrical Engineering - Systems
University of Southern California
Los Angeles, CA 90089

Abstract

Recent trends in the semiconductor industry have resulted in an increasing demand for low power circuits. POSE is a step in providing the EDA community and academia with an environment and tool suite for automatic synthesis and optimization of low power circuits. POSE provides a unified framework for specifying and maintaining power relevant circuit information and means of estimating power consumption of a circuit using different load models. POSE also gives a set of options for making are-power trade-offs during logic optimization.

1. Introduction

In the past, the main objective of designers has been to design faster and denser circuits. In response to this demand, design tools have been developed to help automate the design process for achieving maximum speed and minimum area. These design automation tools have been used extensively in the industry and are an integral part of any design cycle.

With the increased popularity of portable devices, battery size and lifetime are becoming important factors in the design process. At the same time, the amount of data to be processed is increasing at a rapid pace. This also calls for faster digital devices which in turn increases power consumption. The circuit power is also becoming one of the limiting factors in the amount of logic that can be placed in a VLSI chip and the determining factor in the packaging cost. These considerations have resulted in a growing need for minimizing power consumption in today's digital systems.

The demand for low power digital systems has motivated significant research in the area of power estimation and power optimization. Power estimation and optimization techniques have been proposed at all stages of the design process. Power estimation techniques have been proposed by researchers at the gate level [3], [8], [15], [17], [21]. Power optimization techniques have also been proposed at all levels of the design abstraction. Many optimization approaches have been proposed at the behavioral, RT and logic level [2], [5], [9], [10], [16], [19], [20]. Reference [18] contains a detailed survey.

Even though considerable effort has been made in creating new techniques for power estimation and optimization, a unified framework for designing low power digital systems has not yet been developed. The void created by the absence of such a framework has presented designers with serious problems. Optimization algorithms that target low power circuits use the frameworks designed for synthesizing minimum area and delay circuits. This means that the critical information needed for power estimation and optimization is not available when low power techniques are applied. In most cases, minimal information is available to the power conscious design procedures which in turn results in reducing the potentials of these procedures. A more significant problem caused by the lack of a unified framework for low power design is that designers are forced to use low power techniques as isolated procedures. This creates a major obstacle in developing a methodology for effective and efficient power specification, estimation and optimization. The lack of a methodology in turn results in a limited understanding of the applicability of existing techniques which contributes to holding back the state of the art low power technology. At the same time, many optimization techniques are only applicable and relevant when applied in conjunction with other optimization approaches. Therefore without a unified framework, many new techniques will not be discovered.

In this paper we address this challenge by presenting a methodology for designing low power digital circuits at the RT and logic levels. In doing so, we present POSE, the Power Optimization and Synthesis Environment. POSE is the first step in creating a complete and unified framework for design and analysis of low power digital circuits. POSE provides an easy to use interactive environment which is an extension of the familiar environment provided by the SIS package [1].

This paper is organized as follows. In section 2 we present the power model used during optimization. In section 3 we describe the low power design methodology presented in this paper. In section 4 and 5 we discuss issues behind design specification for power and power estimation. Results and conclusions are presented in sections 6 and 7.

2. Power Model

Power dissipation in CMOS circuits is caused by four sources: 1) the leakage current which consists of reverse bias current in the parasitic diodes formed between source and drain diffusions and the bulk region in a MOS transistor as well as the subthreshold current that arises from the inversion charge that exists at the gate voltages below the threshold voltage, 2) the stand-by current which is the current drawn continuously from V_{dd} to ground which happens, for example, when the tri-stated input of a CMOS gate leaks away to a value between V_{dd} and ground, 3)the short-circuit (rush-through) current which is due to the DC path between the supply rails during output transitions and 4) the capacitive current due to charging and discharging of capacitive loads during logic changes.

In well-designed CMOS circuits, the dominant source of power dissipation is due to capacitive currents due to charging and discharging of the gate capacitances (also referred to as the dynamic power dissipation) and is given by:

$$P_i = \frac{1}{2} \cdot V^2 \cdot C_i \cdot f \cdot E_i \qquad (1)$$

where V is the supply voltage, f is the clock frequency, C_i is the capacitance seen by gate n_i and E_i (referred to as the

* This research was supported in part by the Semiconductor Research Corp. under contract no. 94-DJ-559 and by ARPA under contract number F336125-95-C1627.

33rd Design Automation Conference ®

DAC 96 - 06/96 Las Vegas, NV, USA

switching activity) is the expected number of transitions at the output of n_i per clock cycle. The product of C_i and E_i is often referred as the *switched capacitance.*

At the time of logic synthesis, all architectural and technology related issues for the network being optimized have been decided. This means that the values for V and f have already been selected. Therefore the main issues in computing a power estimate are in computing the load and switching activity values for nodes in the network. At the logic level, our optimization goal is also to minimize the total switched capacitance of the circuit.

3. Low Power Design Methodology

A low power design methodology can only be developed when a number of key components are made available to the designer. These components fall into the following categories: *Design Specification*, *Design verification* and *Synthesis Procedures*. Complete design specification is necessary in order to provide the synthesis environment with maximum information necessary for the optimization process. For example, when designing a circuit for minimum area, the design specification for this synthesis process should include a measure of the area for the gates in the target library. Similarly, a synthesis environment for low power should include all power modeling and estimation information that is necessary during the synthesis and validation procedure. An important issue to also consider is that design specification for power may not be the same at different levels of abstraction and therefore a set of consistent specification standards need to be available at different levels. For example, input data activity at RT level may be given in terms of word-level probability density functions while the same information may be provided at logic level in terms of bit-level switching activity. Design specification should describe the possible forms of data and provide means of translation into the other forms.

Design validation is also an important part of any design methodology. Final design has to comply with the design specifications. For conventional logic synthesis, this validation is in the form of checking functional correctness and checking that design meets the area/delay requirements. For power consideration, a design has to meet the target power budget. Power estimation procedures are therefore necessary to check design compliance with the given specifications. At the same time, power estimation is necessary to check the quality of the synthesis steps. Power estimation is also a crucial part of interactive optimization techniques.

Synthesis procedures are the basic steps used to incrementally change the circuit structure while optimizing the target cost function. The combination of these steps guided by the power estimation procedures are used to develop a power optimization methodology.

Figure 1 presents the power optimization methodology used in POSE. The highlighted boxes specify the information that has to be provided by the designer. Input to POSE is a state transition graph describing the finite state machine for the circuit. For FSMs and combinational circuits, input can also be provided as a Boolean network and the statistical information for the primary inputs of the circuit. The set of power information required by design specifications (see section 4) is also provided at this time. State assignment is then used to assign an state encoding to states of the finite state machine. The conventional cost function for state assignment algorithms has been minimum area.

State assignment of a finite state machine has a significant impact on the area of its final logic implementation. In the past, many researchers have addressed the encoding problem for minimum area of two-level or multi-level logic implementations. These techniques can be modified to minimize the power dissipation. An effective approach [19] is to consider the complexity of the combinational logic resulting from the state assignment and to directly modify the objective functions used in conventional encoding schemes such as NOVA [22] and JEDI [12] to achieve lower power dissipation. This approach is used in POSE to perform state assignment. The output of the state assignment program is a Boolean network. A Boolean network is a directed graph where each node represents either a Boolean function or a latch element. The Boolean network is then optimized using a set of logic synthesis operations. The design entry into POSE can also be in form of a Boolean network.

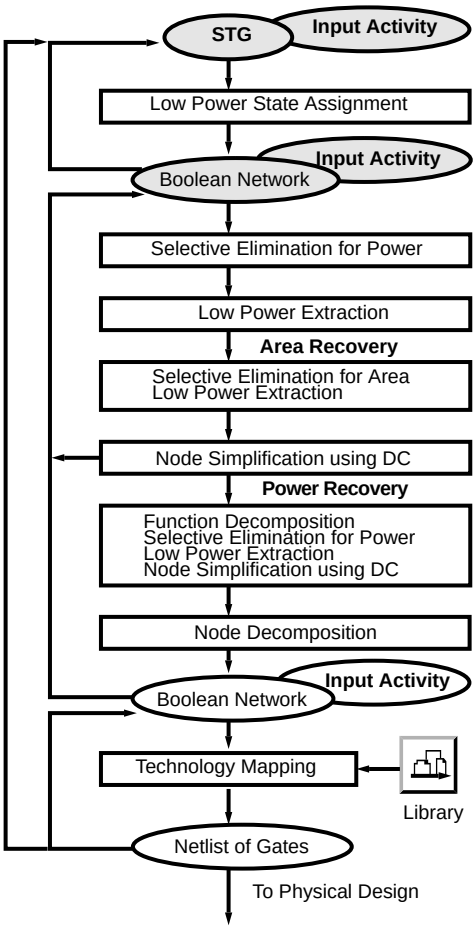


Figure 1: Pose Power Synthesis Methodology

Low power optimization algorithms provided in POSE consist of three categories of optimization techniques: 1)Low power algebraic restructuring techniques, 2) Low power simplification and 3)Low power technology mapping. Algebraic restructuring techniques and function simplification are used during the logic synthesis process while the technology mapping is used to map the optimized networks to gates in the target technology.

Algebraic logic restructuring techniques have been used in the past to minimize the area of a Boolean network by taking advantage of the common sub-functions between different

nodes and within the same node. These operations include common sub-function extraction, function decomposition and factorization. A selective collapse procedure is used to remove nodes from the network which don't contribute to minimizing the cost of the network.

POSE includes operations to perform low power logic restructuring. The low power algebraic operations implemented in POSE are discussed in detail in [10]. The basic approach here is to use extraction, factorization, decomposition and selective collapse operations to reduce load on high activity nodes by introducing new nodes into the network that have low output activity and also removing nodes that are not good candidates for reducing the network power cost.

It should be noted that power optimization techniques are developed to make area-power trade-offs. In the methodology given in Figure 1, the initial logic restructuring have been performed for minimum power. This means that network area may not have maximally decreased during this operation. Therefore the methodology includes a number of operations for recovering some area. This is called "area recovery". A "power recovery" stage is also included before the technology mapping process. The reason for this is that decomposition for minimum power does not maximally decompose all node functions. In other words it leaves some area redundancy in the network that may still be taken advantage of for power optimization. The "power recovery" stage is included to use this flexibility.

After the Boolean network is optimized for power, it is then mapped to the gates in the target technology. This operation will generate a netlist of gates in the library. The goal of low power technology mapping is to generate a mapped network where high activity nodes are hidden inside complex gates. The procedure for low power technology mapping has been discussed in detail in [20]. After technology mapping is performed, it is possible to swap the symmetric inputs to a gate. The power after technology mapping can be further reduced by swapping pins where inputs with high input load are driven by lower activity inputs. By taking advantage of this flexibility, power can be further reduced.

Note that at some points during the optimization, the quality of the results are checked using the power estimation utilities. This is to check how much reduction in power has been obtained after each iteration. The synthesis process is terminated if no further improvement can be obtained or the resulting power estimate is within the design specifications. The next section presents a detailed discussion on the design specification requirement and power estimation.

4. Design Specification for Power

In the past, logic synthesis has concentrated on minimizing the area of a circuit while meeting the timing constraints. The design specification for logic synthesis therefore consisted mainly of providing the functional description of the circuit, the timing constraints and the area/delay characteristics of the target library. As logic synthesis environments are extended to take into account power consumption, the conventional design specification techniques prove to be inadequate. In this section we discuss information that is necessary for effective power estimation and optimization at the logic synthesis stage.

4.1. Input Switching Activity

The power consumption of a CMOS circuit is a function of the expected number of times the logic signals in the circuit change values. Unlike conventional logic synthesis where the circuit performance (area and delay) can be calculated deterministically, the statistical behavior of the input data has a significant impact on the power consumption of the circuit. In order to calculate the circuit power, it is necessary to provide a sequence of bit vectors applied at the primary inputs of the network. This sequence of bit-vectors may in turn be used by *statistical or probabilistic power estimation techniques* to calculate the expected switching rate of gates in a network. POSE relies on the expected behavior of the circuit at the primary inputs in terms of probability values to calculate the expected switching rates of the internal gates in a circuit.

Given a logic value v in a Boolean network, SP^v_1 (SP^v_0), the *signal 1(0) probability* for v, gives the probability that v evaluates to value 1 (0); $TP^v_{x->y}$, the *transition probability* for v, gives the probability that v evaluates to x at time t and evaluates to y at time (t-dt). E_v, the switching activity for v is then defined as ($E_v = TP^v_{0->1} + TP^v_{1->0}$). Assuming temporally independent[1] input vectors, this equation simplifies to:

$$E_v = 2 \cdot SP^v_1 \cdot SP^v_0 = 2 \cdot SP^v_1 \cdot \left(1 - SP^v_1\right) \qquad (2)$$

4.2. Library Load Values

Libraries used during logic synthesis only provide information on area and delay of each gate in the library. More information is however required to accurately measure the power consumption of a gate in a technology mapped network. The power consumption of a gate consists of the power consumed at the output of the gate and the power internal to the gate.

The power at the output of a gate g is a function of the load seen at the output of this gate. This load is a combination of the input loading of the output gates and also the *self-loading capacitance* for the gate itself. The self loading capacitance for a gate is defined as the load driven by the gate when the gate output is left open and is due to the source/drain diffusion capacitances of the gate. Experimental results show that self-loading capacitances contribute up to 20% of the total power consumption in CMOS circuits. Ignoring these capacitances will no doubt affect the accuracy of power estimation and the optimality of power optimization. The *internal power consumption* of a gate refers to the power required to charge and discharge the internal capacitances of this gate. Therefore a power optimization procedure requires knowledge of the internal capacitances (diffusion capacitances) of all gates in the library to be able to compute the power dissipation due to the self-loading capacitance and the internal parasitic capacitances (see Figure 2).

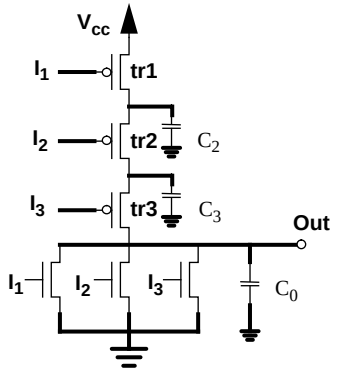


Figure 2: Parasitic Capacitances for a 3 input NOR Gate

1. Temporal independence of variable x implies that value of x at time t is independent of its value at time t+dt

5. Power Estimation

Technology independent power optimization requires a good estimate of the contribution of each node in the technology independent network to the power consumption of the mapped network. Previous methods for technology independent power estimation have used the number of fanouts for a node as an estimate of the load at the output of a node. Experimental results show the number of fanouts in the factored form (see section 5.1) to be a better load estimate for nodes in a technology independent network. At the same time some power optimization algorithms (e.g. algebraic restructuring techniques) operate on the sum-of-products form of nodes. This suggests that technology independent power analysis will be most effective when a number of different load models are considered during technology independent power estimation. This is similar to the area optimization process where a combination of number of literals in the sum-of-products form and number of literals in the factored form are used to minimize the area of the network.

5.1. Load Models

We define the power consumption and power contribution of a node in terms of its input power, internal power and output power. Note that all power values are reported as switched capacitance values where supply voltage and clock frequency are assumed to be fixed.

$$Power(n_i) = InternalPower(n_i) + OutputPower(n_i) \quad (3)$$

$$PowerContribution(n_i) = InputPower(n_i) + InternalPower(n_i) + OutputPower(n_i) \quad (4)$$

The input and output power estimates for a node in a Boolean network are estimated as given below.

$$InputPower(n_i) = \sum_{n_j \in fanins(n_i)} E(n_j) \cdot L(n_j, n_i) \quad (5)$$

$$OutputPower(n_i) = E(n_i) \cdot \left(\sum_{n_k \in fanouts(n_i)} L(n_i, n_k) \right) \quad (6)$$

In this equation $L(n_j, n_i)$ gives load on gate n_j due to its fanout node n_i. We define four load models:1)simple load model, 2)factored form load model and 3)sum-of-products form load model and 4)library load. For simple load model the value $L(n_j, n_i)$ if always equal to 1. For factored form load, $L(n_j, n_i)$ gives the number of times variables n_j is used in the factored form representation of node n_i. For example, in Figure 3, $L(n_1, n_3) = 2$ where the total load in the factored form on node n_1 is equal to 5. For sum-of-products form $L(n_j, n_i)$

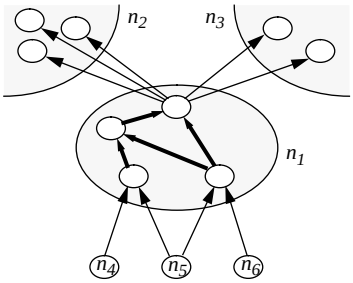


Figure 3: Load in the factored form

gives the number of times variables n_j is used in the sum-of-products form representation of node n_i. For library loads, $L(n_j, n_i)$ is computed using the library parameters.The internal power consumption in the simple load model is assumed to be zero, for factored form load, the internal power is computed by first finding the factored form implementation of the node and then finding the power on the outputs of this factored form implementation. The same technique is applied for sum-of-products form load model. For library load, the internal load is computed using the diffusion capacitance information as discussed in section 4.2.

5.2. Switching Activities Under a Zero-Delay Model

Under a zero-delay model and assuming temporal independence at the circuit primary inputs, the switching activity of each node in the network can be computed by finding the signal probability of the node and then using equation 2 to compute its switching activity. A number of issues need to be considered when computing the signal probability of an internal node. In the following we discuss accuracy-speed tradeoffs for computing node signal probability values. We also discuss how different logic synthesis algorithms may impact the signal probability and therefore the switching activity of nodes in a network.

5.2.1 Speed/Accuracy Trade-offs

The immediate fanins of an internal node are in general spatially correlated[1]. This correlation may be present even if primary inputs are spatially uncorrelated. Under these conditions, the spatial correlation at the immediate fanins of a node *n* is due to the reconvergant fanout regions in the transitive fanin cone of *n*. This means that an exact calculation of the signal probability for an internal node *n* requires that this signal probability be computed using the global function of *n*. BDDs have provided a more feasible approach for representing the global function of nodes in a Boolean network. Reference [4] presents an efficient procedure for computing the signal probability of a function from its BDD representation. Therefore BDD based techniques are a good candidate for computing the signal probability of nodes in a network.

Representing the global function of nodes in some circuits may however become too expensive even when BDDs are used to represent the global functions. Therefore it is necessary to provide a mechanism for making speed-accuracy trade-offs when computing signal probabilities. In the following, we describe and justify our technique for speeding up the procedure for computing these signal probability values.

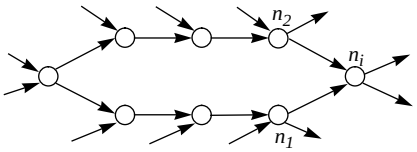
5.2.2 Using Semi-local BDDs

Local BDD for a node *n* is defined as a BDD where immediate fanins of node *n* are used in building the BDD. The semi-local BDD for a node *n* is defined as the BDD where the nodes used as the BDD variables create a cut in the transitive fanin cone of *n*. Note that the global and local BDD for a node are special cases of the semi-local BDDs for that node.

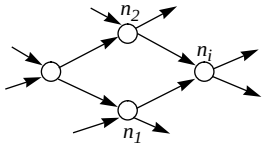
The main reason for using global BDDs when computing the signal probability for a node *n* is to take into account the spatial correlation at the immediate fanins of the node. Figure 4 shows a shallow and a deep Reconvergant fanout region in the fanin cone of node *n*. A shallow Reconvergant region spans over less number of levels than a deep reconvergant region. It can be stated that a shallow reconvergant region, in general, results in more spatially correlated fanins to a node. At the same time, a deep reconvergant region will in general result in lower spatial correlation at the immediate fanins of node *n* due to randomizing effect of the side inputs to the reconvergant path. Therefore it is possible to capture most of the spatial correlations of inputs of a node *n* by using the semi-

1. Spatial correlation between nodes x and y in a network means that the values of x and y in the same clock cycle are dependent.

local BDD for that node. The semi-local BDDs will also take into account the spatial correlation due to shallow reconvergant region. POSE uses semi-local BDDs as a compromise between efficiency and accuracy [7].



a) Deep Reconvergant Fanout



a) Shallow Reconvergant Fanout

Figure 4: Reconvergent Fanout Regions

The definition for deep and shallow regions is implementation dependent. Experimental results however show that in a network that is decomposed into 2-input gates, taking into account Reconvergant regions that span over 5 levels capture a significant part of the spatial correlation at the input of nodes in the network.

Note that as the network is being optimized, the network structure is modified and therefore the definition of semi-local bdds for each node will change. Therefore, if an optimization step changes the network structure drastically, the node switching activity values should be recomputed so that the new activity values reflect the new structure of the network.

5.3. Effect of Optimization on Switching Activities

A correct methodology for low power synthesis requires that the signal probability and switching activity values computed and stored on each node remain correct as the network is being restructured during the logic synthesis process. Most operations on a Boolean network are algebraic in nature. This means that global function of nodes and therefore the zero-delay signal probability of nodes do not change for these operations. Some logic synthesis operations, however, do modify the global function of the node being optimized and other nodes in the network. These operations will result in changing the signal probability of nodes in the network without having modified these nodes. Figure 5 shows an example where simplifying the function of a node using its observability don't care has resulted in changing the signal probability of the node and nodes in its transitive fanout.

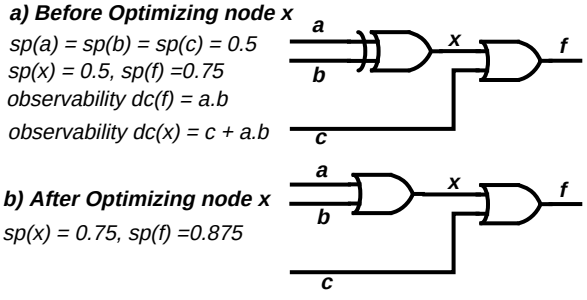


Figure 5: Effect of optimizations on signal probabilities

If switching activity values are estimated using a real delay model, then after changing a node *n*, the delay of the network changes and therefore the switching activity of all nodes in transitive fanout of *n*, and the switching activity of all nodes in the transitive fanouts of the immediate fanins of node *n* change. This is because changing the function of a node, changes the load on its immediate fanins which changes the delay for fanout nodes of the immediate fanin nodes.

6. Results

POSE has been implemented and results have been generated for minimum power circuits. Table 1 gives the power consumption of multi-level circuits in the MCNC benchmark set when they are optimized and mapped for minimum area and when they are optimized and mapped for minimum power. Columns 2, 3 and 4 give the area, delay and power (switched capacitance) of the technology mapped circuits when they are optimized for minimum area. Columns 5, 6 and 7 give the area, delay and power (normalized with respect to values in the first three columns) for the same circuits when they are optimized and technology mapped for minimum power consumption using the methodology presented in section 3. As the results show, POSE has been able to reduce the circuit power consumption on average by 26% at the expense of increasing the area of the circuits by 23%. This clearly shows a trade-off between area and power. The delay of the circuits have only been increased by 4%. Table 2 gives the same statistics for two-level benchmarks in the MCNC benchmark set. The power consumption of the technology mapped circuits has been reduced on average by 29% at the expense of increasing the area on average by 30%. Again, this clearly shows that POSE has been able to successfully find a minimal power solution by trading off area for power.

7. Conclusion

In this paper we have presented a methodology for designing low power digital circuits. POSE has been developed as a unified framework for specifying and maintaining power relevant circuit information. POSE also provides power estimation and optimization procedures which facilitate the low power design methodology.

The emphasis in this paper has been in identifying an effective low power design methodology. In addition POSE has been developed and tested with the proposed methodology where it is shown that the power estimation techniques provided a good method for verifying the quality of the incrementally power optimized circuits. An optimization script has also been developed based on the methodology presented in this paper.

POSE is a first step in developing a complete environment for low power circuit synthesis. Many optimization algorithms at different levels of abstraction are yet to be incorporated into the POSE environment. At the same time, power estimation techniques should also be included at different levels of the design abstraction. In accordance with this philosophy, POSE has been developed to allow researchers to easily incorporate new power estimation and optimization techniques into POSE. This will then allow POSE to be used as a platform for future research on low power CAD tools. The first release of POSE has been made available (http://atrak.usc.edu/~pose) and is being used both as an optimization/estimation tool and as a platform for implementing and experimenting with new low power design techniques.

	Minimum Area			Minimum Power		
Ex	area	delay	power	area	delay	power
C1355	389760	24.4	16.1	1.12	1.16	0.93
C1908	434768	39.0	14.8	1.12	1.13	0.84
C432	170288	43.5	6.65	1.20	0.97	0.84
C5315	1365088	41.5	65.9	1.25	0.99	0.85
alu2	305312	40.4	9.67	1.19	1.20	0.64
alu4	592528	48.7	9.83	1.43	1.17	0.70
b9	110896	9.34	4.17	1.13	1.09	0.78
dalu	760960	50.3	23.8	1.26	1.06	0.68
des	2865200	163	112	1.34	0.78	0.81
frg1	115536	18.9	4.98	1.25	0.86	0.71
frg2	692752	36.4	19.8	1.18	1.13	0.68
i8	797616	39.8	20.5	1.35	0.95	0.68
k2	1011056	32.8	12.3	1.44	1.13	0.65
rot	594848	26.1	22.8	1.20	1.35	0.78
sct	75168	31.4	2.51	1.00	0.44	0.60
t481	612944	30.8	7.89	1.15	0.90	0.68
term1	149408	13.5	5.17	1.03	1.10	0.71
ttt2	193952	17.7	6.32	1.01	0.99	0.61
9symml	159152	22.4	6.99	1.68	1.26	0.77
Average				1.23	1.04	0.74

Table 1.Minimum power solutions for multi-level examples

	Minimum Area			Minimum Power		
Ex	area	delay	power	area	delay	power
apex1	1308944	30.4	31.7	1.33	1.34	0.68
apex5	676512	34.2	12.9	1.40	0.70	0.72
b12	75168	11.0	2.19	1.35	1.10	0.85
bw	130848	33.1	4.41	1.49	0.49	0.67
clip	118320	20.3	5.60	1.17	1.17	0.71
cps	1050960	36.7	22.7	1.34	1.07	0.58
duke2	392544	33.1	10.2	1.13	0.72	0.61
e64	293248	110	4.22	1.00	1.01	0.63
ex4	402288	12.4	16.0	1.39	1.26	0.83
misex1	46864	13.5	1.70	1.43	0.80	0.81
misex2	91872	10.9	2.52	1.34	1.07	0.85
misex3	575360	32.1	15.6	1.40	1.20	0.69
pdc	341040	21.2	12.1	1.40	1.06	0.83
rd73	58928	17.8	1.35	1.25	0.98	0.68
rd84	128992	18.2	4.70	1.24	1.33	0.65
spla	545200	24.2	15.8	1.35	1.15	0.62
vg2	85376	11.0	3.42	1.24	1.47	0.71
5xp1	102544	30.9	4.05	0.99	0.53	0.66
9sym	178640	19.1	9.06	1.48	0.96	0.78
Average				1.30	1.02	0.71

Table 2.Minimum power solutions for two-level examples

8. References

[1] “SIS: A system for sequential circuit synthesis,” Report M92/41, UC Berkeley, 1992.

[2] J. B. Burr. Stanford ultra low power CMOS. In *Proceedings of Hot Chips Symposium V*, pages 583–588, June 1993.

[3] R. Burch, F. N. Najm, P. Yang, and T. Trick. " A Monte Carlo approach for power estimation. " IEEE *Transactions on VLSI Systems*, 1(1):63–71, March 1993.

[4] S. Chakravarty. " On the complexity of using BDDs for the synthesis and analysis of boolean circuits. " In *Proceedings of the 27th Annual Allerton Conf. on Communication, Control and Computing*, pages 730–739, 1989.

[5] A. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-power CMOS design. " *IEEE Journal of Solid-State Circuits*, pages 472-484, April 1992.

[6] A. Chandrakasan, M. Potkonjak, J. Rabaey and R. W. Brodersen, " HYPER-LP: A System for Power Minimization Using Architectural Transformation. "In *Proceedings of the I EEE International Conference on Computer Aided Design*, pages 300-303, November 1992.

[7] C-S. Ding and M. Pedram. "Tagged probabilistic simulation provides accurate and efficient power estimates at the gate level . " In *Proceedings of the Symposium on Low Power Electronics*, September 1995.

[8] A. Ghosh, S. Devadas, K. Keutzer, and J. White. " Estimation of average switching activity in combinational and sequential circuits. " In *Proceedings of the 29th Design Automation Conference*, pages 253–259, June 1992.

[9] S. Iman and M. Pedram. " Multi-level network optimization for low power. " In *Proceedings of the* I EEE *International Conference on Computer Aided Design*, pages 372–377, November 1994.

[10] S. Iman and M. Pedram. " Logic extraction and decomposition for low power. " In *Proceedings of the 32nd Design Automation Conference*, June 1995.

[11] B. Lin and A. R. Newton. " Synthesis of multiple-level logic from symbolic high-level description languages. " In *Proceedings of IFIP International Conf. on VLSI*, pages 187-196, August 1989.

[12] R. Marculescu, D. Marculescu, and M. Pedram. " Logic level power estimation considering spatiotemporal correlations. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 294–299, November 1994.

[13] R. Marculescu, D. Marculescu, and M. Pedram. Efficient power estimation for highly correlated input streams. In Proceedings of the Design Automation Conference, pages 628-634, June 1995.

[14] J. Monteiro, S. Devadas and A. Ghosh. Retiming sequential circuits for low power. In Proceedings of the IEEE International Conference on CAD, pages 398-402, November 1993.

[15] F. Najm Transition density: A new measure of activity in digital circuits. IEEE transactions on CAD of Integrated Circuits and Systems, 13(9):1123-1131, September 1994.

[16] M. Pedram. "Power minimization in IC design: principles and applications," Invited Paper. ACM Transactions on Design Automation of Electronic Systems, Vol. 1, No. 1 (1996), pages 1-54.

[17] C-Y. Tsui, M. Pedram, C-H. Chen, and A. M. Despain. " Low power state assignment targeting two- and multi-level logic implementations. " In *Proceedings of the* IEEE *International Conference on Computer Aided Design*, pages 82–87, November 1994.

[18] C-Y. Tsui, M. Pedram, and A. M. Despain. " Power efficient technology decomposition and mapping under an extended power consumption model. " IEEE *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(9), September 1994.

[19] C-Y. Tsui, M. Pedram, and A. M. Despain. Efficient estimation of dynamic power dissipation under a real delay model. In Proceedings of the International Conference on CAD, pages 224-228, November 1993.

[20] T. Villa and A. Sangiovanni-Vincentelli. " NOVA: State assignment of finite state machines for optimal two-level logic implementations. " IEEE T*ransactions on Computer-Aided Design of Integrated Circuits and Systems*, 9: 905-924, September 1990.