

Copyright © 1996, by the author(s).  
All rights reserved.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission.

**ENGINEERING CHANGE FOR POWER  
OPTIMIZATION USING GLOBAL SENSITIVITY  
AND SYNTHESIS FLEXIBILITY**

by

Premal Buch, Christopher K. Lennard, and A. Richard Newton

Memorandum No. UCB/ERL M96/59

11 October 1996

**ENGINEERING CHANGE FOR POWER  
OPTIMIZATION USING GLOBAL SENSITIVITY  
AND SYNTHESIS FLEXIBILITY**

by

Premal Buch, Christopher K. Lennard, and A. Richard Newton

Memorandum No. UCB/ERL M96/59

11 October 1996

**ELECTRONICS RESEARCH LABORATORY**

College of Engineering  
University of California, Berkeley  
94720

## Abstract

*Power reduction at the logic level is achieved by making functional changes in the circuit using the flexibility provided by don't care conditions. We present a technique for determining the sensitivity of circuit power dissipation to functional changes considering both local and global effects. This sensitivity is used during a power optimization strategy both to target the parts of the circuit with maximum potential for reducing network power and to guide the local functional manipulation itself. The proposed approach is based on two critical observations about the numerical properties of minterms and functions in the Boolean space: the large variance in minterm probabilities, and the small variance in expected size of a function when performing resynthesis of a node.*

*Based on these observations, the technology dependent power optimization problem is formulated as a variant of the engineering change problem. A rewiring based algorithm is proposed. Initial experimental results are very encouraging.*

# 1 Introduction

The proliferation of portable electronics and the threat of chips overheating as clock frequencies and device counts increase is bringing power minimization to the forefront of VLSI circuit design. Minimizing power dissipation of a chip does not only improve energy savings, but also chip reliability. In this work, we present a technique for reconnecting existing gates in a network to minimize power.

In its most general form, logic synthesis is the generation and optimization of a multi-level logic description which implements a specified function. Logic optimization makes use of the fact that nodes internal to a network do not generally have a uniquely specified function for satisfying correctness of an implementation. A subset of the Boolean space known as the Don't Care (DC) [15] set can be generated at each node which gives the range of functionality possible during a valid optimization step. The functionality of nodes internal to a circuit can be manipulated within the DC set to minimize area or delay. If a change is made within the Observability DC (ODC) set, then onset probability is also affected.

Engineering change is a class of synthesis algorithms dedicated to establishing a minimal change in an existing circuit in order to implement a new function. Rewiring is a subset of this set of algorithms which preserves the original gates of the network. We will show that this technique can be used to reduce power by providing incompletely specified rewiring functionality at internal network nodes which is compatible with the existing output functionality. It is the definition of suitable incompletely specified functions to guide the rewiring algorithm which is the major contribution of this work.

The power dissipation of a CMOS logic circuit depends on the gate capacitances and node switching activity. If the gate capacitance for the circuit remains fixed, then synthesis for low-power becomes an issue of reducing switching activity. The switching activity at a node depends on the functionality implemented at the node, and even a minimal Boolean difference in the functionality may significantly alter the switching activity in certain cases. This effect is particularly dramatic in circuits with distributed input switching probabilities (i.e probabilities other than 0.5) implying that it is generally invalid to optimize for power under the assumption that all input switching probabilities are 0.5.

Low-power synthesis algorithms which consider distributed input switching probabilities already exist [1][5][6][8][13][17]. These approaches either try to reduce the functional support from high-activity inputs or guide sub-expression extraction through simple high-level power approximations. However, in these works the effect on capacitance during network restructuring is difficult to predict. This is not the case for rewiring algorithms. At the technology dependent stage of synthesis, there have been works which try to minimize the total switching activity using a procedure similar to Huffman's algorithms [19] as well as methods which use area-delay trade-off curves [12][18].

The variance in minterm probability when input switching probabilities are distributed implies that circuit switching activity is particularly sensitive to functional changes made using some minterms over others. We separate the minterms in the Boolean space based on minterm probabilities into a set of classes which typically contains some high-minterm-probability, small-cardinality classes and some large-cardinality, small-probability-minterm classes. The overlap of the high-minterm-probability classes with the ODC set at a node can be used to bias area optimization towards reducing switching activity. The subset of the ODC set containing the small-probability minterm classes provides significant flexibility for area-optimization without strong influence upon activity.

However, the result of a synthesis step can be any arbitrary function within the provided functional flexibility. It is therefore difficult to define how appropriate resynthesizing a node will be for reduction in power unless the expected final onset size can be estimated. We show that the distribution of possible onset sizes of a function within a specified flexibility has a very small deviation. In fact, when changes in a function are made within the ODC constraints, it is most likely that the final function will cover almost exactly half the ODC set. This is used to predict the expected change in the onset size during rewiring, and is also a fact which guarantees that the prediction of expected TFO activity change is an accurate global measure of sensitivity.

The design of suitable incompletely specified functions for guiding rewiring is based upon the statistical properties of the minterm probability distribution [11], the formulation of a global activity sensitivity

function, first presented in [9] and extended in [10], and an examination of the distribution of possible function sizes in the Boolean space. The nature of these distributions gives rise to some fundamental observations and very intuitive concepts for power optimization.

The notion of sensitivity and flexibility in the context of logic level power optimization is described in Section 3 after a quick definition of CMOS power consumption in Section. 2. In Section 4, the engineering change solution based on these notions is outlined. Section 5 is a presentation of the results of applying this theory to standard benchmarks, and Section 6 concludes with a brief summary of this work.

## 2 Preliminaries

In this paper, the well accepted switching-activity dominated model of [1] is used to model power dissipation. Namely

$$Pow_i = \frac{1}{2} \cdot C_i \cdot \frac{V_{dd}^2}{T} \cdot E_i \quad (1)$$

where  $Pow_i$  denotes the average power dissipated by gate  $g_i$ ,  $C_i$  is the load capacitance at the output of gate  $g_i$ ,  $V_{dd}$  is the supply voltage,  $T$  is the clock period, and  $E_i$  is the average number of gate output transitions per clock cycle.

We are assuming that the primary inputs for combinational logic blocks are independent. Furthermore, it is assumed that functional switching power (the zero-delay model) is the predominant effect in determining power consumption. Under this assumption, the power consumption at a gate  $g_i$  with onset probability  $p_i$  is given by:

$$Pow_i = C_i \cdot \frac{V_{dd}^2}{T} \cdot p_i \cdot (1 - p_i) \quad (2)$$

When load capacitance is constant, functional power minimization is equivalent to optimizing  $p_i$  to be close to zero or one.



### 3 Sensitivity and Flexibility in Synthesis for Low-Power

A change in functionality at a node in a network may influence both the power dissipation local to that node as well as at nodes throughout the TFO. It was established in the work of [9] that there exists a simple and highly accurate numerical technique for computing the expected change in functionality throughout the TFO of a node when functional manipulation is performed within the bounds of the ODC. This was related to power under the assumption that all inputs had a probability of 0.5 of switching during any clock period. In general, however, this assumption which allows functional ‘size’ (i.e. minterm count) to be directly related to switching probability is invalid. The extension of this work in [10] provided a simple mechanism for generalizing the theory under the assumption that the Boolean space could be sectioned into sets of like-probability minterms. In [11] an efficient mechanism for determining such sets was outlined. This was used to establish the concept of a ‘*Power Sensitive Minterm Class*’ which was in turn used to guide functional optimization towards power optimization, neglecting the TFO effects. The work presented here will unify these ideas to establish the concept of power-sensitivity for nodes within a network with arbitrary input probabilities incorporating both local and global considerations.

We are defining functional sensitivity to be independent of the changes in activity within the region of network being rewired. The change in power within the section of reconfigured network would be estimated by structural power sensitivity. This is a much more complex phenomena than even functional sensitivity incorporating TFO effects and it is handled by making a very basic set of assumptions. This is covered further in Section 4.

To establish the concept of functional sensitivity, it is important to briefly outline the relevant work of [10] and [11]. We will first describe the technique for estimating the change in functionality throughout the TFO of a node when it is resynthesized and we will relate this to the estimated change in activity. This will define the necessity for establishing similar minterm-probability classes, and this technique will be outlined.

### 3.1 Transitive Fanout Sensitivity

Consider a node  $n$  with intermediate node inputs  $\{n_1, n_2, n_3, \dots\}$  where the functionality of node  $n_1$  is to change from  $f_{n_1}$  to  $f'_{n_1}$ . Let  $A_{n_1}$  be the set of minterms added to  $f_{n_1}$ ,  $R_{n_1}$  be the set of minterms removed. A minterm is added to the functionality at node  $n$  if it is added to (removed from) the functionality at node  $n_1$  and it is contained within the positive (negative) sensitivity of node  $n$  to  $n_1$ ,  $S_n^{pos}(n_1) (S_n^{neg}(n_1))$ . The *expected* change in function at node  $n$  is therefore given by the probability of overlap of the sensitivity and added/removed minterm sets at  $n_1$ . As the change in onset at node  $n_1$  can only occur within the ODC at that node, and the added (removed) minterms must lie within  $\overline{f_{n_1}}$  ( $f_{n_1}$ ), the following formulation may be derived:

$$E(|A_n|) = CntPr(S_n^{pos}(n_1) | (\overline{f_{n_1}} \cdot ODC_{n_1})) \cdot |A_{n_1}| + CntPr(S_n^{neg}(n_1) | (f_{n_1} \cdot ODC_{n_1})) \cdot |R_{n_1}| \quad (3)$$

Similarly, for the expected number of minterms removed:

$$E(|R_n|) = CntPr(S_n^{neg}(n_1) | (\overline{f_{n_1}} \cdot ODC_{n_1})) \cdot |A_{n_1}| + CntPr(S_n^{pos}(n_1) | (f_{n_1} \cdot ODC_{n_1})) \cdot |R_{n_1}| \quad (4)$$

In this context,  $CntPr$  is defined as:  $CntPr(A|B) = \frac{|A \cap B|}{|B|}$

This formulation can be propagated throughout the TFO to estimate the expected size of the change in functionality at every node influenced by the change at  $n_1$ . (The technique for handling reconvergent fanout is outlined in [10], and omitted here.) Although this is only a prediction of *average* change in functionality without an estimate of standard deviation, extensive practical experimentation has shown that the variance of actual size of functional change versus average estimate is extremely small. This follows from the fact that the vast majority of possible functions which may arise during a synthesis step cover near half the total number of minterms available within the ODC flexibility. This is discussed in greater detail in Section 3.3.

When all minterms (input variable assignments) have the same probability of occurrence, the relationship between the change in switching activity and expected size of the change in functionality is trivial. However, this is not the case when minterm probabilities are distributed. Although the prediction of change

in power assuming that all minterms are equally likely would correctly average out when sensitivity performance is examined over a large number of circuits, in general the standard deviation would become much too large to guarantee the usefulness of the method for any specific instance. To make the technique more viable, it has to be able to be tuned to the specific input probability distribution. This is achieved by splitting the Boolean space into a set of classes,  $\phi$ , of like probability minterms. The technique outlined above is then performed inside each class, resulting in the following sums:

$$E(Pr(A_n)) = \sum_{C_i \in \phi} E(|A_{n_i}|) \cdot \frac{Pr(C_i)}{|C_i|} \quad (5)$$

$$E(Pr(R_n)) = \sum_{C_i \in \phi} E(|R_{n_i}|) \cdot \frac{Pr(C_i)}{|C_i|} \quad (6)$$

As the computation of expected change in power given the local expected change is a completely numerical procedure once the  $CntPr(S_n^{neg}(n_1) | (\overline{f_{n_1}} \cdot ODC_{n_1} \cdot C_i))$  etc. terms have been computed for each node and class in a single pass over the network, a reasonable number of classes can be handled without the computational penalty dominating the synthesis routine.

### 3.2 Defining Classes and Power Sensitive Minterms

Establishing a set of classes of maximally similar probabilities is in general an exponentially difficult problem. To generate a set of classes efficiently, the structure of the probability space needs to be utilized. Consider an  $n$ -input circuit with all inputs having onset probability  $p$ . Clearly, there are  ${}^nC_j$  minterms in the Boolean space defined by the input variables of probability:  $p^j \cdot (1-p)^{n-j}$ , implying that there are  $(n+1)$  classes needed to partition the space for zero-variance minterm-probability per class. However, it is also trivial to compute the maximum possible probability error as the total number of classes is reduced by collapsing these zero-variance classes [11]. In general, the Boolean space is the product of a set of Boolean spaces, each space being described by a set of like-probability variables. A set of classes for a general space is formed by grouping like input probabilities, constructing class-count/error curves for each Boolean subspace which those groups define, and choosing a class count for each subspace based on total toler-

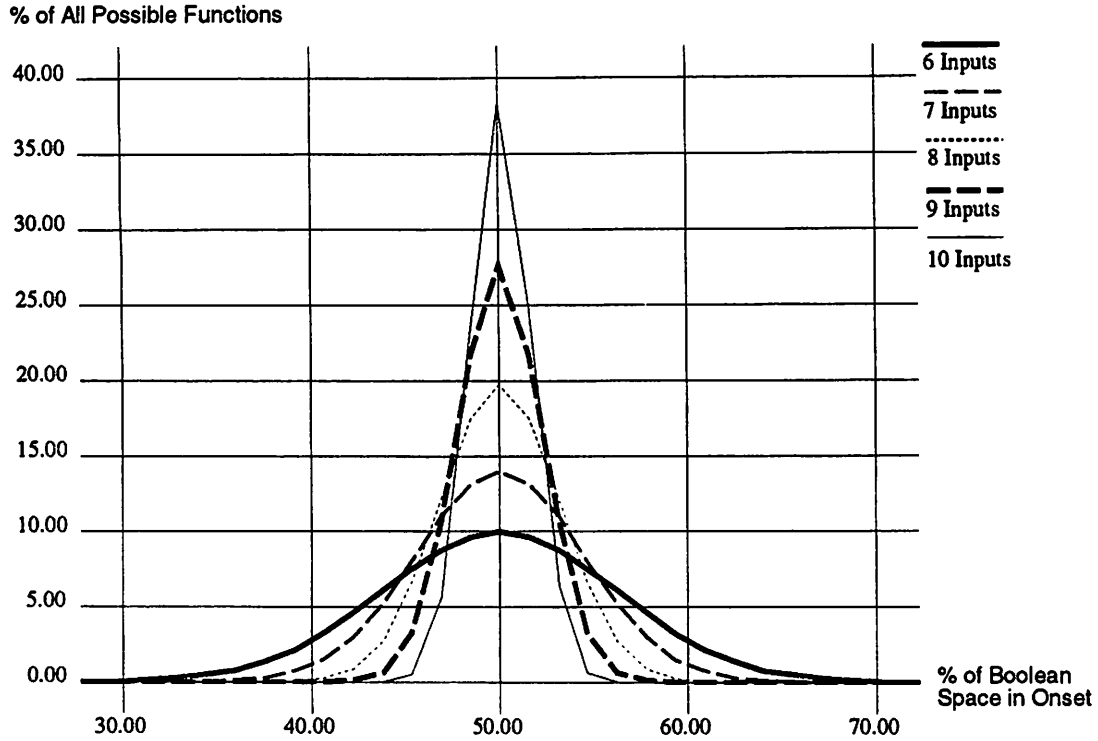


Figure 2. Functional size probability profile

able error for the overall Boolean space. A very simple product relationship exists which bounds the total error given the error in each subspace, and the final set of classes is the product of the classes chosen for each subspace. This very efficient technique is outlined in detail in [11].

### 3.3 Flexibility

We have generalized the TFO sensitivity algorithm for circuits with arbitrary input switching probabilities through the assumption of being able to partition the Boolean space into several classes containing similar-probability minterms. Further, we have now shown how those classes can be efficiently computed. All that remains in the computation of a global power sensitivity is a prediction of the size of the expected functional change during resynthesis. To establish this, we assume that any function within the bounds of the provided flexibility is equally likely. This allows a functional size probability profile to be trivially established as there exists  ${}^N C_k$  ways of forming a  $k$ -minterm size function in an  $N$ -minterm Boolean

space. Example profiles are shown in Figure 2 for several input variable counts. (All profiles are binned into 64 x-axis data points for comparative purposes.)

As the number of variables in the functional support increases, the ‘*centralizing*’ effect becomes more dramatic. For input counts {4, 6, 8, 10} the normalized standard deviation of the function count profile is {0.125, 0.062, 0.031, 0.016} respectively. It is this property which allows an average prediction technique for establishing sensitivity to work incredibly well for estimating the global effect of specific synthesis cases, as experimentally verified in [8].

The functional flexibility is defined with respect to the most likely functional change. For example, if the DC contains  $N$  minterm,  $m$  of which are originally in the function on-set, the flexibility (or ‘expected functional change’) is given by:  $\left| \frac{N}{2} - m \right|$ . The application of this to synthesis is detailed in Section 4.2.1.

## 4 Power Optimization Through Engineering Change

### 4.1 Engineering Change Based Formulation

Given a logic network that has been already synthesized (possibly for low power), we want to reduce the power dissipation by making incremental changes. In [11] a technique was explored whereby regional synthesis was guided to make small changes in node functionality throughout a multi-level network such that the total power was reduced. However, the additional circuitry that might be required to implement those functional changes could possibly offset the power reduction achieved. This adverse effect can be reduced by using an engineering change approach which aims at modifying the circuit in a ‘minimal’ way to realize the new specification at the node.

The problem of minimal modification of the circuit to reduce power differs a little from the engineering change problem in that the target function is not a hard constraint. In general, we just want to achieve an arbitrary expansion/contraction of the onset within the ODC set such that the power dissipation is decreased. This behavior can be captured by an incompletely specified target function, i.e. a target function such that any onset change which meets the target function specification is beneficial. Computing a function which includes all possible such changes is exponentially complex. However, the techniques described in Section 3 may be used to compute a target function for which any arbitrary change which meets the function specification is *expected* to be beneficial.

In the following, we outline an algorithm to compute this target function and a rewiring based approach to solve the engineering change problem. The choice of a rewiring approach is particularly appropriate in the context of power optimization as rewiring a region of the circuit does not affect existing gate capacitance.

### 4.2 Rewiring based Power Optimization

The proposed rewiring based algorithm consists of two phases: identifying the redesign region and applying rewiring to reduce power dissipation. The first phase involves determining the circuit nodes

which have the highest flexibility and power sensitivity. All nodes are ranked based on the sensitivity of network power to expected change in functionality and the nodes contributing the largest beneficial changes are selected for optimization. In a mapped circuit, optimizing just one node does not yield significant power gains. In order to provide a sufficiently large input for the optimizer to manipulate, we identify a region for rewiring with the flexible node at its root.

#### 4.2.1 Choose Rewiring Region based on Flexibility/Sensitivity Considerations

For each node, we estimate the *appropriateness* of the node from a power optimization perspective by computing the expected change in its power dissipation under an arbitrary optimization step, as follows:

We first partition the boolean space in to  $k$  classes using the techniques presented in Section 3.2, such that all minterm probabilities in the  $i$ th class can be approximated by one average probability value, say  $p_i$ . Let the node cover before optimization be  $f$  and after optimization be  $f'$ . Let the essential minterms of  $f$  be represented by the function  $f_{essential}$  and the don't cares by  $f_{DC}$ . The expected probability of  $f'$ ,  $Pr(f')$  is then  $Pr(f') = \sum_{i=0}^k Pr(f'_i)$ , where  $f'_i$  is the projection of  $f'$  over the  $i$ th class. Since minterms in each class are represented by a single probability,  $Pr(f')$  is given by  $Pr(f') = \sum_{i=0}^k p_i \cdot |f'_i|$ .

Based on the current onset probability, we then decide if it is beneficial to expand or contract the onset. If the current probability of the node cover  $f$ ,  $Pr(f) > 0.5$ , it is desirable to expand the onset so that the node power dissipation decreases and similarly it is desirable to contract if  $Pr(f) < 0.5$ . In each class  $i$ , the final cover  $f'_i$  must include  $f_{essential_i}$ , and may include some subset of  $f_{DC_i}$ . Thus, for each class we have two possibilities: keep the original  $f_i$  within the class, or optimize  $f_i$  using  $f_{DC_i}$ . We compute the expected value of the onset size of  $f'_i$ ,  $E(|f'_i|)$ , under the conditions of allowing or not allowing DC flexibility to influence functional specification within the class. The configuration most compatible with the objective of decreasing local and TFO activity is then chosen. For example, in the latter case where ODC flexibility is permitted,  $E(|f'_i|)$  can be computed as  $E(|f'_i|) = |f_{essential_i}| + E(|f'_i \cdot f_{DC_i}|)$ .  $E(|f'_i \cdot f_{DC_i}|)$  is the expected value of the onset size of a function selected from a set of minterms with a cardinality of  $|f_{DC_i}|$  via some optimization step. From the discussion of Section 3.3, we see that

$E(|f_i \cdot f_{DC_i}|) = \frac{|f_{DC_i}|}{2}$  giving  $E(|f_i|) = |f_{essential_i}| + \frac{|f_{DC_i}|}{2}$ . This expected change in functionality is combined with the TFO sensitivity work of Section 3.1 to predict a global power sensitivity.

This expected global power dissipation change estimate is computed for all possible combinations of permitted/not-permitted flexibility in each class and the best flexibility configuration is then chosen for each node. The nodes with the highest potential for reducing power are then chosen for optimization. Note that the above procedure identifies the classes of DC most useful in power optimization. The classes in which  $f_{DC_i}$  is provided as functional flexibility to the synthesis tool are referred to as *useful DC classes*.

#### 4.2.2 The Rewiring for Low Power Algorithm

We propose an algorithm based on rewiring to redesign the hot, flexible network region identified by the techniques of the previous section. The redesign algorithm presented here modifies existing circuitry by reconnecting gates in the region with all the gate types and gate counts unchanged. As a result, the power optimization process does not change the total gate capacitance of the circuit, which means that any reduction in switching activity is made without a capacitance trade-off.

The proposed rewiring algorithm is an adaptation of [7] which formulates the redesign problem as a Boolean-constraint problem and gives an algorithm based on BDDs to generate all possible assignments of gate connections which satisfy the specified target functionality.

The rewiring algorithm assigns a Boolean connection variable for each ordered pair of gate outputs and inputs in the region. The value of the variable is 1 if there exists a connection between this pair in the redesigned circuit and 0 otherwise. It then builds a characteristic function for each gate to capture all possible functionalities that can be implemented at that gate using all possible combinations of the connection variables.

Formally, let  $c_{v_i v_k}$  ( $i = 1, \dots, k-1$ ) be the Boolean variable for the connection from the output of gate  $i$  to an input of a 2-input AND gate  $k$ . Let  $v_i$  be the Boolean variable corresponding to the output of the gate  $i$ . Then this characteristic function  $\chi_{2\text{-AND}}$  of the AND gate  $k$  is



$$\chi_{2\text{-AND}} = \text{LTE2}(c_{v_1 v_k}, \dots, c_{v_{k-1} v_k}) \cdot \left( v_k \equiv \prod_{i=1}^k (c_{v_i v_k} \Rightarrow v_i) \right) \quad (7)$$

where  $\text{LTE2}()$  is the Boolean function which evaluates to one iff  $\leq 2$  of its arguments are one. The  $\text{LTE2}$  thus selects two of the all possible connection variables feeding into the inputs of gate  $k$ , and  $\chi_{2\text{-AND}}$  captures all possible connection assignments which implement the AND function at  $v_k$ .

The complete set of possible functions which can be implemented by the region can then be computed given by ANDing the characteristic functions of all gates. That is, the characteristic function of the circuit after reconnection  $\chi$ , is given by

$$\chi(v, c) = \prod_{g \in G} \chi_g(v, c) \quad (8)$$

where  $G$  is the set of gates in the original network and  $\chi_g$  is the characteristic function for gate  $g$ .  $v$  is the set of all circuit variables and  $c$  is the set of all connection variables.

After smoothing out all the internal variables in  $v$  which are associated with the intermediate nodes in the region, we are left with  $\chi(i, o, c)$ , a function of the primary input ( $i$ ) and output variables( $o$ ) and all the connection variables. We then compare this with the characteristic function  $\chi_s(i, o)$ , of the specifications. Note that in our case, this is an incompletely specified function with the useful DC computed from the previous section providing the don't cares conditions. The condition on the connection variables  $c$  then is that the input-output behavior of the reconnected circuit implies the input-output behavior of the specification. i.e.,

$$\chi_{\text{redesign}}(c) = \zeta_{i, o}(\chi(i, o, c) \Rightarrow \chi_s(i, o)) \quad (9)$$

The consensus operator above extracts all 0-1 assignments to  $c$  such that  $\chi(i, o, c) \Rightarrow \chi_s(i, o)$  is a tautology. Note that each minterm of the characteristic function represents a 0-1 assignment of the connection variables which will satisfy the target functionality. For more details of the algorithm, refer to [7].

The target function for the redesign region is computed using  $f_{\text{essential}}$  and the useful DC classes as determined by the algorithm in the previous section. The union of these useful DC classes gives a subset of

the node DC set which is expected to be beneficial for power optimization and this incompletely specified function is used to direct the rewiring algorithm. The output of the rewiring algorithm is a set of minterms of connection variables. While each of these represents a different wiring scheme with a different power dissipation, based on the reasoning of Section 3.3, the power dissipation is expected to reduce when we randomly pick a single wiring assignment. Without any loss of generality, we pick the assignment which implies the minimum numbers of connection wires. This minimizes the power dissipation due to wiring capacitances, and under a unit delay model guarantees that the critical path length for the region does not increase.

## 5 Experimental Results

The algorithms outlined in this paper have been implemented inside the SIS logic synthesis package. A subset of circuits from the MCNC and ISCAS\_89 benchmark set were used to obtain the experimental results. All circuits were mapped using *msu.genlib*. Without any loss of functional generality, the rewiring algorithm uses a reduced form of this library due to the limitations of the current rewiring implementation. Power estimation and switching activity computation was performed using the symbolic simulation method of [1] using a zero-delay model. All input probabilities were chosen from a uniform distribution over  $[0,1]$ .

The results from the rewiring algorithm are presented in Table 1. The results obtained by first mapping the circuit, then optimizing it for area using *script.rugged* and then applying our rewiring algorithm to it. The normalized power reduction data in Column 4 presents the ratio of the power dissipation of the circuit after our algorithm to the power dissipation of the area optimized circuit input to our algorithm. Overall, a 4% reduction in power was achieved. Note that rewiring can never increase the gate count so in essence there is no trade-off in this power reduction. In fact, there is, in general a reduction in literal count due to the fact that during the rewiring procedure, not all gates are re-used (the number of literals is used as a metric of area since the SIS framework does not provide a gate count without re-mapping the circuit. The validity of the rewiring program was verified by re-mapping the circuit and in all cases the gate count either remained the same or decreased by 1-2 gates). We expect the results to further improve as we extend our benchmarking to large circuits, since these circuits would have more flexibility for redesign.

<b>Circuit</b>	<b>#lit before EC</b>	<b>#lit after EC</b>	<b>Normalized Power Reduction (after/before)</b>
traffic_cl	17	16	0.98
b1	17	15	0.97
mux_cl	45	42	0.98
cm82	42	36	0.95
cm151	45	41	0.94
parity	120	90	0.88
cm42	38	34	0.96
cm138	35	31	0.96
c17	12	12	1.00
tcon	49	48	0.99
decod	56	52	0.97
cmb	84	76	0.96
cm163	76	67	0.95
pcl	117	106	0.95
mux	87	82	0.98
cm162	69	61	0.98
cm150	87	80	0.97
cm85	75	65	0.93
z4ml	66	60	0.96
cu	85	77	0.96
pcler8	150	137	0.95
cc	92	79	0.94
unreg	170	152	0.96
count	210	208	0.99
my_adder	320	288	0.96
comp	213	177	0.94
cht	275	238	0.87
c8	234	210	0.96
lal	162	143	0.95
b9	180	162	0.96
cordic	102	91	0.95
frg1	215	199	0.97
ttt2	303	280	0.96
term1	272	248	0.96
Total			0.96

## 6 Conclusions

We presented a technique for determining the sensitivity of circuit power dissipation to functional changes considering both local and global effects. This sensitivity is used during a power optimization strategy both to target the parts of the circuit with maximum potential for reducing network power and to guide the local functional manipulation itself. The proposed approach is based on two critical observations about the numerical properties of minterms and functions in the Boolean space: the large variance in minterm probabilities, and the small variance in expected size of a function when performing resynthesis of a node.

Based on these observations, the technology dependent power optimization problem was formulated as a variant of the engineering change problem. We have proposed and implemented an engineering change solution for low-power re-synthesis which *reduces power dissipation via activity reduction with minimal impact on circuit structure, and no trade-off with the overall area*. This algorithm uses the concept of power-sensitivity to identify nodes which are most likely to provide power reductions and determines the best restriction on functional flexibility to guide the rewiring algorithm towards lower power.

We combined the theory of [10] and [11] into a unified framework to allow global power sensitivities to be defined for networks with arbitrary input probability distributions. To achieve this, we exploited the concept of similar-minterm-probability classes within which the existing functional sensitivity theory applies. Furthermore, this was combined with new theory for estimating the expected change in functionality during synthesis given a specific flexibility.

We are currently working on extending the implementation to a handle a larger variety of library gates as well as benchmarking larger circuits. Due to the increased amount of flexibility in larger circuits, we expect the results to improve even further in these cases.

## References

- [1] R. I. Bahar, and F. Somenzi, "Boolean Techniques for Low Power Driven Resynthesis," *Proceedings of the International Conference on Computer-Aided Design*, pp. 428-432, Nov. 1995.
- [2] R. K. Brayton, G. D. Hachtel, C. T. McMullen, and A. Sangiovanni-Vincentelli, "Logic Minimization Algorithms for VLSI Synthesis," Kluwer Academic Publishers, 1984.
- [3] A. P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R. W. Broderson, "Optimizing Power Using Transformations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, pp. 12-31, Jan. 1995.
- [4] A. Ghosh, S. Devadas, K. Keutzer, and J. White, "Estimation of Average Switching Activity in Combinational and Sequential Circuits," *Proceedings of the 29th Design Automation Conference*, pp. 253-259, June 1992.
- [5] S. Iman, M. Pedram, "Multi-Level Network Optimization for Low Power," *Proceedings of the International Conference on Computer-Aided Design*, pp. 372-377, Nov. 1994.
- [6] S. Iman, M. Pedram, "Logic-Extraction and Factorization for Low Power," *Proceedings of the 32nd Design Automation Conference*, pp. 248-253, June 1995.
- [7] Y. Kukimoto, M. Fujita, and R. K. Brayton, "A Redesign Technique for Combinational Circuit Based on Gate Reconnections," *Proceedings of the International Conference on Computer-Aided Design*, pp. 632-637, Nov. 1994.
- [8] L. Lavagno, P. C. McGeer, A. Saldanha, A. Sangiovanni-Vincentelli, "Timed Shannon Circuits: A Power Efficient Design Style and Synthesis Tool," *Proceedings of the 28th Design Automation Conference*, pp. 254-260, June 1995.
- [9] C. K. Lennard, A. R. Newton, "An Estimation Technique to Guide Low Power Resynthesis Algorithms," *Proceedings of the 1995 International Symposium on Low Power Design*, pp. 227-232, April 1995.
- [10] C. K. Lennard, A. R. Newton, "On Estimation Accuracy for Guiding Low-Power Resynthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 15, No. 6, pp. 644-664, June 1996.
- [11] C. K. Lennard, P. Buch, and A. R. Newton, "Logic Synthesis using Power Sensitive Don't Care Sets," *Proceedings of the 1996 International Symposium on Low Power Electronics & Design*, August 1996.
- [12] B. Lin, and H. De Mann, "Low-Power Driven Technology Mapping Under Timing Constraints," *Proceedings of*

*the International Workshop on Logic Synthesis*, May 1993

- [13] S. C. Prasad, and K. Roy, "Circuit Activity Driven Multi-level Logic Optimization for Low Power Reliable Operation," *Proceedings of the European Conference on Design Automation*, pp. 368-372, Feb. 1993.
- [14] R. Rudell, and A. Sangiovanni-Vincentelli, "Multiple-valued Minimization for PLA Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 6, pp. 727-750, Sept. 1987.
- [15] H. Savoj, R. K. Brayton, "The Use of Observability and External Don't Cares for the Simplification of Multi-Level Networks," *Proceedings of the 28th Design Automation Conference*, pp. 297-301, June 1990.
- [16] E. M. Sentovich, K. J. Singh, C. Moon, H. Savoj, R. K. Brayton, and A. Sangiovanni-Vincentelli, "Sequential Circuit Design Using Synthesis and Optimization," *Proceedings of the International Conference on Computer-Aided Design*, pp. 328-333, Oct. 1992.
- [17] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer, "On Average Power Dissipation and Random Pattern Testability of CMOS Combinational Logic Networks," *Proceedings of the International Conference on Computer-Aided Design*, pp. 402-407, Nov. 1992.
- [18] V. Tiwari, P. Ashar, and S. Malik, "Technology Mapping for Low Power," *Proceedings of the 28th Design Automation Conference*, pp. 74-79, June 1993.
- [19] C.-Y. Tsui, M. Pedram, and A. Despain, "Technology Decomposition and Mapping Targeting Low Power Dissipation," *Proceedings of the 28th Design Automation Conference*, pp. 68-73, June 1993.