Engineering Change for Power Optimization Using Global Sensitivity and Synthesis Flexibility

Premal Buch   Christopher K. Lennard   A. Richard Newton

Department of Electrical Engineering & Computer Sciences
University of California, Berkeley, CA 94720
{premal, chrisl, newton}@EECS.Berkeley.EDU

Abstract

A technology dependent power optimization technique is proposed which formulates the problem of hot spot reduction as a variant of the engineering change (EC) problem. A technique is presented for determining the sensitivity of circuit power dissipation to functional changes considering both local and global effects. This sensitivity is combined with a measure of synthesis flexibility to identify hot regions in the circuit which have a lot of flexibility in making functional changes and for whom a small functional change can greatly affect the overall power dissipation. An incompletely specified target function is constructed for the hot region such that any implementation satisfying it is expected to reduce power. A rewiring algorithm is used to solve the resulting EC problem without affecting circuit area, gate capacitance or delay under the unit delay model. Experimental results on a set of MCNC benchmark circuits show that the proposed approach can give up to 13% reduction in power dissipation with an average reduction of 4%.

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 not only improves energy savings, but also chip reliability. In this work, we present a technique for reconnecting existing gates in a network to minimize power and reduce hot spots in the circuit.

Logic synthesis uses 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) [8] set can be generated at each node which gives the range of functionality possible at the node. The functionality of nodes can then be manipulated within the DC set to minimize area, delay or power.

Engineering change (EC) is a class of synthesis algorithms which aim at implementing a new function in a circuit by making minimal changes to an existing circuit Rewiring is a subset of this set of algorithms which preserves the original gates of the network. Fig. 1 shows an example where rewiring a region inside the circuit reduces the switching activity. Note that rewiring an internal region affects the switching activity of the transitive fanout (TFO) of the region as well. In this example, rewiring takes advantage of the external DC to yield a function with lower switching activity.

The power dissipation of a CMOS logic circuit depends on the gate capacitances and node switching activity. There have been several works on power optimization during technology mapping [7][10][11]. Rewiring, which is employed after technology mapping, allows us to make small functional changes in the circuit which do not change the circuit structure and thus the capacitance. This gives us more control over power dissipation and yields results which can be expected to translate in power gains even at the transistor and layout level. [8] and [1] optimize a mapped circuit for power by evaluating candidate circuits generated by a set of structural transformations using ATPG methods and learning based redundancy addition/removal respectively. Both of these methods can change area and delay properties of the circuit.

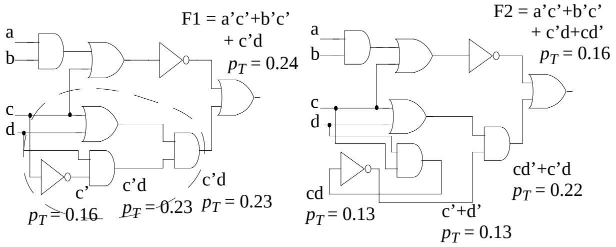

Figure 1: Reducing switching activity via rewiring: F = a'c'+b'c'+c'd', DC = cd', p(a) = 0.4, p(b) = 0.6, p(c) = 0.2, p(d) = 0.8. transition probability $p_T(x) \equiv p(x=1)(1-p(x=1))$.

We will show that rewiring can be used to reduce power of hot regions by providing it an incompletely specified target function for the region such that any rewiring compatible with the target function is expected to dissipate less power. This rewiring does not change the circuit area or gate capacitance and does not increase the circuit delay under the unit delay model. The selection of hot regions, EC formulation and the construction of this incompletely specified function for EC are the major contributions of this work.

An overview of the proposed algorithm is presented in Section 3, after some definitions in Section 2. The notion of sensitivity and flexibility in logic level power optimization is described in Section 4. Section 5 outlines the proposed EC solution. Section 6 presents the experimental results.

2 Preliminaries

Dynamic power dissipation in static CMOS circuits is given by

$$P_i = 0.5 \cdot C_i \cdot V_{dd}^2 \cdot E_i \cdot f = C_i \cdot V_{dd}^2 \cdot p_i \cdot (1 - p_i) \cdot f \quad \text{(EQ 1)}$$

where P_i denotes the average power dissipated by gate g_i, C_i is the load capacitance at the gate output, V_{dd} is the supply voltage, f is the clock frequency, and E_i is the average number of gate output transitions per clock cycle. Under the zero-delay model (appropriate when functional power is the dominant effect) and assuming that the primary inputs are independent, the power consumption at a gate g_i with onset probability p_i can be further simplified as above. Functional power minimization under constant load capacitance is then equivalent to optimizing p_i to be close to zero or one.

3 Algorithm Overview

In this work, we are interested in reducing the power dissipation of a circuit by making very small functional changes. An EC based formulation is used to achieve this (in Section 5 we discuss the rationale behind this in detail) after constructing the target function for EC using global sensitivity and synthesis flexibility measures.

This work was supported in part by SRC contract DC-324-96

The proposed algorithm to achieve this is as follows:

1. The Boolean space is partitioned into minterm classes such that each class has minterms of similar probabilities, which can each be approximated by a single value per class. This allows us to relate probability of a function in each class to its onset size in each class (Section 4.1).
2. A global sensitivity based formulation is used to relate the expected change in overall power dissipation to a change in the onset size of an internal node (Section 4.2).
3. We make some observations about the distribution of the number of possible logic functions in a Boolean space as a function of their onset sizes. This is used as a measure of the flexibility available at a node in making functional changes and to predict the expected onset size when an arbitrary functional change is made at an internal node. (Section 4.3).
4. (2) and (3) above are used to select nodes with a lot flexibility in making functional changes and for whom a small functional change can greatly affect the overall power dissipation.
5. For each selected node, (1) and (3) are used to select the classes of the DC minterms where the predicted onset size when making a functional change results in a beneficial probability. This is used to construct the incompletely specified target function for EC (Section 5.1).
6. Rewiring is used to select the minimum wire implementation satisfying the target function. This guarantees that the circuit area and gate capacitance do not change and the critical path does not increase under the unit delay model (Section 5.2).

4 Global Sensitivity and Synthesis Flexibility

A change in fuctionality 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 our work of [4] 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 DC. 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 [5] 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 [6] we outlined an efficient mechanism for determining such sets. The work presented here will unify these ideas to establish the notion of power-sensitivity for nodes within a network with arbitrary input probabilities with both local and global considerations. The concepts presented here can also be applied to improve technology independent synthesis of [6].

4.1 Constructing Minterm Classes

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*. There are ${}^{n}C_{j}$ 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 approximation error minterm classes. However, it is possible to generate a set of classes such that total error is bounded. A very efficient technique for this is outlined in detail in [6].

4.2 Transitive Fanout Sensitivity

Consider a node *n* with intermediate node inputs $\{n_1, n_2, n_3, \ldots\}$ where the functionality of node n_1 is to change from f_{n_1} to f'_{n_1}. Let A_n be the set of minterms added to f_{n_1}, R_n 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 DC at that node, and the added (removed) minterms must lie within $\overline{f_{n_1}}$ (f_{n_1}), the following formulation may be derived:

$$E(|A_n|) = p(S_n^{pos}(n_1)|(\overline{f}_{n_1} \cdot ODC_{n_1})) \cdot |A_{n_1}| + p(S_n^{neg}(n_1)|(f_{n_1} \cdot ODC_{n_1})) \cdot |R_{n_1}| \qquad \text{(EQ 2)}$$

Similarly, for the expected number of minterms removed:

$$E(|R_n|) = p(S_n^{neg}(n_1)|(\overline{f}_{n_1} \cdot ODC_{n_1})) \cdot |A_{n_1}| + p(S_n^{pos}(n_1)|(f_{n_1} \cdot ODC_{n_1})) \cdot |R_{n_1}| \qquad \text{(EQ 3)}$$

where $p(A|B) = |A \cap B| / |B|$

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 [5], 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 DC flexibility (Section 4.3).

When all minterms have the same probability of occurrence, the relationship between the change in switching activity and expected size of the change in functionality is straightforward. 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. This is resolved by splitting the Boolean space into a set of classes, φ, of like probability minterms. The technique outlined above is then performed inside each class, resulting in:

$$E(p(A_n)) = \sum_{C_i \in \varphi} E(|A_{n_i}|) \cdot p(C_i) / |C_i| \qquad \text{(EQ 4)}$$

$$E(p(R_n)) = \sum_{C_i \in \varphi} E(|R_{n_i}|) \cdot p(C_i) / |C_i| \qquad \text{(EQ 5)}$$

As the computation of expected change in power given the local expected change is a completely numerical procedure once the $p(S_n^{neg}(n_1)|(\overline{f}_{n_1} \cdot ODC_{n_1} \cdot C_i))$ 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.

4.3 Flexibility

We have generalized the TFO sensitivity algorithm for circuits with arbitrary input switching probabilities via the assumption of being able to partition the Boolean space into several classes containing similar-probability minterms. Further, we have now shown how these 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 straightforward construction of a functional size probability profile 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 Fig. 2 for several input variable counts. (All profiles are binned into 64 x-axis data points for comparative purposes.)

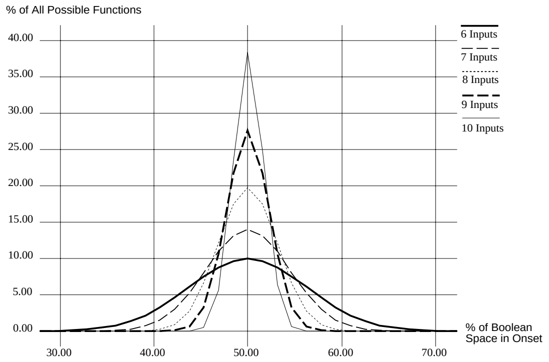

Figure 2. Functional size probability profile

As the number of variables in the functional support increases, the *centralizing* effect becomes more dramatic. Even 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. This would decrease exponentially for real-life functions with more inputs. It is this property which allows an average prediction technique for establishing sensitivity to work very well for estimating the global effect of specific synthesis cases.

This property also lets use define the functional flexibility for an arbitrary synthesis step. The *flexibility* is defined as the *expected functional change*. Due to the above centralization property, any function resulting from an arbitrary functional change is *expected* to have half of the DC flexibility in its onset. For example, for a DC containing N minterm, m of which are originally in the function on-set, it is given by: $0.5 \cdot |N - m|$. The application of this to synthesis is detailed in Section 5.1.

5 Engineering Change via Rewiring

Given a logic network that has been already been synthesized (possibly for low power), we want to reduce the power dissipation by making incremental changes. In [6] 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 EC approach which aims at minimally modifying the circuit to realize the new specification at the node.

The problem of minimal modification of the circuit to reduce power differs a little from the EC 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 4 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 EC 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 change gate capacitance. 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.

5.1 Choosing The Rewiring Region

For each node, we estimate the *suitability* of the node for power optimization 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 4.1, such that all minterm probabilities in the ith class can be approximated by one average probability value, say p_i. For a node implementing a functionality f, 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 DC by f_{DC}. The probability of f'' , $p(f'')$ is then $p(f'') = \sum_{i=0}^{k} p(f_i'')$, where f_i'' is the projection of f'' over the ith class. Since minterms in each class are represented by a single probability, $p(f'')$ is given by $p(f'') = \sum_{i=0}^{k} p_i \cdot |f_i''|$.

Based on the current onset probability, we then decide if it is beneficial to expand or contract the onset. Since from (EQ 1) $P(f) \propto p(f) \cdot (1 - p(f))$, if the current probability of the node cover f' , $p(f') > 0.5$, it is desirable to expand the onset so that the node power dissipation decreases, and to contract if $p(f') < 0.5$.

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 the 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 case where DC flexibility is permitted, $E(| f_i'' |)$ can be computed as $E(|f_i''|) = |f_{essential_i}| + E(|f_{DC_i}''|)$. $E(|f_{DC_i}''|)$ 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 in Section 4.3, $E(|f_{DC_i}''|) = 0.5 \cdot |f_{DC_i}|$ giving $E(|f_i''|) = |f_{essential_i}| + 0.5 \cdot |f_{DC_i}|$. This expected change in functionality is combined with the TFO sensitivity work of Section 4.2 to predict a global power change.

This expected global power dissipation change estimate is computed for all possible combinations of allowing/not allowing the use of DC in each class and the best combination of DC classes is then chosen for each node. The minterm classes i for which f_{DC_i} is used in the best flexibility combination are referred to as *useful DC classes*. The nodes with the highest potential for reducing power are then chosen for optimization and the useful DC classes for each are used to define the incompletely specified target function to be implemented at the node.

5.2 The Rewiring for Low Power Algorithm

We propose the use of 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 to the first order approximation (switching capacitance can increase if a higher switching activity net is connected to a high capacitance pin).

The proposed rewiring algorithm is an adaptation of [3] which formulates the redesign problem as a Boolean-constraint problem and gives a BDD-based algorithm to generate all possible assignments of gate connections which satisfy the given target function.

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. Each minterm of the characteristic function represents a 0-1 assignment of the connection variables which will satisfy the target functionality. The reader is referred to [3] for more details of the algorithm. We use a breadth-first ordering to order gates when introducing connection variables. This ensures that no assignment of connection variables increases the number of levels in the rewired circuit, thus guaranteeing that the critical path length, and consequently the circuit delay under a unit delay model does not increase.

The target function for the redesign region is computed using $f_{essential}$ and the useful DC classes as determined by the algorithm in the previous section. The output of the rewiring algorithm is a set of minterms of connection variables, each of which satisfy the incompletely specified target function. While each of these represents a different wiring scheme with a different power dissipation, based on the reasoning of Section 4.3, the target function is already constructed such that the power dissipation is expected to reduce when we arbitrarily pick a single wiring assignment. Without any loss of generality, we pick the assignment which implies the minimum numbers of connection wires (i.e., the minterm of the characteristic function with the minimum number of 1's). This minimizes the power dissipation due to wiring capacitances.

6 Experimental Results

The algorithms outlined in this paper have been implemented inside the SIS package. A set of circuits from the MCNC and ISCAS_89 benchmark were used to generate the experimental results. Power estimation was performed using the symbolic simulation method of [2] using a zero-delay model. All input probabilities were chosen from a uniform distribution over [0,1]

The results presented in Table 1 were obtained by first optimizing the circuit for area using script.rugged and then mapping it before applying our algorithm to it. The mapping was performed using a subset of msu.genlib. This is purely an artifact of the current implementation and does not represent an algorithmic limitation. The runtimes in Column 2 are on a DEC Alpha machine. Column 3 contains the power dissipations of the mapped, area optimized circuit input to our algorithm and Column 4 has the power dissipation of the rewired circuit resulting from our algorithm. Column 5 contains the ratio of these two. Overall, a 4% reduction in power was achieved, with reductions of up to 13% in some cases. Note that rewiring can never increase the gate count so in essence there is no trade-off in this power reduction. In fact, there was in general a reduction in literal count due to the fact that during the rewiring procedure, not all gates pins are necessarily reused. We expect the results to further improve as we extend our benchmarking to large circuits, since these circuits would have more flexibility for redesign.

Circuit	run time	power (μW)		power
	(sec)	before EC	after EC	ratio
traffic_cl	0.1	26.1	25.7	0.98
b1	0.1	16.6	16.1	0.97
mux_cl	0.6	96.0	93.6	0.98
cm82	0.6	83.6	79.2	0.95
cm151	3.2	94.7	88.6	0.94
parity	4.2	225.5	197.4	0.88
cm42	0.3	59.4	57.3	0.96
cm138	0.3	49.4	47.3	0.96
c17	0.1	25.8	25.8	1.00
tcon	0.7	112.2	110.9	0.99
decod	0.7	67.2	65.1	0.97
cmb	1.2	174.3	167.9	0.96
cm163	1.3	157.2	149.6	0.95
pcle	2.8	168.5	160.1	0.95
mux	2.0	94.7	190.4	0.98
cm162	0.7	104.0	101.7	0.98
cm150	1.9	176.4	171.3	0.97
cm85	1.1	95.3	88.2	0.93
z4ml	1.2	105.4	100.9	0.96
cu	1.2	131.6	126.1	0.96
pcler8	0.7	229.5	218.8	0.95
cc	2.3	161.6	151.4	0.94
unreg	36.1	328.4	314.3	0.96
count	10.9	414.3	412.1	0.99
my_adder	38.2	649.0	622.5	0.96
comp	87.0	437.8	398.7	0.94
cht	33.6	218.1	190.3	0.87
c8	10.5	488.3	467.4	0.96
lal	4.9	310.4	295.7	0.95
b9	66.5	364.9	350.2	0.96
cordic	2.4	195.1	185.6	0.95
frg1	12.0	457.5	444.1	0.97
ttt2	14.2	362.3	348.9	0.96
term1	36.7	548.9	527.8	0.96
Total				0.96

Table 1 : Power reduction on area-opt. MCNC/ISCAS-89 circuits

7 Conclusions

We have addressed the problem of technology-dependent power optimization and hot spot reduction. The main contributions of this work are:

- An EC based formulation of the problem of resynthesis for low power which allows the adaptation of EC algorithms to power minimization. The proposed approach constructs a target function for EC, such that any implementation satisfying it is expected to reduce power without changing the circuit area, gate capacitance or delay under the unit delay model.
- A unified framework combining the theory of [5] and [6] to allow global power sensitivities (which account for the affect of TFO power change for local functional changes) to be defined for networks with arbitrary input probability distributions.
- New theory for estimating the expected change in onset size of a logic function during synthesis given a specific flexibility. This was combined with the global sensitivities to identify hot nodes in the circuit and construct the target function for EC.

Power reductions of up to 13%, with an average of 4% were achieved on a set of MCNC benchmark circuits.

References

[1] R. Bahar, M. Burns, G. Hachtel, E. Macii, H. Shin, and F. Somenzi, "Symbolic computation of logic implications for technology-dependent low-power synthesis," *ISLPED 96*

[2] A. Ghosh, S. Devadas, K. Keutzer, J. White, "Estimation of average switching activity in combinational and sequential circuits," *DAC 92.*

[3] Y. Kukimoto, M. Fujita, R. Brayton, "A redesign technique for combinational circuit based on gate reconnections," *ICCAD 94.*

[4] C. Lennard, A. Newton, "An estimation technique to guide low power resynthesis algorithms," *ISLPD 95.*

[5] C. Lennard, A. Newton, "On estimation accuracy for guiding low-power resynthesis," *IEEE Trans. CAD*, Vol. 15, No. 6, June 1996.

[6] C. Lennard, P. Buch, A. Newton, "Logic synthesis using power sensitive don't care sets," *ISLPED 96.*

[7] B. Lin, H. De Mann, "Low-power driven technology mapping under timing constraints," *IWLS 93.*

[8] B. Rohfleisch, A. Kölbel, B. Wurth, "Reducing power dissipation after technology mapping by structural transformations," *DAC 96.*

[9] H. Savoj, R. Brayton, "The use of observability and external don't cares for the simplification of multi-level networks," *DAC 90.*

[10] V. Tiwari, P. Ashar, S. Malik, "Technology mapping for low power," *DAC 93.*

[11] C. Tsui, M. Pedram, A. Despain, "Technology decomposition and mapping targeting low power dissipation," *DAC 93.*