PowerShake: A Low Power Driven Clustering and Factoring Methodology for Boolean Expressions

Sumit Roy
Ambit Design Systems
Santa Clara, USA
sroy, harm@ambit.com

Harm Arts
Ambit Design Systems
Santa Clara, USA
sroy, harm@ambit.com

Prithviraj Banerjee†
Center for Parallel & Distributed Computing
Northwestern University, USA
banerjee@ece.nwu.edu

Abstract

This paper describes algebraic techniques that target low power consumption. A unique power cost function based on decomposed factored form representation of a Boolean expression is introduced to guide the structural transformations. Circuits synthesized by the SIS [5] and POSE [1] consume 54.5% and 10.4% more power than that obtained by our tool respectively.

1 Introduction

Recent trends in digital circuit technology have made power optimization an important factor in the design of digital systems. Although, power optimization can be applied at different levels of the design, in this work, we focus on algebraic restructuring techniques, namely algebraic factorization and collapsing, to reduce power at the combinational logic level.

Previous research efforts have been in identifying suitable power cost function for guiding the logic synthesis. In [4], the power gain for extracting a subexpression was defined as the difference between the power cost of the expression (based on the same factored form) before and after the extraction. One shortcoming of [4] is that once the sub-expressions are extracted, the expression will not necessarily have the assumed factored form. In [1], a cost function based on the sum-of-products (SOP) form of a expression was introduced to alleviate the above inaccuracy. Although it performs well for two-level implementation of circuits, it can be potentially inaccurate for multi-level implementation since the cost function in [1] cannot distinguish between difference in the power consumed by the several factored forms of a given expression. In this paper, a new cost function is defined that estimates accurately the average power consumption of the mapped circuit. We propose a new methodology based on iterative clustering and algebraic factoring.

2 Cost function

In this section, we introduce an alternative approach for computing the power cost based on the factored form representation [7].

This research was supported in part by the Semiconductor Research Corporation under Contract SRC 95-DP-109 and the Advanced Research Projects Agency under contract DAAH04-94-G-0273 administered by the Army Research Office.
3.1 Algebraic Factorization

Algebraic factorization identifies common expressions and introduces them as new nodes into a network in order to optimize a given cost function. A recent algebraic factorization technique extracts only two-cube divisors and two-literal single-cube divisors both in normal and complement form [6]. It uses the weight of a divisor to measure the magnitude of area saving it brings about. We have extended it for power optimization by introducing our power cost as the weight of the divisors.

We estimate the gain in power of extracting a divisor by taking the difference of the power cost of the Boolean expression before and after the extraction. Note that we refactorize the extracted expression to obtain its power cost unlike the approach in [4].

3.2 Clustering

Clustering is the technique of selectively collapsing a set of Boolean expressions into a single expression such that it provides a better structure for the circuit. It tries to cluster nodes with high switching activity into a single expression such that the latter can be implicitly decomposed into nodes with smaller switching activity.

Example 3.1 Let \( F = K * L, G = L + e, H = L + f, K = a * b, L = c * d \) be a given circuit with \( p_L(a) = 0.5, p_L(b) = 0.85, p_L(e) = 0.2, p_L(d) = 0.85, p_L(e) = 0.5 \) and \( p_H(f) = 0.5 \). \( (p_L(x) = \text{signal probability of } x) \]

The initial circuit has expressions \( K \) and \( L \) with high switching activity \( p_L(K) = 0.499, p_L(L) = 0.308 \). Clustering identifies them, collapses into \( F \) and finally implicitly decomposes it into \( F = m + d, m = n + b, n = a + c \) by introducing less active nodes \( m \) and \( n(p_L(m) = 0.174, p_L(n) = 0.211) \). Clustering proceeds as follows: each expression \( x \) has all its fanout \( y_1, y_2, \ldots, y_k, k > 0 \) as clustering candidates. It evaluates the transformation of collapsing into \( y_k \) by estimating the new value of \( P(y_k) \). As part of the estimation, \( y_k \) gets implicitly decomposed and its power cost calculated. If the new power cost of \( y_k \) is better than the old cost of \( y_k \), \( x \) is collapsed into \( y_k \). If \( x \) gets collapsed into all its fanouts, the power cost associated with it gets added to the power weight. In [1], a similar technique known as selective collapse has been reported where it selectively eliminating nodes in a network in order to reduce the power cost of the network. Our clustering algorithm is superior to the published algorithm for selective collapse [1] in two ways: it restructures circuits to provide a better power structure using implicit decomposition; it can partially collapse a node into some of its fanouts which the approach in [1] cannot.

4 Experimental Results

We developed a tool PowerShake which takes in a multi-level network and the input switching activity and performs algebraic restructuring. Initially the switching activity of all the internal nodes are computed based on the input switching activity. Then it iteratively performs clustering and algebraic factorization based on the techniques discussed in Section 3 on the expressions in the circuit till the average power consumption of the circuit cannot be reduced.

We compared our tool, PowerShake, with SIS [5] (script:algebraic) and POSE [1] (script:power) on the first 35 circuits (sorted according to initial size) from the MCNC benchmark. In each case, we optimized the circuit using each of the tools (SIS, POSE, PowerShake), mapped it to mcnc.genlib using power_estimate in SIS at 20MHz. We noticed that SIS optimized circuits consume 54.5% more power than that obtained by PowerShake on an average over the 35 circuits. Also, POSE optimized circuits on an average consume 10.4% more power than our approach. In particular, it can consume up to 77.19% more power than obtained using our tool. Due to shortage of space, we tabulate the results obtained by the 8 largest circuits amongst the given 35 MCNC circuits in Table 1.

<table>
<thead>
<tr>
<th>Circuit</th>
<th>SIS</th>
<th>POSE</th>
<th>PowerShake</th>
</tr>
</thead>
<tbody>
<tr>
<td>a1</td>
<td>p1</td>
<td>a2</td>
<td>p2</td>
</tr>
<tr>
<td>square</td>
<td>191</td>
<td>442</td>
<td>201</td>
</tr>
<tr>
<td>sct</td>
<td>195</td>
<td>677</td>
<td>145</td>
</tr>
<tr>
<td>sqrt8x4</td>
<td>191</td>
<td>643</td>
<td>144</td>
</tr>
<tr>
<td>unreg</td>
<td>147</td>
<td>725</td>
<td>155</td>
</tr>
<tr>
<td>comp</td>
<td>217</td>
<td>791</td>
<td>251</td>
</tr>
<tr>
<td>i3</td>
<td>186</td>
<td>859</td>
<td>186</td>
</tr>
<tr>
<td>o64</td>
<td>186</td>
<td>835</td>
<td>-</td>
</tr>
<tr>
<td>9symml</td>
<td>331</td>
<td>1227</td>
<td>344</td>
</tr>
</tbody>
</table>

In conclusion, we have proposed a novel and accurate power cost function based on decomposed factored form representation of a Boolean expression. We also developed a synthesis methodology based on clustering and factorization. Finally, our experimental results show that our approach is quite promising.

References