Co-Optimization For Test Data Compression and Testing Power Based On Variable-Tail Code

Yinhe Han Yongjun Xu Xiaowei Li
(Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, P.R China)
(Graduate School of the Chinese Academy of Sciences, Beijing, 100039, P.R China)
Email: {yinhes, lxx}@ict.ac.cn

Abstract
We present a novel code: Variable-Tail code to compress the test data, which is proved to be better than the Golomb code. An efficient reordering algorithm(ERA) is proposed to optimize the distribution of run-lengths in the test data in order to improve the compression effectiveness. On the other hand, since the power during testing is always higher than the normal functional mode, the power dissipated in the CUT must be reduced during testing which is omitted previously. The analysis and experiments have demonstrated that our ERA can significantly reduce the total testing power and average power dissipated in the CUT with no hardware penalty.

Keywords: Golomb Code, Variable-Tail Code, total transitions count(TTC), total, hamming distance(THD)

1.Introduction
Embedded intellectual property (IP) cores are now increasingly used in large system-on-a-chip designs. However, too many IP cores integrated in a chip pose major test challenges. Two problems that are becoming increasingly important are test data volume and testing power. In order to be able to reduce the test data volume, researchers have suggested various compression techniques coupled with building on-chip decompression circuits to decompress the test data. These efforts including some new codes such as Golomb code[3], FDR code[4], Alternating Run-length code[5], symmetric code[6] and Kay code[7], Mutation Encoding[11]. The overriding goals in the design of compression techniques are comprised of a superior compression ratio and minimal hardware overhead. Also, since there is a significantly higher switching activity during testing when compared to normal functional operation. Special care must be taken to ensure that maximum power allowance for the CUT can’t be exceeded. A number of the techniques to reduce testing power have been presented. These techniques including two classes: 1)Reducing the power dissipated in the scan chains. These methods in the literature are based on some structural methods and

algorithmic methods[5]. 2)Reducing the power dissipated in the CUT. These methods mainly comprise some reordering algorithms[1,2,8] since the testing power dissipated in the CUT is only determined by the order of the test vectors if the test set is given.

The two goals, reducing testing power and reducing test data volume, seem to be not always conflicting in [5,6]. The key idea of [5,6] is that through mapping the don’t care bit to 0, the significant reduction in both scan power and test data volume is obtained. A drawback of this simply mapping procedure is that it only focuses on reducing the scan power and doesn’t care the power of the CUT, which is a practical main source of power.

In this paper, we will focus on improving the test data compression through a novel code and a novel reordering algorithm which can optimize the distribution of run-lengths, and reduce the testing power dissipated in the CUT. The main contributions of this paper are listed below:

- We present a novel code: Variable-Tail Code(VTC) to compress the test data, which is proved to be more efficient than Golomb code.
- We define a compression optimization model to yield best-case compression and a testing power optimization model. Then we find they are consistent.
- An efficient greedy algorithm to reorder the test vectors is presented to solve the above two optimization models.

2.Compression/Decompression Using Variable-Tail Code(VTC)

The Variable-Tail code is also a variable-to-variable code which can map variable-length runs of 0s to variable-codewords. It is similar to the Golomb code[3]. The major difference between Variable-Tail code and Golomb code is that in Variable-Tail code the tail of every group isn’t equal length and is equal length in Golomb code. In Variable-Tail code, the tail size increase by one from group i to group i+1. So the number of run-lengths in one group is always half of its next group. We define the radix r as the tail size of the first group. For example r =2 in Figure 1, for the first group its tail contains 2 bits and so 4 run-lengths(0-3) are included, and for the second group its tail contains 3
3. Co-optimization for Testing Power and Compression

In this section, we will develop two greedy optimization models to optimize the test-data compression and the power dissipated in the CUT.

3.1 Optimize the testing power dissipated in the CUT

The power dissipation in CMOS circuit is dominated by the dynamic power dissipation that occurs at a node when it switches from 0 to 1 or from 1 to 0. Thus, when a complete set of test vectors $T_D = \{t_1, t_2, ..., t_n\}$ is applied, the total power during testing can be expressed [10]:

$$P_{test} = \sum_{all \text{vectors}} \frac{1}{2} \sum_{all \text{gates}} C_x E_x (svx)$$

However, calculation of $P_{test}$ is difficult as it depends on a number of circuit parameter. An alternative approach is to use the total transitions count (TTC) to estimate $P_{test}$, where

$$TTC = \sum_{all \text{vectors}} \sum_{all \text{gates}} \text{fan-out} \times E_x (svx)$$

A complete undirected graph can be constructed to compute the TTC in the circuit in which each vertex represents a test vector and each edge represents the number of transitions activated in the circuit after application of the vectors pair. We call this graph as "TTC-Graph". An example in Figure 2, or C17 a test sequence $T$ is applied:

![Figure 2 TTC-Graph of C17](image)

We can see that minimizing the power dissipated in the CUT is equal to find a hamiltonian path of the lowest-cost in TTC-Graph and this hamiltonian path can be mapped to an order of test vectors. Therefore, we can use a simple model to determine the order of test vectors which can minimize the power.

**[Model 1] Find a lowest-cost path in TTC-Graph**

**Subject to:** a given circuit and test vectors sequence

However, directly using TTC to optimize isn't practical because constructing the TTC-Graph need simulate the circuit to compute the number of transitions on each edge. Another main disadvantage of this method is that it needs the accurate gate net-list which can't be got for some hard cores. We find the hamming distance reorder can help to reduce the TTC. It can't solve the Model 1 completely but our experiments presented show it can significantly reduce the total power compared to the original test vectors and reduce the average power compared to the vectors generated by Minitest with static compaction.

3.2 Optimize the test compression

The effectiveness of compression can be measured by the compression gain $Y_{cg}$, which is defined:

$$Y_{cg} = \frac{L - L_E}{L} \times 100\%$$

where $L_E$ is the length of encoded codeword and $L$ that is length of original test sets. We can analyze the Variable-Tail code with the radix $r$ and group index $k$. This yields:

$$Y_{cg} = 1 - (1 - p)(\frac{2^k}{r^k} - (\sum_{i=0}^{r-1} p^r)^k) + r - 1$$

$p$ is the 0s probability in the test vectors sequence. From this expression, we can see if we chose the Variable-Tail code, another way to improve $Y_{cg}$ is to increase $p$. This indicates that we should reduce the hamming distance between all the vectors if we use difference vectors to compress.

We define **hamming distance between a vector pair** $t_i, t_{i+1}$ (each vector has $r$ bits) as:

$$HD(t_i, t_{i+1}) = \sum_{j=0}^{r-1} t_i(j) \oplus t_{i+1}(j)$$

We can use an undirected graph in which each vertex represents a test vector of $T_D$, each edge represents the hamming distances between a vectors pair. See example 2:

![Figure 3 HD-Graph of C17](image)

Similar to example 1, minimizing total hamming distances (TTC) between all test vectors can be changed to find a lowest-cost of path in HD-Graph. Because Figure 4 is simply, we can see the order $O(1, 3, 2, 4)$ is one of best-cases. The length of Variable-Tail codeword can be reduced from 18 to 16 if we use the order $O$. In formulation, a simply model can be constructed to optimize the test compression.
4. Efficient Reordering Algorithm (ERA)

From the analysis of model 1 and model 2, finding a low-cost path in HD-Graph or TTC-Graph both can be reduced to finding an order of test vectors. So how to reorder the vectors or which goal the reordering based on is important. In this paper, we present an algorithm which is based on hamming distance reordering. This is mainly used to solve model 2, but it can also help us to solve model 1 efficiently since the TTC relates to the hamming distance.

The minimizing hamming distance reordering is known to be a NP-Hard problem, so exact algorithms for solving this problem doesn’t exist. A greedy heuristic is present in the next algorithm to find a close-to-optimal solution.

Algorithm 1: Fast vectors reordering algorithm

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>%Best</td>
<td>m</td>
<td>r</td>
</tr>
<tr>
<td>CG</td>
<td></td>
<td>CG</td>
<td>CG</td>
</tr>
<tr>
<td>S13207</td>
<td>16 74.8</td>
<td>32 83.8</td>
<td>2</td>
</tr>
<tr>
<td>S15850</td>
<td>4 47.1</td>
<td>8 66.9</td>
<td>1</td>
</tr>
<tr>
<td>S35932</td>
<td>--</td>
<td>2 12.4</td>
<td>0</td>
</tr>
<tr>
<td>S38417</td>
<td>4 44.1</td>
<td>8 57.6</td>
<td>0</td>
</tr>
<tr>
<td>S38584</td>
<td>4 47.7</td>
<td>8 59.7</td>
<td>1</td>
</tr>
<tr>
<td>Avg</td>
<td>-- 53.4</td>
<td>-- 67</td>
<td>--</td>
</tr>
</tbody>
</table>

Table 1: Test data reduction using our Variable-Tail code and ERA (when calculating the avg. the shadow line is excluded since [3] doesn’t present the proper data)

Next, we will present details about the three codes in [3, 4, 6] and compare them in our proposed code. Table 2 lists the size of original test set, the minimum size of encoded test set in [3, 4, 6], corresponding parameters and the minimum size of the optimized and encoded test set using our ERA and Variable-Tail code.

We note that the Variable-Tail code provide better compression than Golomb code [3] in all cases. A round average 20% improvement of CG is obtained which can save the ATE memory and test time as indicated in [3]. Also the Variable-Tail code is better than the alternating run-length code which is considered the best code to compress.

Finally, we present the results on the power dissipated in the CUT in Table 3 for both combinational and full-scan sequential circuits when the complete test vectors are applied. We assume that the circuits are free-delay. The total transitions count (TTC) activated in the circuit and total hamming distance (THD) are given in Column 2 (before reordering) and Column 3 (after reordering). All the don’t care bits in the original test vectors will be assign 0 to calculate THD and TTC. From the results listed in Table 3 (we don’t list some smaller circuits in ISCAS85 and large circuits in ISCAS 89 because of the limited space), we can see that about 20% reduction in transitions count has been obtained on the average, for both the combinational and full-scan sequential circuits.
6. Conclusions

In this paper we introduce a novel code: Variable-Tail code to compress the test data. Our code outperforms Golomb code and is scalable, so an additional modification can be done, such as alternative or symmetric characteristic. We also focus on the distribution of the run-lengths and an efficient minimum Hamming distance reordering (ERA) algorithm is presented to improve the distribution as improving the compression gain. Another main advantage using the ERA is that the reordered test vectors have smaller Hamming distance so that the power dissipated in the CUT is significantly reduced (23%) when these test vectors are applied to the CUT. It does not bring the additional power penalty compared to the static compaction.

7. References


