System-on-Chip Test Parallelization Under Power Constraints

Erik Larsson and Zebo Peng

Department of Computer and Information Science, Linköpings University, Sweden

Abstract[1]

This paper deals with test parallelization (scan-chain subdivision) which is used as a technique to reduce test application time for systems-on-chip. An approach for test parallelization taking into account test conflicts and test power limitations is described. The main features of the proposed approach are the combination of test parallelization with test scheduling as well as the development of an extremely fast algorithm which can be used repeatedly in the design space exploration process. The efficiency and usefulness of our approach have been demonstrated with an industrial design.

1 Introduction

The increasing complexity of System-On-Chip (SOC) has created many testing problems, and long test application time is one of them. Minimization of test application time has become an important issue and several techniques have been developed for this purpose, including test scheduling [1], [2], [3], [4], and test vector set reduction[5].

The basic idea of test scheduling is to schedule tests in parallel so that many test activities can be performed concurrently. However, there are usually many conflicts, such as sharing of common resource, in a system under test which inhibit parallel testing. Therefore the test scheduling issue must be taken into account during the design of the system under test, in order to maximize the possibility for parallel test. Further, test power constraints must be considered carefully, otherwise the system under test may be damaged due to overheating. Chakrabarty showed that the test scheduling problem is equal to the open-shop scheduling [1] which is known to be NP-complete and the use of heuristics are therefore justified. In the approach by Chou *et al.* [3] a resource graph is used to model the system, and from it, a test compatibility graph (TCG) is generated (Figure 1).

We have recently proposed an integrated framework for the testing of SOC [6], which provides a design environment to treat test scheduling under test conflicts and test power constraints as well as test set selection, test resource placement and test access mechanism design in a systematic way. In this paper, the issue of test scheduling will be treated in depth, especially the problem of test parallelization. We will present a technique for test parallelization under test power consumption and show how it can be used to find the optimal test time for the system under test. Our technique is based on a greedy algorithm which runs fast and can be therefore used during the design space exploration process. The usefulness of the algorithm is demonstrated with an industrial design.

2 Preliminaries

2.1 System Modeling

An example of a system under test is given in Figure 2 where each core is placed in a wrapper in order to achieve efficient test isolation and to ease test access. In our approach, a system under test, such as the one shown in Figure 2, is by a notation,

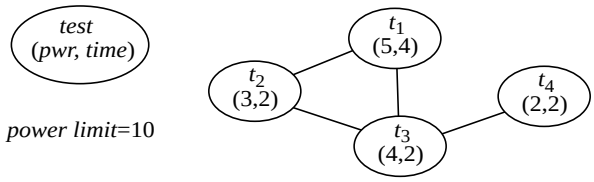

Figure 1. A TCG of the example system.

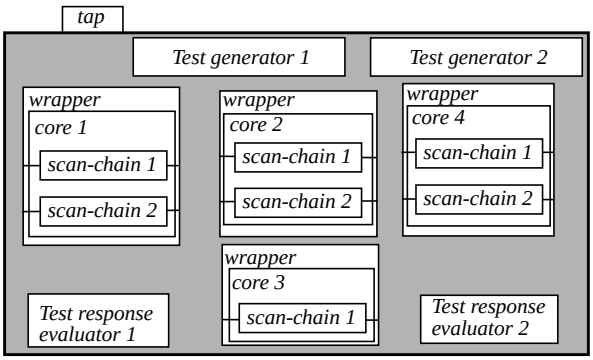

Figure 2. An illustrative example.

design with test, $DT = (C, R_{source}, R_{sink}, p_{max}, T, source, sink, constraint, bandwidth)$[2], where:

$C = \{c_1, c_2, ..., c_n\}$ is a finite set of cores and each core $c_i \in C$ is characterized by:

$p_{idle}(c_i)$: its idle power,

$par_{min}(c_i)$: its minimal parallelization degree, and

$par_{max}(c_i)$: its maximal parallelization degree;

$R_{source} = \{r_1, r_2, ..., r_m\}$ is a finite set of test sources;

$R_{sink} = \{r_1, r_2, ..., r_p\}$ is a finite set of test sinks;

p_{max}: maximal allowed power at any time;

$T = \{t_{11}, t_{12}, ..., t_{oq}\}$ is a finite set of tests, each consisting of a set of test vectors. And each core, c_i, is associated with several tests, t_{ij} (j=1,2,...,k). Each test t_{ij} is characterized by:

$t_{test}(t_{ij})$: test time at parallelization degree 1, $par(t_{ij})$=1,

$p_{test}(t_{ij})$: test power dissipated when test t_{ij} alone is applied;

source: $T \rightarrow R_{source}$ defines the test sources for the tests;

sink: $T \rightarrow R_{sink}$ defines the test sinks for the tests;

constraint: $T \rightarrow 2^C$ gives the cores required for a test;

bandwidth(r_i): bandwidth at test source $r_i \in R_{source}$.

2.2 Test Power Consumption

Generally speaking, there are more switching activities during the testing mode of a system than when it is operated under the normal mode. A simplification of the estimation of the power consumption was introduced by Chou et al. [3] and has been used by Zorian [2] and by Muresan *et al.* [4] and we will use this assumption also in our approach.

Aerts *et al*. have defined for scan-based designs the change of test time when a scan-chain is subdivided into several chains of shorter length[5]. In our approach, we use a formula which follows the idea introduced by Aerts *et al*.:

$$t'_{test}(t_{ij}) = \lceil (t_{test}(t_{ij}))/n_{ij} \rceil \qquad (1)$$

where n_{ij} is the degree of parallelization of a test t_{ij}.

1. This work has partially been supported by the Swedish National Board for Industrial and Technical Development (NUTEK).

2. This is a simplification of the model we used in [6].

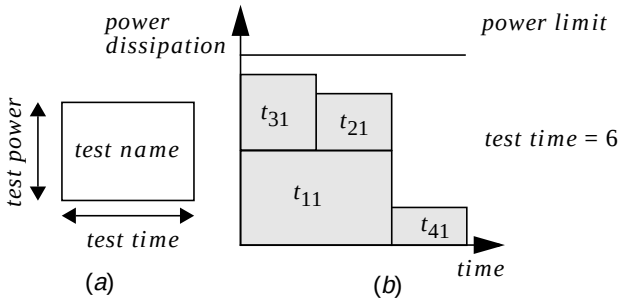

Figure 3. The test time and test test power consumption for a test (a) and the test schedule of the example system (b).

Gerstendörfer and Wunderlich investigated the test power consumption for scan-based BIST and used the weighted switching activity (WSA) defined as the number of switches multiplied by the capacitance [7]. The average power is WSA divided by the test time as a measure of the average power consumption for a test where WSA is defined as the number of switches multiplied by the capacitance. As a result, the test power increases as test time is reduced.

$$p'_{test}(t_{ij}) = p_{test}(t_{ij}) \times n_{ij} \qquad 2$$

The simplification defined in this section is used in order to discuss the impact on test time and test power. For our practical algorithms more accurate estimations are included.

3 Proposed Approach

3.3 Optimal Test Time

In this section we first discuss the possibility of achieving optimal test time with the help of test parallelization under power constraints.

A test schedule can be illustrated by placing all tests in a diagram as in Figure 3(b). At any moment the test power consumption must be below the maximal allowed power limit p_{max}. The rectangle where the vertical side is given by p_{max} and the horizontal side is defined by the total test application time t_{total} characterizes the test feature of a given system under test.

If the rectangle defined by $p_{max} \times t_{total}$ is equal to the summation of $t_{test}(t_{ij}) \times p_{test}(t_{ij})$ for all tests, as given by the following equation, we have the optimal solution.

$$\sum_{\forall i \forall j} t_{test}(t_{ij}) \times p_{test}(t_{ij}) = p_{max} \times t_{opt} \qquad 3$$

The optimal test time for a system under test is thus:

$$t_{opt} = \sum_{\forall i \forall j} \frac{t_{test}(t_{ij}) \times p_{test}(t_{ij})}{p_{max}} \qquad 4$$

Usually, the optimal test time cannot be achieved due to test conflicts. The worst case occur when all tests are in conflicts with each other and all tests must be scheduled in sequence. The total test time is then given by:

$$t_{sequence} = \sum_{\forall i \forall j} t_{test}(t_{ij}) \qquad 5$$

For a scan-based design the scan-chains can be divided into several which reduces the test application time. If every test t_{ij} is allowed to be parallelized by a factor n_{ij}, the total test time when all tests are scheduled in sequence is:

$$\sum_{\forall i \forall j} (t_{test}(t_{ij}))/n_{ij} \qquad 6$$

The lower bound of the degree of parallelization is $n_{ij} = 1$. For a scan-based core, it means a single scan-chain. The upper bound of the degree of parallelization is defined by the maximal test power consumption:

$$n_{ij} = p_{max}/(p_{test}(t_{ij})) \qquad 7$$

By using the upper bound as the degree of parallelization in combination with Equation 6, the following is obtained:

$$\sum_{\forall i \forall j} \frac{t_{test}(t_{ij})}{n_{ij}} = \left\{ n_{ij} \rightarrow \frac{p_{max}}{p_{test}(t_{ij})} \right\} =$$

$$\sum_{\forall i \forall j} \frac{t_{test}(t_{ij}) \times p_{test}(t_{ij})}{p_{max}} = t_{opt} \qquad 8$$

A testable unit is often tested by two test sets, one produced by an external test generator and one produced by BIST. A problem arises when two tests at a testable unit require different degree of parallelization. For instance, if a scan-chain is to be divided into n_{ij} chains at one moment and n_{ik} chains at another moment where $j \neq k$. However, if the core is placed in a wrapper such as P1500 [8] it is possible to allow different length of the scan-chain. As an example, in Figure 4, the bold wiring marks how to set up the wrapper in order to make the two scan-chains to be connected into a single scan-chain.

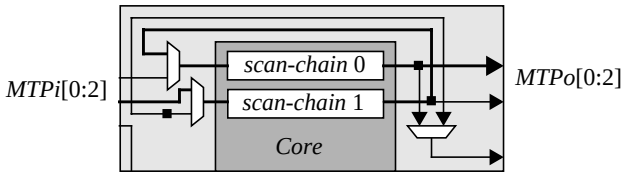

Figure 4. Part of a wrapper where the two scan-chains are connected to a single chain.

For a given core c_i tested by the tests t_{i1} *and* t_{i2}, we have two test sets each with its degree of parallelization calculated as n_{i1} and n_{i2}. It means that the number of scan-chains at c_i should, when test t_{i1} is applied, be n_{i1} and, when t_{i2} is applied, n_{i2}. For instance if n_{i1}=10 and n_{i2}=15 the number of scan-chains are 30 which is the *least common multiple* (*lcm*). This means that we also generalize our solution to make it applicable to an arbitrary number of tests per block.

3.4 Optimal Test Algorithm

The optimal test scheduling algorithm is illustrated in Figure 5. The time τ determines when a test is to start and it is initially set to zero. In each iteration over the set of cores and the set of tests at a core, the degree of parallelization n_{ij} is computed for the test t_{ij}; its new test time is calculated; and the starting time for the test is set to τ. Finally τ is increased by $t_{test}(t_{ij})/n_{ij}$. When the parallelization is calculated for all tests at a core, the final degree of parallelization can be computed.

```
τ = 0;
for all cores c_i
   for all tests t_ij at core c_i
      n_ij = p_max / p_test(t_ij)
      start test t_ij at time τ;
      τ=τ+t_test(t_ij)/n_ij;
   n_i = lcm(n_i1,..., n_in)
```

Figure 5. Optimal test parallelization algorithm.

The algorithm consists of a loop over the set of cores and at each core a loop over the set of its test, it corresponds to a loop over all tests resulting in a complexity $O(|T|)$ where $|T|$ is the number of tests.

Sort T according to the key (p, t or p×t) and store the result in P;
Schedule S=∅, τ=0;
Repeat until P=∅
  For all tests t_{ij} in P do
    n_{ij}=*min*{⌊ *available power during* [τ, τ+$t_{test}(t_{ij})$]/ $p_{test}(t_{ij})$⌋,
      $par_{max}(c_i)$, *available bandwidth during* [τ, τ+$t_{test}(t_{ij})$]}
    $\tau_{end}=\tau+t_{test}(t_{ij})$
    $p_{test}(t_{ij})=p_{test}(t_{ij})\times n_{ij}$;
    If all constraints are satisfied during [τ, τ_{end}] *then*
      Insert t_{ij} in S with starting at time τ;
      Remove t_{ij} from P;
  τ = *nexttime*(τ);

Figure 6. The system test algorithm.

3.5 Practical limitations

The optimal degree of parallelization for a test t_i has been defined as $p_{max}/p_{test}(t_{ij})$ (Equation 7). However, such division does not usually give an integer result. The practical degree of parallelization for a test t_i is given by:

$$n_{ij} = \lfloor p_{max} / (p_{test}(t_{ij})) \rfloor \qquad 9$$

For each test t_{ij}, the difference between the optimal and the practical degree of parallelization is given by:

$$P_{max} = p_{test}(t_{ij}) \times n_{ij} + \Delta_{ij} \qquad 10$$

and the difference Δ_{ij} for each test t_{ij} is given by:

$$\Delta_{ij} = p_{test}(t_{ij}) \times n_{ij} - p_{test}(t_{ij}) \times \lfloor n_{ij} \rfloor = p_{test}(t_{ij}) \times (n_{ij} - \lfloor n_{ij} \rfloor) \qquad 11$$

Δ_i reaches its maximum when n_{ij}-⌊n_{ij} ⌋ is approximately 1 which occur when n_{ij} = 0.99.. leading to $\Delta_{ij} \approx p_{test}(t_{ij})$. The worst case test time occurs when $\Delta_{ij} \approx p_{test}(t_{ij})$ for all test t_{ij} and n_{ij} = 1, resulting in a test time given by Equation 6 which is equal to $t_{sequence}$ computed using Equation 5 since n_{ij} = 1.

We now show the difference between the worst case test time for the system and its optimal test time. The worst case occurred when $\Delta_{ij} = p_{test}(t_{ij})$ and n_{ij}= 0.99... which in Equation 11 results in the following:

$$P_{max} = p_{test}(t_{ij}) + p_{test}(t_{ij}) \qquad 12$$

which only has one solution, $p_{test}(p_{ij}) = P_{max} / 2$ (assuming $P_{max} > p_{test}(t_{ij}) > 0$). However, we can not make any conclusions in respect to test time since two tests t_{ij} and t_{ik} may have equal test power consumption but different test time. The difference between the optimal test time and the worst total test time given by:

$$\sum_{\forall i \forall j} t_{test}(t_{ij}) - \sum_{\forall i \forall j} \frac{t_{test}(t_{ij})}{2} = \sum_{\forall i \forall j} \frac{t_{test}(t_{ij})}{2} \qquad 13$$

This motivates the use of an integrated test scheduling and test parallelization approach.

3.6 Test Scheduling and Test Parallelization Algorithm

In this section, we outline the test scheduling and test parallelization part of the algorithm and leave the function for *constraint checking* and *nexttime* out. The tests are initially sorted based on either *power* (*p*), *time*(*t*) or *power*×*time* (*p*×*t*) and placed in P (Figure 6). An iteration is performed until P is empty (all tests are scheduled). For all tests in P at a certain time τ, the maximal possible parallelization is determined as the minimum among:

- ⌊ *available power during* [τ, τ+$t_{test}(t_{ij})$]/ $p_{test}(t_{ij})$⌋,
- $par_{max}(c_i)$, and
- *available bandwidth during* [τ, τ+$t_{test}(t_{ij})$].

The constraints are checked and if all are satisfied, the test is scheduled in S at time τ and removed from P.

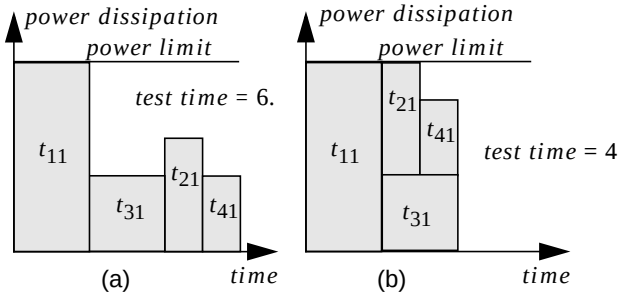

Figure 7. The test schedule of the example design using test parallelization (a) and combined test parallelization and test scheduling (b).

4 Experimental Results

We have performed experiments on a design example and an industrial design. For the design example (Figure 2) with the TCG in Figure 1 all tests are allowed to be parallelized by a factor 2 except for test t_{31} which is fixed. The test schedule when not allowing test parallelization results in a test time of 6 time units (Figure 3(b)) and when only test parallelization is used the test time is also 6 time units (Figure 7(a)). However, when combining test scheduling test parallelization the test time is reduced to 4 time units (Figure 7(b)).

The industrial design [6] and a designers solution requires a test application time of 1592 and using the test scheduling approach we proposed [6] results in a test schedule a test application time of 1077 which is an improvement of the designers solution with 32%. The test schedule achieved using the approach proposed in this paper results in a test time of 383.

5 Conclusions

We have investigated the effect of test parallelization under test power constraints and shown that the optimal solution for test application time can be found in the ideal case and we have defined an algorithm for it. Furthermore, we have showed that practical limitations may make it impossible to find the optimal solution and therefore test parallelization must be integrated into the test scheduling process. We have performed experiments on an industrial design to show the efficiency of the proposed technique.

References

[1] K. Chakrabarty, Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming, *Trans. CAD of IC and Systems*, Vol. 19, No. 10, pp. 1163-1174, Oct. 2000.

[2] Y. Zorian, A distributed BIST control scheme for complex VLSI devices, *Proc. of VLSI Test Symp.*, pp. 4-9, April 1993.

[3] R. Chou et al., Scheduling Tests for VLSI Systems Under Power Constraints, *Transactions on VLSI Systems*, Vol. 5, No. 2, pp. 175-185, June 1997.

[4] V. Muresan et al., A Comparison of Classical Scheduling Approaches in Power-Constrained Block-Test Scheduling, *Proceedings of Int. Test Conference*, pp. 882-891, Oct. 2000.

[5] J. Aerts, E. J. Marinissen, Scan Chain Design for Test Time Reduction in Core-Based ICs, *Proceedings of Int. Test Conference*, pp 448-457, Washington,DC, Oct. 1998.

[6] E. Larsson, Z. Peng, An Integrated System-On-Chip Test Framework, *DATE*, Munchen, Germany, March 13-16, 2001.

[7] S. Gerstendörfer, H-J Wunderlich, Minimized Power Consumption for Scan-Based BIST, *Proc. of International Test Conference*, pp 77-84, Atlantic City, NJ, Sep. 1999.

[8] E. J. Marinissen et al., Towards a Standard for Embedded Core Test: An Example, *Proceedings of International Test Conference*, pp. 616-627, Atlantic City, NJ, Sep. 1999.