# Test Scheduling and Scan-Chain Division Under Power Constraint

Erik Larsson and Zebo Peng

Department of Computer and Information Science,

Linköpings University, Sweden

## Abstract[1]

*An integrated technique for test scheduling and scan-chain division under power constraints is proposed in this paper. We demonstrate that optimal test time can be achieved for systems tested by an arbitrary number of tests per core using scan-chain division and we define an algorithm for it. The design of wrappers to allow different lengths of scan-chains per core is also outlined. We investigate the practical limitations of such wrapper design and make a worst case analysis that motivates our integrated test scheduling and scan-chain division algorithm. 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 test problems, and long test application time is one of them. Minimization of test 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.

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 scan-chain division (test parallelization). We will present a technique for test parallelization under test power constraints and demonstrate 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.

The rest of the paper is organized as follows. Related work is described in Section 2, and preliminaries are given in Section 3. The details of our approach are presented in Section 4. The paper is concluded with experimental results in Section 5 and conclusions in Section 6.

## 2 Related Work

### 2.1 Test Scheduling

Scheduling the tests in a system means that the start time and end time for each test are determined in such a way that all constraints are satisfied and the test time is minimized. Chakrabarty showed that the test scheduling problem where each test is denoted with a fixed test time is equal to the open-shop scheduling [1] which is known to be NP-complete and the use of heuristics are therefore justified.

Several techniques to minimize the test time have been proposed. For instance, Chakrabarty proposed a test scheduling technique for core-based systems considering test conflicts [1]. Zorian has proposed a technique for test time minimization under test power limitations of Built-In Self-Test (BIST) systems [2]. The test conflicts in such systems are few dues to that each testable unit has its dedicated test resources.

For general systems, Chou *et al.* [3] and Muresan *et al.* [4] have proposed techniques to minimize test time under power limitations and conflicts. In the approach by Chou *et al.* [3] a resource graph is used to model the system where an arc between a test and a resource indicate that the resource is required for the test, Figure 1. From the resource graph, a test compatibility graph (TCG) is generated (Figure 2) where each test is a node and an arc between two nodes indicate that the tests can be scheduled concurrently. For instance $t_1$ and $t_2$ can be scheduled at the same time. Each test is attached with its test time and its power consumption and the maximal allowed power consumption is 10. The tests $t_1$, $t_2$, $t_3$ are compatible, however, due to the power limit they can not be scheduled at the same time.

### 2.2 Test Parallelization

By test parallelization we mean that the test vectors in a given test are rearranged in such a way that several tests can be executed in parallel. For a scan-based design, each test vector is shifted in (scanned in), and after applying a capture cycle, the test response is shifted out (scanned out). Even if

[1] This work has partially been supported by the Swedish Agency for Innovation Systems (VINNOVA).

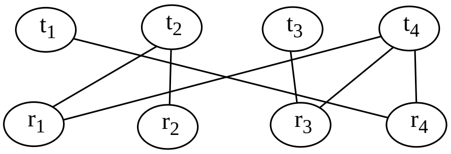

**Figure 1. Resource graph of an example system.**

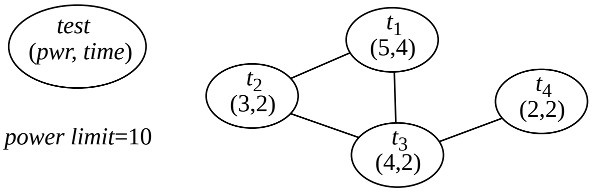

**Figure 2. Test compatibility graph (TCG) of the example system (Figure 1).**

a new test vector is shifted in at the same time as the test response from the previous test vector is shifted out, the shift-in and shift-out process contributes to a major part of the test time due to the length of the scan-chain (number of flip-flops). By dividing a scan-chain into several chains of shorter length, the test time is reduced.

Another advantage with test parallelization, beside test time minimization, is that the time a resource is required for a particular test is reduced, which reduces the impact of test conflicts. For instance, if test $t_4$ that requires $r_1$ and $r_4$, in the example given in Figure 1, is parallelized by a factor 2, the time when $r_1$ and $r_4$ is used by $t_4$ is reduced to 1.

Aerts *et al.* [5] have investigated the problem of dividing scan-chains for test time minimization where the constraints are defined by available pins (bandwidth). We focus on the limitations defined by maximal power consumption and test resources conflicts. However, for the integrated test scheduling and scan-chain division algorithm, bandwidth limitations are considered.

# 3 Preliminaries

## 3.1 System Modeling

An example of a system under test is given in Figure 3 where each core is placed in a wrapper in order to achieve efficient test isolation and to ease test access. Each core consists of at least one block with added DFT technique and in this example all blocks are tested using the scan technique. The test access port (*tap*) is the connection to an external tester and the test resources, *test generator* (TG) 1, *test generator* 2, *test response evaluator* (TRE) 1 and *test response evaluator* 2, are implemented on the chip.

Applying several sets of tests where each set is created at some test generator (source) and the test response is analysed at some test response evaluator (sink) tests the system.

In our approach, a system under test, such as the one shown in Figure 3, is by a notation, *design with test*, $DT = (C, R_{source}, R_{sink}, p_{max}, T, source, sink, constraint, bandwidth)$[2], where:

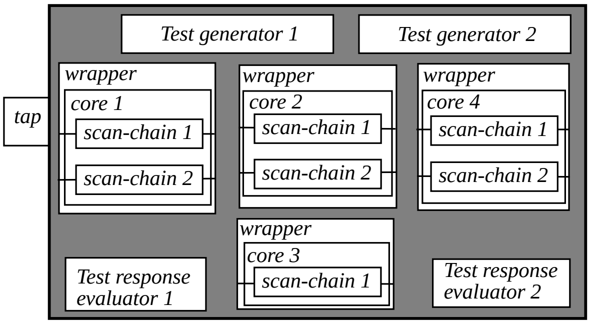

**Figure 3. An illustrative example.**

$C = \{c_1, c_2,..., c_n\}$ is a finite set of cores and each core $c_i \in C$ is characterized by:
$p_{idle}(c_i)$: idle power,
$par_{min}(c_i)$: minimal parallelization degree, and
$par_{max}(c_i)$: 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 at parallelization degree 1, $par(t_{ij})$=1,
*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 $r_i \in R_{source}$.

If the system in Figure 3 is tested by one test per core ($j$=1) and $r_1$ is TG1 /TRE1, $r_2$ is a shared test bus, $r_3$ is TG2/ TRE2 and $r_4$ is the tap, the test resource graph given in Figure 1 is valid for the system.

## 3.2 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. The power consumption of a CMOS circuit is given by a static part and a dynamic part. The dynamic part dominates and can be characterized by:

$$p = C \times V^2 \times f \times \alpha \qquad 1$$

where the capacitance $C$, the voltage $V$, and the clock frequency $f$ are fixed for a given design [7]. The switch activity $\alpha$, on the other hand, depends on the input to the system which during test mode are test vectors and therefore the power dissipation vary depending on the test vectors.

An example illustrating the test power dissipation variation over time $\tau$ for two test $t_i$ and $t_j$ is given in Figure 4. Let $p_i(\tau)$ and $p_j(\tau)$ be the instantaneous power dissipation of two compatible tests $t_i$ and $t_j$, respectively,

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

and $P(t_i)$ and $P(t_j)$ be the corresponding maximal power dissipation.

If $p_i(\tau) + p_j(\tau) < P_{max}$, the two tests can be scheduled at the same time. However, instantaneous power of each test vector is hard to obtain. To simplify the analysis, a fixed value $p_{test}(t_i)$ is usually assigned for all test vectors in a test $t_i$ such that when the test is performed the power dissipation is no more then $p_{test}(t_i)$ at any moment.

The $p_{test}(t_i)$ can be assigned as the average power dissipation over all test vectors in $t_i$ or as the maximum power dissipation over all test vectors in $t_i$. The former approach could be too optimistic, leading to an undesirable test schedule which exceeds the test power constraints. The latter could be too pessimistic; however, it guarantees that the power dissipation will satisfy the constraints. Usually, in a test environment the difference between the average and the maximal power dissipation for each test is often small since the objective is to maximize the circuit activity so that it can be tested in the shortest possible time [3]. Therefore, the definition of power dissipation $p_{test}(t_i)$ for a test $t_i$ is usually assigned to the maximal test power dissipation ($P(t_i)$) when test $t_i$ alone is applied to the device. This simplification was introduced by Chou et al. [3] and has been used by Zorian [2] and by Muresan *et al.* [4]. We will use this assumption also in our approach.

For the parallelization of a particular test a model is also required. Aerts *et al*. have defined such formulas for scan-based designs to determine the change of test time when a scan-chain is subdivided into several chains of shorter length[5], the test time for a test $t_i$ is given by:

$$t_{test}(t_i) = (tv_i + 1) \times \lceil f_i / n_i \rceil + tv_i \qquad 2$$

at a core with $f_i$ scanned flip-flops, $n_i$ number of scan-chains, and $tv_i$ test vectors. The formulas assume that a new test vector is scanned in at the same time as the test response is shifted out. This scheme is applicable for all test vectors but when the test response from the last test vector is shifted out and therefore the term +1 is added in Equation 2.

In our approach, we use the a formula which follows the idea introduced by Aerts *et al*., namely:

$$t'_{test}(t_{ij}) = \lceil (t_{test}(t_{ij}))/n_{ij} \rceil \qquad 3$$

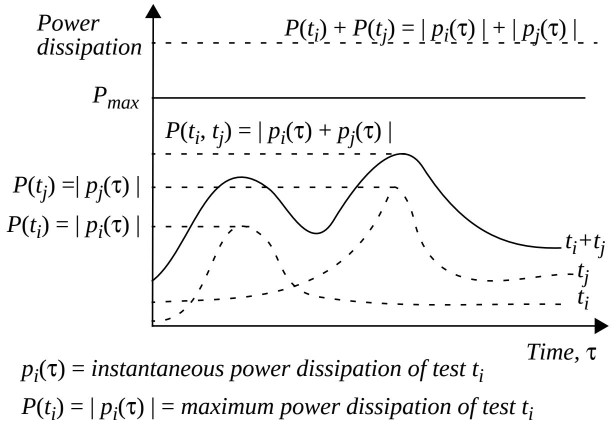

**Figure 4. Power dissipation as a function of time [3].**

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

Finally, we need an estimation on the relation between test power and test time when parallelizing a test. When a test is parallelizad and the test time is reduced, three options are possible for the change of test power, namely: (1) not affected, (2) decreased or (3) increased.

If the test power is not affected (option 1) or if it is decreased (option 2) while the test time is reduced, it is desirable to parallelize the test as much as possible.

The worst case occurs when the test power increases after a test parallelization since it means that the maximal power limit must be considered in order not to damage the system. In this paper we investigat the worst case.

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 [8]. 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 [8]. As a result, when the test time decrease, the test power increases:

$$p'_{test}(t_{ij}) = p_{test}(t_{ij}) \times n_{ij} \qquad 4$$

The simplifications we have defined in this section are used in order to discuss the impact on test time and test power. Especially note that the assumption in Equation 4 is a worst case assumption. For instance, if the test time for a test is reduced by a factor 2, the test power increases by a factor 2.

### 3.3 Test Wrapper Design

Test conflicts can be minimized by placing the core in a wrapper such as the TestShell proposed by Marinissen *et al.* [9]. A standard under development is the IEEE P1500 Standard for Embedded Core Test, consisting of a Core Test Language and a Core Test Wrapper [10] (Figure 5). The P1500 wrapper is similar to the TestShell. A major difference between TestShell and P1500 is that the latter only allow a single bit bypasses while the TestShell allows a TAM wide bypass.

Recently, Marinissen *et al*. proposed a library of wrapper cells allowing a flexible design [11]. For instance, it is possible to design non-clocked bypass structures of TAM width.

## 4 Proposed Approach

### 4.1 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. We assume a given system to be modelled as described in Section 3.1 where each test has a test time and a test power consumption attached to it. This can be illustrated using a rectangle for each test (as shown in Figure 6(a)) where the horizontal side corresponds to its

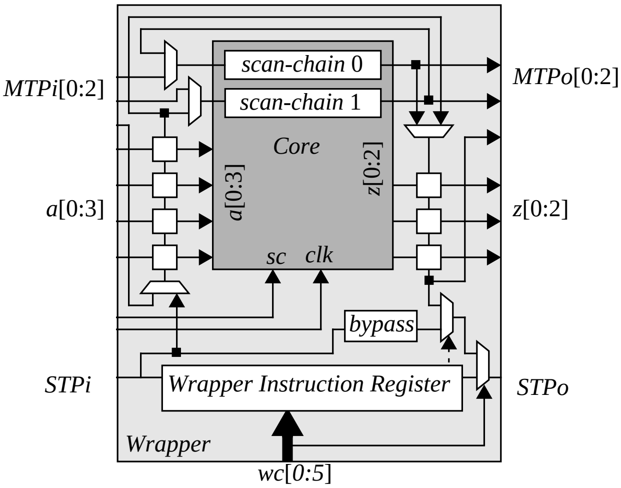

**Figure 5. Conceptual view of P1500 [10].**

test time while the vertical side corresponds to its test power consumption.

A test schedule can be illustrated by placing all tests in a diagram as in Figure 6(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.

$$\sum_{\forall i \forall j} t_{test}(t_{ij}) \times p_{test}(t_{ij}) = p_{max} \times t_{opt} \qquad 5$$

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

$$t_{opt} = \sum_{\forall i \forall j} \frac{t_{test}(t_{ij}) \times p_{test}(t_{ij})}{p_{max}} \qquad 6$$

Usually, the optimal test time cannot be achieved due to test conflicts. The worst case occurs 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:

$$t_{sequence} = \sum_{\forall i \forall j} t_{test}(t_{ij}) \qquad 7$$

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:

$$\sum_{\forall i \forall j} (t_{test}(t_{ij})) / n_{ij} \qquad 8$$

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:

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

By using the upper bound as the degree of parallelization in combination with Equation 8, the following is obtained:

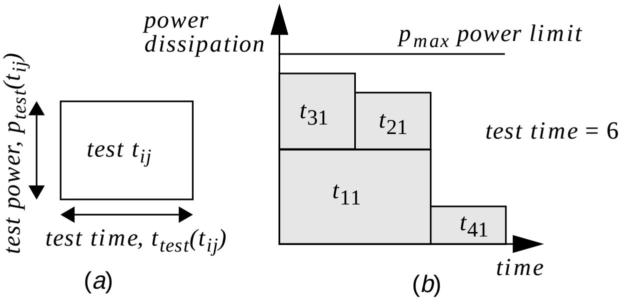

**Figure 6. The test time and test power consumption for a test (a) and the test schedule of the example system (Figure 2) (b).**

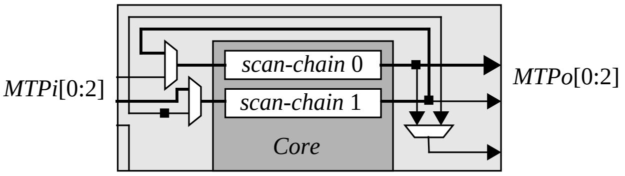

**Figure 7. Part of Figure 5 where the two scan-chains are connected to a single chain.**

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

$$\sum_{\forall i \forall j} \frac{t_{test}(t_{ij}) \times p_{test}(t_{ij})}{p_{max}} = t_{opt} \qquad 10$$

The above equation indicates the possibility to obtain optimal test time by parallelization, in theory. However, in the analysis, it is assumed that we have only one test set per block or that all test sets for a core are considered as a single test. In such case, the above analysis is valid. However, 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 the degree of parallelization of two tests at a testable unit require different degree of parallelization. For instance, a scan-chain is to be divided into $n_{ij}$ chains at one moment and into $n_{ik}$ chains at another moment where $j \neq k$. However, if the core is placed in a wrapper such as P1500 it is possible to allow different lengths of the scan-chains. As an example, in Figure 7, 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.

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 given by 2×5×3=30 which is *least common multiplier* (*lcm*). This means that we also generalize our solution to make it applicable to an arbitrary number of tests per testable unit (core).

## 4.2 Optimal Test Algorithm

The optimal test scheduling algorithm is illustrated in Figure 8. The time $\tau$ 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 $\tau$. Finally $\tau$ 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.

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.

```
$\tau = 0$;
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 $\tau$;
        $\tau=\tau+t_{test}(t_{ij})/n_{ij}$;
    $n_i = lcm(n_{i1},..., n_{in})$
```

**Figure 8. Optimal test parallelization algorithm.**

## 4.3 Practical limitations

The optimal degree of parallelization for a test $t_i$ has been defined as $p_{max}/p_{test}(t_{ij})$ (Equation 9). However, such division does not usually give an integer result. For instance, assume a system with a maximal test power consumption as $p_{max}$ = 10 and the test power for a test $t_{ij}$ at a scan-based core as $p_{test}(t_{ij}) = 4$. In this case $n_{ij} = 2.5$. However, the number of scan-chains in a core can not be 2.5. In practice, $n_{ij}$ should be rounded down, in this case into 2 (rounding up to 3 leads to a test power of 12, which is bigger than $p_{max}$). The practical degree of parallelization for a test $t_i$ is given by:

$$n_{ij} = \lfloor p_{max}/(p_{test}(t_{ij})) \rfloor \qquad 11$$

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

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

and the difference $\Delta_{ij}$ for each test $t_{ij}$ is given by:

$$\Delta_{ij} = p_{test}(t_{ij}) \times n_{ij} - p_{test}(t_{ij}) \times \lfloor n_{ij} \rfloor = p_{test}(t_{ij}) \times (n_{ij} - \lfloor n_{ij} \rfloor) \qquad 13$$

$\Delta_i$ reaches its maximum when $n_{ij}$-$\lfloor n_{ij} \rfloor$ 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 8 which is equal to $t_{sequence}$ computed using Equation 7 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 13 results in the following:

$$P_{max} = p_{test}(t_{ij}) + p_{test}(t_{ij}) \qquad 14$$

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 test $t_{ij}$ 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:

$$\sum_{\forall i \forall j} t_{test}(t_{ij}) - \sum_{\forall i \forall j} \frac{t_{test}(t_{ij})}{2} = \sum_{\forall i \forall j} \frac{t_{test}(t_{ij})}{2} \qquad 15$$

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

**Figure 9. The system test algorithm.**

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

## 4.4 An Integrated 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 9). Iterations are performed until P is empty (all tests are scheduled). For all tests in P at a certain time $\tau$, the maximal possible parallelization is determined as the minimum among:

- $\lfloor$*available power during* [$\tau$, $\tau+t_{test}(t_{ij})$]/ $p_{test}(t_{ij})\rfloor$,
- $par_{max}(c_i)$, and
- *available bandwidth during* [$\tau$, $\tau+t_{test}(t_{ij})$].

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

The computational complexity of the algorithm, comes from sorting and two loops. The sorting can be performed using a sorting algorithm at $O(|T| \times log\ |T|)$. The worst case for the loops occurs when only one test is scheduled in each iteration resulting in a complexity given by:

$$\sum_{i=0}^{|T-1|} (T-i) = \frac{|T|^2}{2} + \frac{|T|}{2}$$

where $|T|$ is the number of tests in the system. The total worst case execution time is $|T| \times log\ |T| + |T|^2/2 + |T|/2$ which is of $O(|T|^2)$. For instance, the shortest-task-first approach by Chakrabarty has a worst case complexity of $O(|T|^3)$ [1].

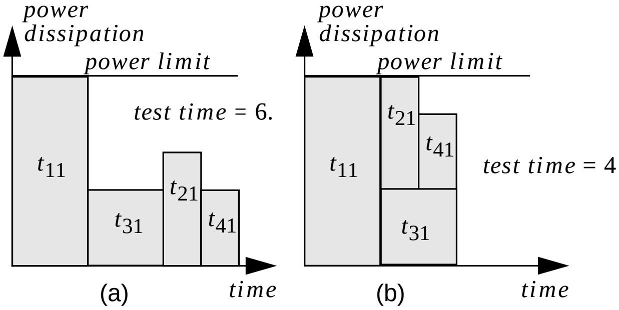

**Figure 10. The test schedule of the example design using test parallelization (a) and combined test parallelization and test scheduling (b).**

## 5 Experimental Results

We have performed experiments on a design example and an industrial design. For the design example (Figure 3) with resource graph in Figure 1 and the TCG in Figure 2 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 6(b)) and when only test parallelization is used the test time is also 6 time units (Figure 10(a)). However, when combining test scheduling test parallelization the test time is reduced to 4 time units (Figure 10(b)).

The industrial design has characteristics given in Table 1 and the power limitation is 1200 mW and only one test may use the test bus or the functional pins (fp) at a time. Furthermore block-level tests may not be scheduled concurrently with top-level tests. The minimal and maximal degree of parallelization is also given for each test.

A designers solution requires a test time of 1592 where the tests are scheduled in the following sequence: A, B, C, E, F, I, J, K, L, M, N, O, P, Q. Using the test scheduling approach we proposed [6] results in a test schedule as: N, {A || B, I, E, F, C, J, M}, P, O, Q, L, K where A is scheduled concurrent with B, I, E, F, C, J, M. The test time is 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, Table 2.

## 6 Conclusions

In this paper, we have proposed an integrated technique for test scheduling and scan-chain division under power constraints for the testing of SOCs. We have investigated scan-chain division 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 finding such solutions. We have also outlined the wrapper design allowing the core to be tested by several test sets at a variable length of the scan-chain. For such wrapper design, we have made a worst case analysis, which motivates that scan-chain division must be integrated into the test scheduling process. We have performed experiments on an industrial design to show the efficiency of the proposed technique.

| Test | Block | Test | Test Time | Test power | Test Port | Min Par. | Max Par. |
|---|---|---|---|---|---|---|---|
| Block-level tests | A | Test A | 515 | 379 | scan | 1 | 8 |
| | B | Test B | 160 | 205 | testbus | 1 | 8 |
| | C | Test C | 110 | 23 | testbus | 1 | 8 |
| | E | Test E | 61 | 57 | testbus | 1 | 8 |
| | F | Test F | 38 | 27 | testbus | 1 | 8 |
| | I | Test I | 29 | 120 | testbus | 1 | 8 |
| | J | Test J | 6 | 13 | testbus | 1 | 1 |
| | K | Test K | 3 | 9 | testbus | 1 | 1 |
| | L | Test L | 3 | 9 | testbus | 1 | 1 |
| | M | Test M | 218 | 5 | testbus | 1 | 8 |
| Top-level tests | A | Test N | 232 | 379 | fp | 1 | 8 |
| | N | Test O | 41 | 50 | fp | 1 | 8 |
| | B | Test P | 72 | 205 | fp | 1 | 8 |
| | D | Test Q | 104 | 39 | fp | 1 | 8 |

**Table 1. Characteristics of the industrial design.**

| Approach | Test time | Improvement |
|---|---|---|
| Designer | 1592 | - |
| Test scheduling | 1077 | 32% |
| Test parallelization | 383 | 76% |

**Table 2. Results on the industrial design.**

## References

[1] K. Chakrabarty, Test Scheduling for Core-Based Systems Using Mixed-Integer Linear Programming, *Trans. CAD of IC and Syst*, 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, K. Saluja, and V. Agrawal, 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, *Proc. of Int. Test Conference*, pp. 882-891, Oct. 2000.

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

[6] E. Larsson and Z. Peng, An Integrated System-On-Chip Test Framework, *Proc. of Design, Automation and Test in Europe*, pp 138-144, Munchen, Germany, March, 2001.

[7] N. Weste, K. Eshraghian, Principles of CMOS VLSI Design, *Addison-Wesley*, ISBN 0-201-53376-6, 1993.

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

[9] E. J. Marinissen et al., A Structured And Scalable Mechanism for Test Access to Embedded Reusable Cores, *Proc. of ITC.*, pp 284-293, Washington, DC, Oct. 1998.

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

[11] E. J. Marinissen, S. K. Goel, and M. Lousberg, Wrapper Design for Embedded Core Test, *Proc. of International Test Conference*, pp. 911-920, Washington, DC, Oct., 2000.