

A Reconfigurable Power Conscious Core Wrapper and its Application to System-on-Chip Test Scheduling

Erik Larsson · Zebo Peng

Received: 18 July 2006 / Accepted: 15 April 2008 / Published online: 27 June 2008
© Springer Science + Business Media, LLC 2008

Abstract The increasing test application times required for testing system-on-chips (SOCs) is a problem that leads to higher costs. For modular core based SOC's it is possible to employ a concurrent test scheme in order to lower the test application times. To allow each core to be tested as a separate unit, a wrapper is inserted for each core, the scan chains at each core are configured into a fixed number of wrapper chains, and the wrapper chains are connected to the test access mechanism. A problem with concurrent testing is that it leads to higher power consumption as several cores are active at a time. Power consumption above the specified limit of a core or above the limit of the system will cause damage and must be avoided. The power consumption must be controlled both at core level as well as on system level. In this paper, we propose a reconfigurable power conscious core wrapper that we include in a preemptive power constrained test scheduling algorithm. The advantages with the wrapper are that the number of wrapper chains at each core can dynamically be changed during test application and the possibility, through clock gating, to select the appropriate test power consumption for each core. The scheduling technique produces optimal solutions in respect to test time and selects wrapper configurations in a systematic manner while ensuring the

power limits at core level and system level are not violated. The wrapper configurations are selected such that the number of wrapper configurations as well as the number of wrapper chains at each wrapper are minimized, which minimizes the wrapper logic as well as the total TAM routing. We have implemented the technique and the experimental results show the efficiency of our approach.

Keywords Test scheduling · Power constraint · Core wrapper · Preemption

1 Introduction

The rapid development in semiconductor technology makes it possible to fabricate Integrated Circuits (ICs) that include a complete system. In order to design these ICs, which often are referred to as system chips or Systems-on-a-Chip (SoCs), in a timely manner, a modular design approach, where pre-designed and pre-verified blocks of logic, so called cores, is frequently used.

Due to imperfections at manufacturing, all ICs must be tested. As ICs are becoming increasingly complex, the test application times increase. For modular designed SOC's, concurrent testing is an attractive alternative to lower the test application times. In order to apply concurrent testing, each core must be designed such that it can be tested as an individual unit. This is usually achieved by inserting wrappers to each core. The wrapper acts as the interface between the scan chains at a core and the infrastructure for transporting test data in the system; Test Access Mechanism (TAM). The scan chains at each core are formed into wrapper chains which through the wrapper are connected to the TAM wires.

Responsible Editor: K. K. Saluja

The research is supported by the Swedish Foundation for Strategic Research (SFS) under the Strategic Integrated Electronic Systems Research (STRINGENT) program.

E. Larsson (✉) · Z. Peng
Embedded Systems Laboratory, Department of Computer Science,
Linköping University, Linköping, Sweden
e-mail: erila@ida.liu.se, zebpe@ida.liu.se

Concurrent testing leads to higher activity and consequently higher power consumption in the system. High power consumption can damage the system under test. At a global system level perspective, the total power consumed at any time must be kept under a given limit. At a local core level perspective, extensive power consumption can lead to a local hot spot. It is therefore important that at any time the total power consumption for the system is kept under the system's power budget and that the power consumption at an individual core at any time is kept under respective core's power budget.

In this paper [10], we propose a reconfigurable power conscious core wrapper, which we include in a preemptive test scheduling algorithm. The core wrapper can be used to regulate the test power at core level while the scheduling approach ensures that the system level power limit is not violated. We formulate a power condition that, if satisfied, guarantees that the preemptive test scheduling scheme produces optimal test application time for the system. The core wrapper combines the gated sub-chain scheme presented by Saxena et al. [11] and the reconfigurable core test wrapper introduced by Koranne [8], while the test scheduling technique is based on the approach proposed by Larsson and Fujiwara [9].

The main advantages with the proposed wrapper and combining it into a scheduling algorithm are that:

- a power constrained test schedule is produced in linear time,
- reconfigurable wrappers are for each core selected and inserted in a systematic manner that minimizes
- the number of wrapper chains at each core, which maximizes the possibility for clock gating, and minimizes the required number of TAM wires; hence cost for TAM routing is implicitly minimized.
- the number of wrapper configurations, which minimizes the added logic,
- an upper bound on the added wrapper logic is defined,
- it is possible to control the power consumption at each individual core, which allows the test clock speed to increase, and
- it is possible to control the test power consumption at system level, which should be kept below a given value in order to reduce the risk of over heating which might damage the system under test.

The rest of the paper is organized as follows. Related work is reviewed in Related Work and our reconfigurable power conscious test wrapper is introduced in A Reconfigurable Power Conscious Wrapper. In Power Constrained Test Scheduling we show how to include the wrapper in a preemptive test scheduling technique such that power consumption also is considered. In the experiments we have made a comparison with previous approaches and we

illustrate the advantages with our wrapper and its use in the proposed scheduling approach in Experimental Results. The paper is concluded with conclusions in Conclusions.

2 Related Work

The problem with high test power consumption and long test application times can be tackled by:

- *design for low power testing*—the system is designed to minimize test power consumption, which allows consequently testing at a higher clock frequency to lower test times [11], and
- *power constrained test scheduling*—the tests are organized in such a way that the test time is minimized while considering test power limitations [3–6].

Saxena et al. [11] proposed scan chains gating, a design for low power test approach, to address power consumption at core level. Gating scan chains makes it possible to increase the number of cores that are tested concurrently alternatively test a given core at a higher frequency. Power consumption can be controlled at core level to avoid local hot spots, however, Saxena et al. [11] do not include a test scheduling algorithm to select which cores in the system to test concurrently.

Chou et al. proposed a power constrained test scheduling technique where each testable unit has one test with a fixed test time and a fixed power consumption value. The objective is to organize the tests such that the total test application time is minimized while considering test conflicts and not violating test power limitation [3]. Iyengar and Chakrabarty [5] defined a preemptive power constrained test scheduling technique. The idea with preemption is that each test can be partitioned into parts and applied as separate units. The advantage is that it can ease the scheduling by avoiding conflicts. For stuck-at test, preemption can be done at any time as only a single capture is used. For dealy test preemption cannot be applied between the initialization and capture.

Several test scheduling techniques that address long test application time but not power consumption have been proposed [1, 4, 7]. The general idea is to group the scan chains at each core into a fixed number of wrapper chains and connect the wrapper chains to the TAM. Iyengar et al. [6] and Huang et al. [4] contributed by proposing power constrained test scheduling techniques. Similar to the

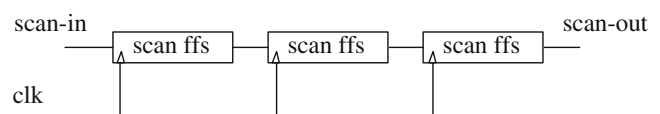


Fig. 1 Original scan chain [11]

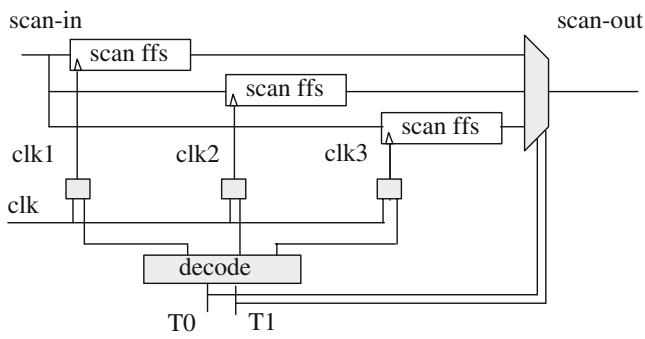


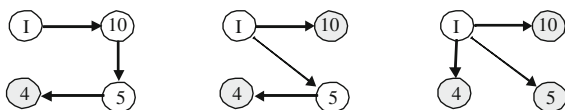
Fig. 2 Scan chain with gated sub-chains [11]

approaches by Chou et al. [3] and Iyengar and Chakrabarty [5], the techniques by Iyengar et al. [6] and Huang et al. [4] focus on system power limit only; hence local hot spots are not considered.

Koranne [8] introduced a reconfigurable wrapper with the advantage of allowing N_{tam} wrapper chain configurations per wrapped core. In order to minimize the overhead due to the reconfigurable wrapper, a limited number of cores are selected prior to the scheduling to have a reconfigurable wrapper. Larsson and Fujiwara [9] proposed a preemptive scheduling technique where the reconfigurable core wrappers are systematically selected during the scheduling process. The approaches by Koranne [8] and Larsson and Fujiwara [9] do not address test power consumption.

3 A Reconfigurable Power Conscious Wrapper

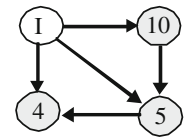
We propose a reconfigurable power conscious (RPC) wrapper that combines the gated sub-chain approach proposed by Saxena et al. [11] and the reconfigurable wrapper introduced by Koranne [8]. The basic idea in the approach proposed by Saxena et al. [11] is to use a gating scheme to lower the test power dissipation during the shift process. For example, given a set of three scan chains connected into a single chain as in Fig. 1. During the shift process, all flip-flops in all scan chains are active. It leads to high switch activity and therefore high power consumption. However, if the scan chains are gated (Fig. 2), only one of the three chains is active at a time during the shift process. The switch activity is reduced and also in the clock tree distribution while the test time remains the same in the two cases [11].



(a) 1 wrapper-chain (b) 2 wrapper-chains (c) 3 wrapper-chains

Fig. 3 Di-graph representations

Fig. 4 The union of di-graphs in Fig. 3



The wrapper proposed by Koranne allows, in contrast to standard wrappers, several wrapper chain configurations [8]. The configurations can be changed during test application. The main advantage is increased flexibility in the scheduling process. We use a core with three scan chains of length {10, 5, 4} to illustrate the approach. The scan chains and their partitioning into wrapper chains are specified in. Scan chain partitions.

For each TAM widths (1, 2, and 3) a di-graph (directed graph) is generated where a node denotes a scan chain and the input TAM, node I (Fig. 3). An arc is added between two nodes to indicate that the two are connected. The shaded nodes are to be connected to the output TAM. A combined di-graph is generated as the union of the di-graphs. Figure 4 shows the result of the generated combined di-graph from the three di-graphs in Fig. 3. The indegree at each node (scan chain) in the combined di-graph gives the number of signals to multiplex. For instance, the scan chain of length five has two input arcs, which means that a multiplexer selecting between an input signal and the output of the scan chain of length ten is needed. The multiplexing for the example is outlined in Fig. 5.

Our approach works in two steps. First, we generate the reconfigurable wrapper using Koranne’s approach. Second, we add clock gating, which means we connect the *inputs* of each scan chain to the multiplexers, which is to be compared to connecting the *outputs* of each scan chain as in the approach by Koranne. We illustrate our approach using the scan chains specified in. Table 1. The result is given in Fig. 6, and the generated control signals are in Table 2.

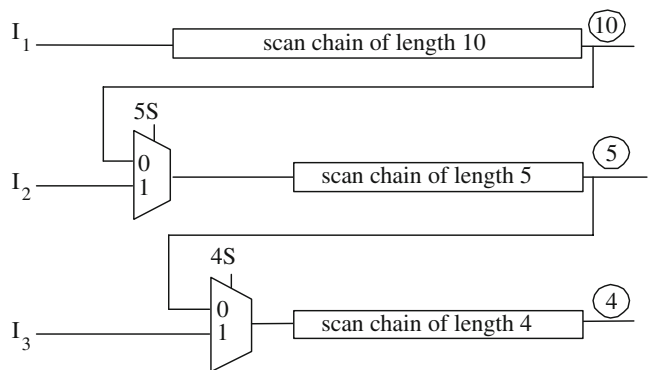


Fig. 5 Multiplexing strategy [8]

Table 1 Scan chain partitions

TAM width	Wrapper chain partitions	Max length
1	[(10,5,4)]	19
2	[(10),(5,4)]	10
3	[(10),(5),(4)]	10

The advantages are that we gain control of the test power consumption at each core, and we do not require the extra routing needed with Koranne’s approach, as illustrated in Fig. 7.

4 Power Constrained Test Scheduling

In this section we describe how the test scheduling proposed by Larsson and Fujiwara [9] is extended to also include the proposed wrapper in order to handle power constraints.

4.1 Test Scheduling

We could make use of the RPC wrapper at all cores, which would lead to a high flexibility since we could reconfigure the wrapper into any configuration. However, in order to minimize the overhead, we will use a systematic approach to select cores and number of configurations at each core.

Larsson and Fujiwara [9] showed that the test scheduling problem of core tests is equal to the independent job scheduling on identical machines since each test t_i at core c_i , ($i=1, 2, \dots, n$) with testing time τ_i is independent on all other core tests and each TAM wire w_j ($j=1, 2, \dots, N_{tam}$) corresponds to an independent machine used to transport test data. The testing time τ_i is the test time when all scanned elements at a core are connected into a single chain (a single wrapper chain). The lower bound (LB) of the test time for a given TAM width N_{tam} can be computed by [2]:

$$LB = \max \left\{ \max(\tau_i), \sum_{i=1}^n \tau_i / N_{tam} \right\} \tag{1}$$

Larsson and Fujiwara [9] also showed that the problem of independent job scheduling on identical machines can be solved in linear time ($O(n)$ for n tests) by using preemption

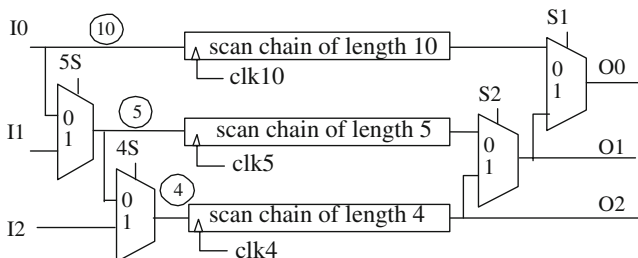


Fig. 6 Our multiplexing and clocking strategy

Table 2 Control signals

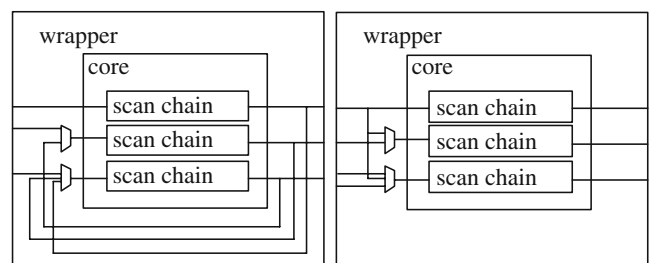
Wrapper chains	T0 T1 T2	5S 4S	S1 S2	clk10 clk5 clk4
3	0 0 0	1 1	0 0	1 1 1
2	0 0 1	1 x	0 0	1 0 0
	0 1 0	1 0	0 1	0 1 1
1	0 1 1	x x	0 x	1 0 0
	1 0 0	0 x	1 0	0 1 0
	1 0 1	0 0	1 1	0 0 1

[2]: assign tests to the TAM wires successively, assign the tests in any order and preempt tests into two parts whenever the LB is reached. Assign the second part of the preempted test on the next TAM wire starting from time point zero. The preemption can be done at any clock cycle in the case of testing for stuck-at faults. In the case of delay testing, preemption cannot be allowed to take place between the initialization and the capture cycle.

An example (Fig. 8) illustrates the approach where the five cores and their test times are given. The LB is computed to 7 (Eq. 1) and due to that $\tau_i \leq LB$ for all tests, the two parts of any preempted test will not overlap. The scheduling proceeds as follows: The tests are considered one by one, for instance, starting with a test at c_1 which is scheduled at time point 0 on wire w_1 . At time point 4, when the test at c_1 is finished, the next test, for example, test at c_2 is scheduled to start. At time point 7 when LB is reached, the test at c_2 is preempted and the rest of the test is scheduled to start at time 0 on wire w_2 . The test for c_2 is partitioned into two parts.

A long test time for one of the cores in the system may limit the solution, *i.e.* LB is given by the test time of a test ($\max(\tau_i)$ in Eq. 1). In such a case, the test time can be reduced by assigning more TAM wires to that particular core so that the length of the wrapper chains becomes shorter. The LB equation does not require the $\max(\tau_i)$ part (Eq. 1) and becomes:

$$LB = \max \sum_{i=1}^n \tau_i / N_{tam}. \tag{2}$$



(a) Koranne’s routing [8]. (b) Our approach.

Fig. 7 Wrapper routing

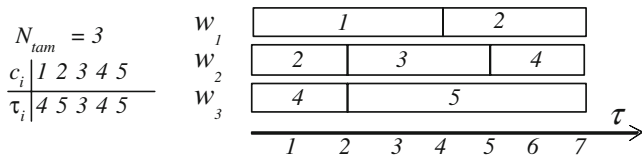


Fig. 8 Optimal TAM assignment and preemptive scheduling

After LB is computed, the scheduling approach described above is used (Fig. 8). For illustration, we use the same example but with a wider TAM ($N_{tam}=7$). The final test schedule is in Fig. 9. A test may now overlap in using the wires (machines). For instance, the test at c_1 uses wire w_1 and w_2 during time period 0 to 1 and only wire w_1 during period 1 to 3. A reconfigurable wrapper is required to handle this [9].

After assigning TAM wires to all cores, the wrapper chains for each core are determined, which is illustrated in Fig. 9. For instance, in partition 1 of the test at c_2 , w_3 is used during period τ_{21} and in partition 2 of the test at c_2 , w_2 and w_3 are used during period τ_{22} . From this we determine that two wrapper chains are initially needed and then a single wrapper chain is needed. In total, two configurations are needed for core c_2 .

The generic partitioning of a test’s usage of wires over the testing time is given in Fig. 10. For each test, a start time $start_i$ and an end end_i are assigned by the algorithm, respectively. The number of partitions, which will be the number of configurations, is computed for each test by the algorithm given in Fig. 11. If the test time τ_i for a test t_i is below LB, only one configuration is needed. A multiplexer might be required for wire selection if $start_i > end_i$. From the algorithm, we find that the maximal number of partitions per test is three, which means we in the worst case have to use three configurations per core. The wrapper logic is then in range $|C| \times 3 \times \text{technology parameter}$ (maximum three configurations per core).

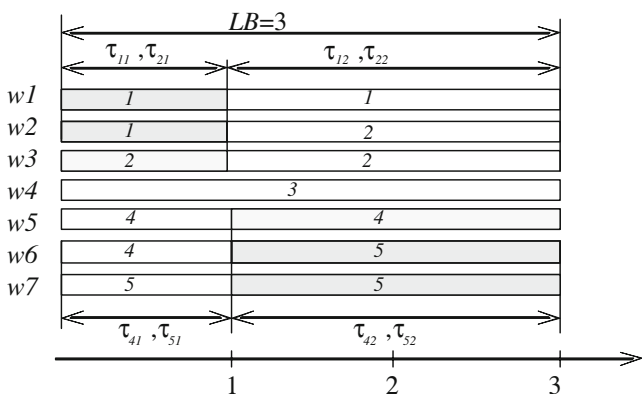


Fig. 9 Partitioning of the schedule in Fig. 9

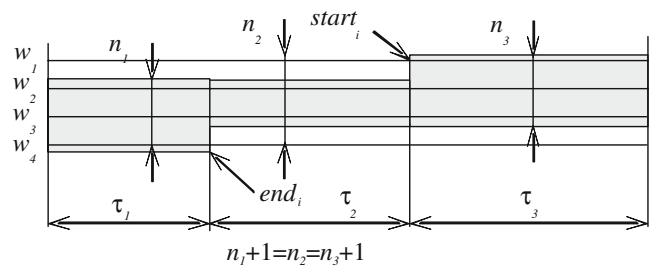


Fig. 10 Bandwidth requirement for a general test

4.2 Power Constrained Test Scheduling

We use an example to illustrate the test power modelling at core level (Fig. 12). In Fig. 12a a single wire is assigned to the core; hence the three scan chains form a single wrapper chain. The result is that the wire usage is minimized but both the test time and the test power are relatively high. In Fig. 12b three TAM wires (one per wrapper chain) are used resulting in a lower test time while the test power consumption remain the same as in Fig. 12a. In our approach which uses scan gating (Fig. 12c) results in the same test time as in Fig. 12a but at a lower test power consumption. The reduction in test power is due to that each scan chain is loaded in a sequence, and not more than one scan chain is activated at a time.

And as our test scheduling technique minimizes the number of TAM wires at each core by assigning as few wires as possible to each core, the result is that each wrapper chain includes a high number of scanned elements. This is an advantage since it maximizes the possibility to

```

for all  $t_i$  begin
  if  $\tau_i \leq LB$  then begin
    if  $start_i \leq end_i$  then begin
      no pre-emption for  $t_i \rightarrow$  no wrapper logic is added.
    end else begin
      the test is split into two parts at different wires  $\rightarrow$ 
      1 multiplexer is inserted.
    end
  end else begin
    if  $start_i \leq end_i$  then begin
      one configuration needed from time point 0 to  $start_i$ ,
      one configuration needed from time point  $start_i$  to  $end_i$ ,
      one configuration needed from time point  $end_i$  to LB,
    end else begin
      one configuration needed from time point 0 to  $end_i$ ,
      one configuration needed from time point  $end_i$  to  $start_i$ ,
      one configuration needed from time point  $start_i$  to LB,
    end
  end
end
end
    
```

Fig. 11 Algorithm to determine wrapper logic

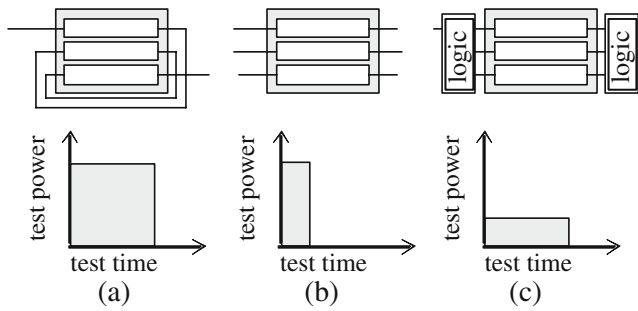


Fig. 12 Core design alternatives

gate scan chains at each wrapper chain. In other words, we have a high number of scan chains at each wrapper chain and that means we can have a high number of gated scan chains, and hence high control of the test power consumption at each core.

We use the power model based on the results by Saxena et al., which means the power depends on the number and the length of the wrapper chain partitions. However, a more elaborate power model can easily be adopted in our approach. We assume that the test power at a core is evenly distributed over the scanned elements. The algorithm to compute the power limit (P_{limit}) for a system is in Fig. 13. At step 2, the LB is computed, and at step 3, the maximal number of required TAM wires are computed. At step 4, the amount of test power consumed by each scan chain, and wrapper cell is computed. At steps 5 and 6, the N_{tam} values with highest test power are summarized which is the P_{limit} . If P_{limit} is below P_{max} ($P_{limit} \leq P_{max}$), optimal test time can be achieved.

1. Given: a system with i cores, where each core c_i consists of ff_i scanned flip-flops and wrapper cells partitioned into j scan chains each of length sc_{ij} (including wrapper cells). The test time τ_i is computed as if all ff_i elements are connected to a single wrapper chain. The test power when all ff_i elements are active is given by p_i . N_{tam} is the TAM bandwidth.
2. Compute LB (lower bound)
3. For each core i compute w_i as the maximal number of TAM wires that can be assigned to it assuming preemptive scheduling:

$$w_i = \left\lceil \frac{\tau_i}{LB} \right\rceil$$

4. For each scan chain s_{ij} compute its power:

$$P_{ij} = \frac{P_i}{ff_i} \cdot sc_{ij}$$

5. For all cores: select the w_i scan elements with highest power value and sort them descending in a list L .
6. For all scan elements in L select the N_{tam} first and the P_{limit} is equal to the summation of the N_{tam} values.

Fig. 13 Algorithm to compute the power limit

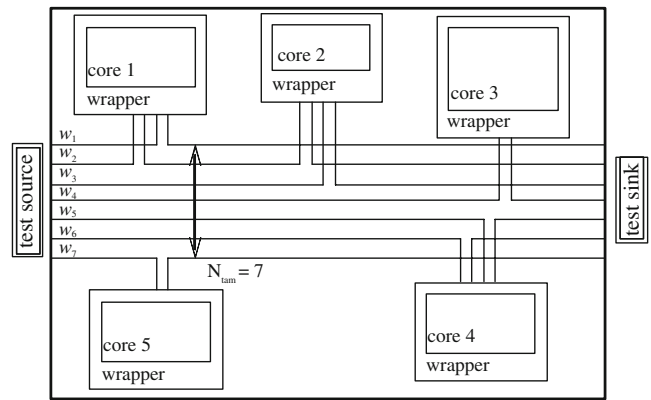


Fig. 14 The example system assuming the five wrapped cores to be floor-planned

We have now a relationship between the TAM bandwidth and the test power. We can determine that the TAM bandwidth; N_{tam} can be increased as long as $P_{limit} \leq P_{max}$. It is also possible to increase the frequency of the test clock in order to minimize the test time as long as $P_{limit} \leq P_{max}$.

4.3 TAM Wiring Minimization

The test scheduling approach above minimizes the number of TAM wires assigned to each core. The advantage is that even if the floor-plan for the cores is unknown the TAM routing cost is minimized as a minimal number of TAM wires are assigned to each core. If the floor-plan is known, we can further minimize the TAM routing since the scheduling approach above does not require any particular sorting of the tests. We take the system in Fig. 8 with $N_{tam} = 7$ resulting in a test schedule as in Fig. 9 where the cores are sorted (and numbered) clock-wise as in Fig. 14. The advantage is that neighbouring cores share TAM wires. For instance core 2, which makes use of TAM wire w_2 as soon as core 1 finish its use of w_2 . Cores placed far away from each other are not sharing TAM wires, such as core 5 and core 3.

Table 3 Test power consumption options at core 12 (P93791) with the RPC wrapper at a fixed test time (1813502) and fixed TAM bandwidth (1)

Wrapper chains	Power consumption	Wrapper logic
1	4,634	0
2	2,317	1
3	1,545	3
4	1,159	4
5	927	5
6	773	6

Table 4 Test time on P93791 for the multiplexing architecture [1], the distribution architecture [1], and our approach at different power limitations

P_{max}	TAM width	Multiplexing architecture Test time	Distribution architecture Test time	Our approach Test time
100,000	4	7113317	Not applicable	6997584
	8	3625510	Not applicable	3498611
	16	1862427	Not applicable	1752336
	24	1262427	Not applicable	1174252
	32	1210398	5317007	877977
	40	1119393	1813502	703219
	48	660143	1126316	592214
	56	645698	907097	511925
50,000	64	645682	639989	442478
	4	7113317	Not applicable	6997584
	8	3625510	Not applicable	3498611
	16	1862427	Not applicable	1752336
	24	1262427	Not applicable	1174252
	32	1210398	Not applicable	877977
	40	1119393	Not applicable	703219
	48	660143	Not applicable	592214
20,000	56	645698	Not applicable	511925
	64	645682	Not applicable	442478
	4	Not applicable	Not applicable	6997584
	8	Not applicable	Not applicable	3498611
	16	Not applicable	Not applicable	1752336
	24	Not applicable	Not applicable	1174252
	32	Not applicable	Not applicable	877977
	40	Not applicable	Not applicable	703219
48	Not applicable	Not applicable	592214	
56	Not applicable	Not applicable	511925	
64	Not applicable	Not applicable	442478	

5 Experimental Results

We have made experiments using the ITC'02 design P93791. The design consists of 32 cores. Most cores are scan tested while a few do not have any scan chains.

First, we illustrate core level test power control with our wrapper using core 12 in design P93791. We assume a single TAM wire and show that the test time remains the

same while the test power consumption can be adjusted depending on the number of gated wrapper chains. The results are in Table 3.

Second, we compare our approach with the multiplexing [1] and the distribution architecture [1] to show the advantage of system level power control. We make experiments at three given system power constraints; 100,000, 50,000 and 20,000. In the multiplexing approach all cores

Table 5 Number of configurations per core for our scheduling approach on P93791

TAM bandwidth	Test time	Wrapper logic at core c_i												Cores with no scan chains	Total
		c_1	c_6	c_{11}	c_{12}	c_{13}	c_{14}	c_{17}	c_{19}	c_{20}	c_{23}	c_{27}	c_{29}		
4	6997584	0	1	0	0	0	1	0	0	1	0	0	0	0	3
8	3498611	0	3	0	1	0	1	0	1	1	0	1	0	1	9
16	1752336	1	3	0	1	1	1	1	1	3	1	1	1	1	16
24	1174252	2	3	0	3	3	3	3	1	3	3	3	1	1	29
32	877977	2	3	1	3	3	3	3	3	3	3	3	3	1+1	35
40	703219	2	3	0	3	3	3	3	3	3	3	3	3	3+1	36
48	592214	2	3	0	3	3	3	3	3	3	3	3	3	3+2	37
56	511925	2	3	0	3	3	3	3	3	3	3	3	3	3+2+3	40
64	442478	2	3	1	3	3	3	3	3	3	3	3	3	3+3+3	42

are tested in a sequence where the full bandwidth is given to each core at a time. In the distribution architecture, every core is given its dedicated TAM wires. The distribution architecture is sensitive to test power consumption since the testing of all cores is started at the same time. The results are collected in Table 4. The distribution architect is not applicable when the TAM bandwidth is below the number of cores (32 in P93791). At the 50,000 power limit, the distribution architecture cannot be used since activating all cores exceeds the power limit. At the limit 20,000, the multiplexing approach is not applicable since core 6 limits the solution with its consumption of 24,674. Our approach results in the same test time, however, the wrapper logic is increased in order to gate the wrapper chains. Note that we have defined an upper bound on the wrapper logic. It means we always have control on the added overhead. We have in Table 5 collected the overhead due to the use of our reconfigurable wrapper. The overhead is computed as follows. For cores with a single TAM bandwidth assigned to it, only one bandwidth is required and the cost is assumed to be zero. In some cases, only a multiplexer is to be added for the selection between wires and we assumed such cost to be equal to 1. For cores with three configurations, we assumed the cost to be equal to 3.

6 Conclusion

The test application times are increasing for Integrated Circuits. For modular System-on-Chip, the test application times can be reduced by concurrent execution of tests; however, it leads to higher power consumption. Test power consumption must be controlled at core level to avoid local hot spots as well as at system level. In this paper we propose a reconfigurable power conscious core test wrapper and described its application to SOC test scheduling. The advantages with our approach are that (1) the power constrained test schedule is produced in linear time, (2) the reconfigurable wrappers are selected and inserted in a systematic manner that (a) minimizes the number of wrapper chains at each core, which maximizes the possibility for clock-gating, and minimizes the required number of Test Access Mechanism wires; hence TAM routing is implicitly minimized and (b) minimizes the number of wrapper configurations, which minimizes the

added logic, (3) it is possible to control the power consumption at each individual core, which can be used to adjust and lower test time while avoiding local hot spots, (4) it is possible to control the test power consumption at system level, which should be kept below a given value in order to reduce the risk of over-heating which might damage the system under test, and (5) an upper bound on the added wrapper logic is defined. We have implemented the technique and made experiments that show the efficiency of the approach.

References

1. Aerts J, Marinissen EJ (1998) Scan chain design for test time reduction in core-based ICs. In: Proceedings of the IEE international test conference (ITC), pp 448–457, October
2. Brucker P (1998) Scheduling algorithms. Springer-Verlag, Berlin. ISBN 3-540-64105-X
3. Chou R, Saluja K, Agrawal V (1997) Scheduling tests for VLSI systems under power constraints. *IEE Trans VLSI Systems* 5(2): 175–185
4. Huang Y, Reddy SM, Cheng W-T, Reuter P, Mukherjee N, Tsai C-C, Samman O, Zaidan Y (2002) Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm. In: Proceedings of international test conference (ITC), Baltimore, Maryland, USA, pp 74–82, October
5. Iyengar V, Chakrabarty K (2001) Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip. In: Proceedings of VTS, Marina Del Rey, CA, USA, pp 368–374
6. Iyengar V, Chakrabarty K, Marinissen EJ (2002) Wrapper/TAM co-optimization, constraint-driven test scheduling, and test data volume reduction for SOCs. In: Proceedings of IEEE/ACM design automation conference (DAC) Louisiana, New Orleans, pp 685–690, June
7. Iyengar V, Chakrabarty K, Marinissen EJ (2002) Co-optimization of test wrapper and test access architecture for embedded cores. *J Electron Test* 18(2):213–230, April
8. Koranne S (2002) A novel reconfigurable wrapper for testing embedded core-based and its associated scheduling. *J Electron Test* 18(4):415–434, August
9. Larsson E, Fujiwara H (2006) System-on-chip test scheduling with reconfigurable core wrappers. *IEE Trans VLSI Systems* 14(3): 305–309, March
10. Larsson E, Peng Z (2003) A reconfigurable power-conscious core wrapper and its application to SOC test scheduling. In: Proceedings of the IEE international test conference (ITC), Charlotte, NC, USA, September–October 2003, (paper 44.2)
11. Saxena J, Butler KM, Whetsel L (2001) An analysis of power reduction techniques in scan testing. In: Proceedings of the IEE international test conference (ITC), pp 670–677, Baltimore, MD, October 2001