Power Conscious Test Synthesis and Scheduling for BIST RTL Data Paths

Nicola Nicolici and Bashir M. Al-Hashimi
Electronic Systems Design Group
Department of Electronics and Computer Science
University of Southampton, Southampton SO17 1BJ, UK
{nn99r,bmah}@ecs.soton.ac.uk

Abstract

Previous research has outlined that power dissipated during test application is substantially higher than during functional operation, which leads to loss of yield and decreases reliability. This paper shows for the first time how power is minimized in BIST RTL data paths by using power conscious test synthesis and test scheduling. According to the necessity for achieving the required test efficiency power dissipation is classified into necessary and useless power dissipation. According to the occurrence during the testing process power dissipation is classified into test application and shifting power dissipation. The effect of test synthesis and scheduling on power dissipation is analyzed and power minimization is achieved in two steps. Firstly, during the testable design space exploration only power conscious test synthesis moves are accepted leading to minimization of useless power dissipation. Secondly, module selection during power conscious test scheduling satisfies power constraints while reducing test application time. Experimental results using generic power models show savings up to 28% in test application power dissipation and up to 29% in shifting power dissipation.

1. Introduction

Power dissipation is one of the major design constraints for modern very large scale integration (VLSI) circuits due to the continuing increase in chip density. While many techniques have investigated power minimization during the normal (functional) mode of operation [1], an emerging research area is power minimization during test application [2]. Low power testing is important to increase reliability [3] and yield [4] of the circuit under test. Previous work has addressed power minimization during test application at logic and system level. At logic level for combinational circuits employing built-in self-test (BIST) [5] several techniques for minimizing power dissipation have been proposed recently [6–13]. In [6] the use of dual speed linear feedback shift register (LFSR) lowers the transition density at the circuit inputs leading to minimized power dissipation. Optimal weight sets for input signal distribution are determined in order to minimize average power [7], while the peak power is reduced by finding the best initial conditions in the cellular automata (CA) cells used for pattern generation [8]. It has been proved in [9] that all the primitive polynomial LFSR of the same size, produce the same power dissipation in the circuit under test, thus advising to use the LFSR with smaller number of XOR gates since it yields lowest power dissipation by itself. A mixed solution based on reseeding LFSRs and test vector inhibiting to filter non-detecting subsequences of a pseudorandom test sequence has been proposed in [10]. An enhancement of test vector inhibiting technique has been proposed in [11] where all the non-detecting subsequences are filtered. A different approach for filtering non-detecting vectors inspired by the precomputation architecture is presented in [12]. An improvement in area overhead associated with filtering non-detecting vectors without penalty in fault coverage or test length has been achieved using non-linear hybrid cellular automata [13]. Regardless of the type of test pattern generator, BIST architectures significantly differ one from another in terms of power dissipation as outlined in [14]. Thus, circuit partitioning for low power BIST and test session planning have an important influence on power dissipation as shown in [15]. Regularity of multiplier modules and linear sized test set required to achieve high fault coverage lead to efficient low power BIST implementations [16].

To minimize power dissipation in scan-BIST sequential circuits during test application numerous techniques have been proposed recently [17–20]. To minimize shifting power dissipation, test vector inhibiting techniques proposed for combinational circuits are extended to scan sequential circuits [17]. In [18] the test vector inhibiting technique is extended where the modules and modes with the highest power dissipation are identified, and gating logic has been introduced to reduce power dissipation. Despite substantial savings in power dissipation vector detection and gating logic introduce not only area overhead but also

Proceedings of the International Test Conference (ITC'00)
1089-3539/00 $17.00 © 2003 IEEE

performance degradation for modified scan cell design. In the low transition random test pattern generator (LT-RTPG) proposed in [19], neighbouring bits of the test vectors are assigned identical values in most test vectors. A simple and fast procedure to compact scan vectors as much as possible without exceeding power dissipation has been proposed in [20]. It should be noted that techniques proposed for reducing power dissipation in standard scan design for test (DFT) methodology [21–25] can equally be applied to scan-BIST environment subject to minor modifications. While techniques for power minimization at logic level yield modest savings they can be combined with techniques proposed at higher levels of abstraction to produce further savings in power dissipation. Furthermore, new techniques for power minimization at higher levels of abstraction are required when applying BIST [26, 27] for register-transfer level (RTL) data paths synthesized using high level synthesis for low power [28].

To overcome the problem of high power dissipation during test application during BIST at higher levels of abstraction, numerous power-constrained test scheduling algorithms have been proposed [3, 29–33]. The approach in [3] schedules the tests under power constraints by grouping and ordering based on floorplan information. In [29] a power-constrained test scheduling algorithm at the system level has been presented for high performance memories and multichip modules. A resource graph formulation for the test problem is given and tests are scheduled concurrently without exceeding their power ratings during test application. To overcome the identification of all the cliques in a graph and the covering table minimization problem applied in [29], which are well known NP-hard problems, the solution proposed in [30] uses the left edge algorithm and tree growing technique as an heuristic for the block test scheduling problem. Several new solutions for scheduling tests under power constraints [31–33] have been proposed recently. Scheduling tests for complex digital systems based on test application time and power dissipation estimates has been proposed in [31]. In [32] a polynomial-time algorithm for power constrained testing of core based systems has been proposed. The similarity of the cores being tested and their layout-relation are addressed in order to minimize test application time by merging two test sessions. Test access architectures for systems-on-a-chip based on integer linear programming that incorporate system level constraints on power dissipation have been introduced in [33]. Despite the efficiency achieved at the system level, the test scheduling algorithms [3, 29–33] cannot be applied for BIST RTL data paths due to the following two reasons. Firstly, all the previously outlined test scheduling algorithms are formulated for a fixed test resource allocation. Since test synthesis and test scheduling are strictly interrelated [26, 27] the fixed test resource formulation will lead to prohibitively large computational time hindering efficient exploration of the testable design space. Secondly, previously proposed power-constrained test scheduling algorithms assume fixed amount of power dissipation associated with each test. This assumption is not valid in the case of BIST RTL data paths where transitions associated with necessary power dissipation required for testing each module can propagate to other registers and further to untested modules leading to useless power dissipation (described in section 2) which does not have any influence on test efficiency. Thus, new techniques which take into account the interrelation between test synthesis and test scheduling for minimization of necessary power dissipation and elimination of useless power dissipation are required.

The aim of this paper is to introduce a new power conscious technique for test synthesis and scheduling which takes into account the complexity of the testable design space caused by the interrelation between test synthesis and test scheduling leading to power-minimized testable data paths in low computational time. Furthermore, by considering both necessary and useless power dissipation in BIST RTL data paths the proposed technique generates high savings in power dissipation during test application and while shifting out of test responses. The rest of the paper is organized as follows. Section 2 accounts for sources of power dissipation in BIST RTL data paths, and the effect of test synthesis and scheduling is investigated in section 3. Power constrained testable design space exploration is described in section 4, and experimental results and conclusions are given in sections 5 and 6 respectively.

2. Power Dissipation Classification During Test Application in BIST RTL Data Paths

This section gives a taxonomy of power dissipation during test application in BIST RTL data paths. According to the necessity for achieving the required test efficiency power dissipation may be classified into two components. *Necessary power dissipation* is the power dissipated in test registers and tested modules during each test session. *Useless power dissipation* is the power dissipated in registers and untested modules due to spurious transitions which cannot be eliminated by any configuration of control signals of data path multiplexers. Since the basic idea beyond the proposed technique is to choose control signals so whatever driven by the multiplexers has the constant data inputs, the multiplexer itself does not have constant data inputs. The power dissipated by multiplexers is not targeted by the proposed technique. However, while during the normal operation the multiplexer power is very large [34] it is very low during test application. This is due to the fact that multiplexer control signals are modified only at the start and end of each test session thus avoiding any glitching activity which can propagate from control logic. The following example clarifies the previously outlined classification.

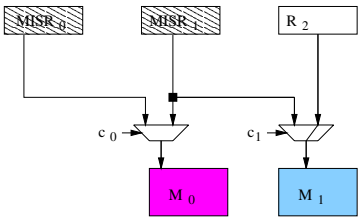


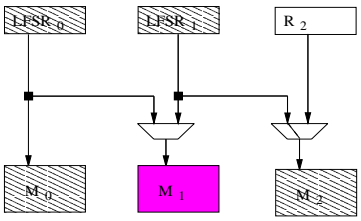
Figure 1. Useless power dissipation during test application in BIST RTL data paths.

Example 1 Consider the circuit shown in Fig. 1 where multiple-input signature analysis registers $MISR_0$ and $MISR_1$ are active. For the sake of simplicity the modules under test do not appear in the Fig. 1. The output of $MISR_0$ is connected to M_0 while $MISR_1$ is connected to M_0 and M_1. In the case of M_1 the inactive register R_2 can be selected by using the appropriate value of control signal c_1 which stops the propagation of transitions that occur in $MISR_1$. However, in the case of M_0 there is no value for control signal c_0 such that transitions which occur in $MISR_0$ or $MISR_1$ are eliminated at the input of M_0. Thus spurious transitions occur in M_0 without any influence on test efficiency leading to useless power dissipation.

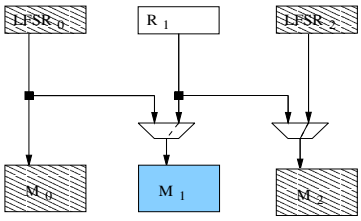
According to the occurrence during the testing process power dissipation is classified into: *Test Application Power dissipation* (TAP) and *Shifting Power dissipation* (SP). TAP is the power which occurs during test application of test vectors required to achieve maximum fault coverage. On the other hand SP is the power which occurs while shifting in the seeds for LFSRs required for next test session and shifting out the signatures stored in MISRs. Note that both TAP and SP comprise necessary and useless power dissipation. While the necessary power dissipation is compulsory for achieving the required test efficiency, the useless power dissipation may be eliminated and it is targeted by power conscious test synthesis and scheduling as described in section 4.

3. Effect of Test Synthesis and Scheduling on Useless Power Dissipation

In order to eliminate useless power dissipation the effect of test synthesis and scheduling on useless power dissipation is analyzed through two detailed examples. The first example analyzes the effect of test synthesis on both test application and shifting power. The second example examines the effect of module selection during test scheduling on elimination of useless power dissipation in both registers and modules.



(a) M_1 dissipates useless power



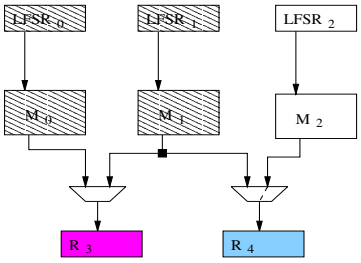
(b) M_1 does not dissipate power

Figure 2. Test synthesis for useless power elimination.

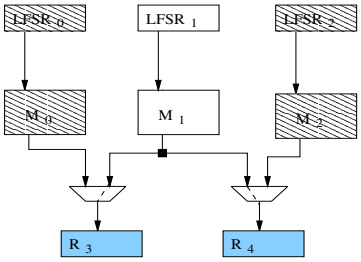
Example 2 Consider the circuit shown in Fig. 2 and assume that modules M_0 and M_2 are tested simultaneously without exceeding the power constraints. When linear feedback shift register $LFSR_1$ generates test patterns for M_2 any configuration of control signals for multiplexer at the input of M_1 will lead to useless power dissipation in M_1 (Fig. 2(a)). However when $LFSR_2$ generates test patterns for M_2, by selecting the inactive register R_1 at the input of M_1 will lead to the elimination of useless power in M_1 without any penalty in test area or test efficiency (Fig. 2(b)). It should be noted that test synthesis has a profound impact also on shifting power (SP in section 2) since an inappropriate selection of test registers may lead to useless power dissipation in modules while shifting out test responses.

Having described the effect of test synthesis on useless power dissipation, the following example examines the effect of module selection during test scheduling on useless power elimination in both registers and modules.

Example 3 Consider the circuit shown in Fig. 3. Assume that module M_0 is already scheduled in the current test session and the selection of M_1 and M_2 is examined. It should be noted that for the sake of simplicity signature analysis registers for M_0, M_1, and M_2 are not shown in Fig. 3 and registers R_3 and R_4 are not used as analyzers in the current test session. By selecting M_1 to be tested simultaneously

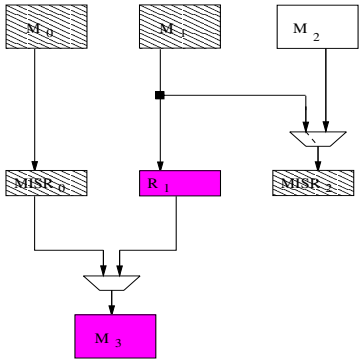


(a) R_3 dissipates useless power

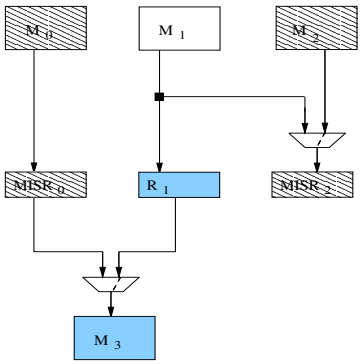


(b) R_3 does not dissipate power

Figure 3. Module selection during test scheduling for useless power elimination in registers.



(a) M_3 dissipates useless power



(b) M_3 does not dissipate power

Figure 4. Module selection during test scheduling for useless power elimination in modules.

with M_0, any configuration of control signals for multiplexer at the input of R_3 will lead to useless power dissipation in R_3 (Fig. 3(a)). The useless power in both R_3 and R_4 is eliminated by selecting M_2 to be tested simultaneously with M_0 and setting the appropriate values on control signals of multiplexers at the inputs of R_3 and R_4 (Fig. 3(b)). So far the effect of test scheduling on useless power in registers has been outlined. To examine the effect of test scheduling on useless power in modules consider the circuit shown in Fig. 4. Assume that $MISR_0$ analyzes test responses from M_0 and $MISR_2$ analyzes test responses from either M_1 or M_2. Scheduling the test for M_1 at the same time with the test for M_0 will lead not only to useless power dissipation in R_1 but also in M_3. This is because both $MISR_0$ and R_1 are active at the same time which leads to propagation of spurious transitions from R_1 to M_3 (Fig. 4(a)). The useless power is eliminated in both R_1 and M_3 by selecting M_2 to be tested simultaneously with M_0 and setting the appropriate values on control signals of multiplexers at the input of M_3 (Fig. 4(b)). This clearly shows that test scheduling has effect not only on useless power in registers, but also on useless power in modules.

It should be noted that for designs where gated clocks are employed at register transfer level, useless power dissipation in registers can also be eliminated by gating the clock of the inactive registers. However, elimination of useless power dissipation in modules by controlling multiplexer inputs is necessary even in the case when modules are highly sequential and use power enable/disable signals to turn off the modules which are not targeted. This is due to the fact that useless power dissipation is eliminated in the combinational logic up to the first sequential boundary in the module only where power enable/disable signals take effect. Therefore, although the techniques in this paper are geared towards design styles which do not employ gated clocks and/or power enable/disable signals they can successfully be combined with clock gating power reduction methodologies leading to further savings in power dissipation.

4. Power Conscious Test Synthesis and Scheduling (PC-TSS)

Having described in section 3 the effect of test synthesis and test scheduling on useless power dissipation, now power conscious test synthesis and scheduling (PC-TSS) is considered. The proposed PC-TSS has been integrated into an efficient tabu search-based testable design space exploration which combines the accuracy of incremental test scheduling algorithms with the exploration speed of test scheduling algorithms based on fixed test resource allocation [27]. Elimination of useless power dissipation introduced in section 2 is carried out in two steps. In order to provide a meaningful understanding of the two steps an overview of the tabu search-based testable design space exploration is summarised in section 4.1. Section 4.2 introduces the first step based on power conscious test synthesis moves during testable design space exploration, while section 4.3 gives the second step by describing module selection during power conscious test scheduling.

4.1 Tabu search-based testable design space exploration

This section summarises tabu search-based testable design space exploration [27, 35] in order to provide a general framework for understanding how useless power dissipation is eliminated in BIST RTL data paths.

Tabu search [36] was proposed as a general combinatorial optimization technique. Tabu search falls under the larger category of move-based heuristics which iteratively construct new candidate solutions based on the neighborhood that is defined over the set of feasible solutions and the history of optimization. The neighborhood is implicitly defined by a move that specifies how one solution is transformed into another solution in a single step. The philosophy of tabu search is to derive and exploit a collection of principles of intelligent problem solving. Tabu search controls uphill moves and stimulates convergence toward global optima by maintaining a tabu list of its r most recent moves, where r is called tabu tenure and it is a prescribed constant. Occasionally, it is useful to override the tabu status of a move when the move is aspirated (i.e., improves the search and does not produce cycling near a local minima). Tabu search based heuristics are simple to describe and implement. Furthermore, a well defined cost function and the use of topological information of the design space will lead to an intelligent search of high quality solutions in very low computational time. A solution in the testable design space is a testable data path **T-DP** where test pattern generators and signature analysis register are allocated for each data path module. The proposed tabu search-based testable design space exploration is summarized in Figure 5. The algorithm starts with an initial solution which is a testable data path T-DP_{init} obtained by randomly assigning a single test pattern generator to each input port of every module from the data path as shown from lines 1 to 4. During the optimization process (lines 5 to 22) for each current solution $\text{T-DP}_{current}$ neighbor solutions are generated by moves in the design space (line 7). The neighborhood of the current solution in the testable design space $\text{T-DP}_{current}$ is defined with n_{reg} feasible neighbor solutions, where n_{reg} is the number of data path registers. For each data path register there is a single neighbor solution. Each of the n_{reg} solutions is provided by an independent subroutine designed to identify better configuration of test registers based on two metrics which measure the potential of each solution to reduce test application time and BIST area overhead. A detailed description of moves in the design space and speedup techniques for fast testable design space exploration are given in [27, 35]. If the new testable design does not generate useless power dissipation (algorithm *ACCEPT-MOVE* from section 4.2), test application time T_x and BIST area overhead A_x are computed after a test schedule S_x is generated using the algorithm from section 4.3, as shown from lines 8 to 12. The optimization process is guided towards the objective of minimal test application time design by a cost function which is defined as follows. The cost function is a 2-tuple $C_x = (T_x, A_x)$, where T_x is the test application time, A_x is the BIST area overhead and the following relations are defined:

1. $C_{x_1} = C_{x_2}$ if $(T_{x_1} = T_{x_2})$ and $(A_{x_1} = A_{x_2})$
2. $C_{x_1} < C_{x_2}$ if $(T_{x_1} < T_{x_2})$ or $(T_{x_1} = T_{x_2}$ and $A_{x_1} < A_{x_2})$
3. $C_{x_1} > C_{x_2}$ if $(T_{x_1} > T_{x_2})$ or $(T_{x_1} = T_{x_2}$ and $A_{x_1} > A_{x_2})$

The main objective of the cost function is test application time with BIST area overhead used as tie-breaking mechanism among many possible solutions with same test application time. It should be noted that the minimization of other parameters such as performance degradation, volume of output data, overall test application time and fault escape probability, is a by-product of the proposed optimization using the previously defined cost function. Based on the value of the cost function and on the tabu status of a move, a new solution is accepted or rejected as described from lines 15 to 20 in Figure 5. The tabu list contains registers involved in a move. A move is classified as tabu if a register involved in the move is present in the tabu list. The tabu tenure (length of the tabu list) varies from 5 (small designs) to 10 (complex designs). A move is aspirated as shown in line 15 if it has produced a solution which is better than the best solution reached so far. The testable design space exploration continues until the number of iterations since the previous best solution exceeds a predefined N_{iter}.

It should be noted that in order to explore the testable design space under power constraints, generic power models need to be considered. Estimating power dissipation at a lower level of abstraction for each solution will hinder the

```
ALGORITHM: Testable Design Space Exploration
INPUT: Data Path DP
OUTPUT:Testable Data Path T-DP_best

1  for every M_a from DP with a = 1,...,n_mod do {
2      choose randomly R_l from LIRS_a
       choose randomly R_r from RIRS_a
       choose randomly R_s from OMS_a
       this results into T-DP_init
3  }
4  T-DP_current ←T-DP_init
5  repeat
6    for every R_x from T-DP_current with x = 1,...,n_reg do {
7        generate a new move ([27, 35])
8        if ACCEPT-MOVE (section 4.2) is TRUE{
9          generate test schedule S_x
           using SELECT-MODULE (section 4.3)
10         compute test application time T_x
           using test schedule S_x
11         compute BIST area overhead A_x using T-DP_x
12       }
13   }
14   for each T-DP_x ordered using T_x and A_x do {
15       if not tabu(T-DP_x) or aspirated(T-DP_x) then {
16             T-DP_current ←T-DP_x
17             if best solution so far then
18                   T-DP_best ←T-DP_x
19             break
20       }
21   }
22 until iterations since previous best solution > N_iter
23 return T-DP_best
```

Figure 5. Tabu search-based testable design space exploration

efficient exploration of the testable design space. This is due to the fact that module selection during test scheduling (algorithm *SELECT-MODULE* from section 4.3) requires power computation during *each* stage of the test scheduling algorithm for *each* solution in the design space. Thus by having high level power models will lead to quick computation of power dissipation and fast examination of different alternatives in the solution space. Therefore, the single extra constraint on the design library is that besides performance, cost and pseudorandom test length, an average and/or peak power value should be provided.

4.2 Move Acceptance During Power Conscious Test Synthesis

Previous section 4.1 introduced tabu search-based testable design space exploration. In order to minimize power dissipation, move acceptance criteria (line 8 of Fig. 5) must be modified to examine if the newly generated testable design leads to useless power dissipation (see example 2 of section 3). If a move generates a testable design with useless power dissipation then it is rejected. Fig. 6 shows the new *ACCEPT-MOVE* algorithm. Given the testable data path **T-DP** and the test registers of the current solution (i.e. left and right test pattern generators, $\mathbf{TPG_L}$ and $\mathbf{TPG_R}$, and signature analyzers **SA**), the proposed algorithm accepts or rejects the new testable designs by analyzing the interconnect between test registers and modules. For every module from the output module set of test registers of the current solution, the left and right input register sets (*LIRS* and *RIRS*) are examined. If all the registers from either input register set are test registers then the move is rejected. This is because there will be no value of control signals for multiplexers at the input of data path modules that will eliminate the propagation of spurious transitions. By rejecting the testable data paths using the proposed *ACCEPT-MOVE* algorithm useless power will be eliminated both during test application and while shifting out test responses. It should be noted that if all the moves lead to useless power dissipation then the move which leads to lowest test application time is accepted and the useless power dissipation is minimized using power conscious test scheduling described in the following section.

4.3 Module Selection During Power Conscious Test Scheduling

Testable design space exploration aims to minimize test application time under power constraints with BIST area overhead used as tie-breaking mechanism among many possible solutions with same test application time. In order to satisfy the power constraints during test application, the test application time is computed by carrying out the following two modifications to the test scheduling algorithm based on partitioned testing with run to completion [37].

a. A new module selection algorithm *SELECT-MODULE* is proposed such that useless power is eliminated;

b. The power dissipated by scheduling the selected module M_i is computed and if the power constraint is not satisfied during the current test time then test t_i for M_i is removed from the candidate node set [37] being postponed for a later test time;

Fig. 7 shows the proposed algorithm for module selection during power conscious test scheduling. The module selection aims to eliminate useless power dissipation not only

```
ALGORITHM: ACCEPT-MOVE
INPUT:Testable Data Path T-DP
        Potential test registers {TPG_L, TPG_R, SA}
OUTPUT:boolean variable ACCEPT

1  ACCEPT = TRUE;
2  for every TR_i from {TPG_L, TPG_R, SA} {
3      OMS_i = output module set of TR_i
4      for every M_j from OMS_i {
5          LIRS_j = left input register set of M_j
6          RIRS_j = right input register set of M_j
7          if (every R_k from LIRS_j is test register)
           or (every R_l from RIRS_j is test register)
8              ACCEPT = FALSE;
9      }
10 }
11 return ACCEPT;
```

Figure 6. Proposed algorithm for power conscious test synthesis moves during testable design space exploration (step 1).

```
ALGORITHM: SELECT-MODULE
INPUT:Testable Data Path DP
        Tested Modules {TM_0, ..., TM_n}
        Candidate Modules {CM_0, ..., CM_m}
OUTPUT:module to be scheduled CM_s

1  Active Module Set AMS = {TM_0, ..., TM_n};
2  Active Register Set ARS=TestRegisters(AMS);
3  for every CM_i from {CM_0, ..., CM_m} {
4      UR ← ∅; UM ← ∅;
5      repeat
6          ARS ← ARS ∪ UR;
7          UM ← GetUselessMod(ARS);
8          AMS ← AMS ∪ UM;
9          UR ← GetUselessReg(AMS ∪ UM);
10     until UR ≠ ∅;
11     compute P_i using AMS∪ARS;
12 }
13 select the CM_s with least power dissipation P_s;
14 return CM_s;
```

Figure 7. Proposed algorithm for module selection during power conscious test scheduling (step 2).

in *useless registers* (UR) at the output of currently tested modules, but it considers also the useless power dissipation in *useless modules* (UM) to which spurious transitions are propagated through *useless registers* (see example 3 of section 3). Given the testable data path, the modules scheduled at the current test time (tested modules) and the candidate modules to be scheduled according to the resource conflict graph [27, 37], the algorithm *SELECT-MODULE* selects the candidate module which when scheduled at the current test time will lead to the minimum increase in power dissipation. Initially the active module set (AMS) contains the tested modules, while the active register set (ARS) contains the test registers which generate test patterns and analyze test responses for currently tested modules. For each candidate module the power dissipation is computed by recursively propagating spurious transitions through UR and UM (lines 4 to 11 in Fig. 7). Initially both sets of useless registers and useless modules are null. UM is computed using ARS and UR. A module is assigned to UM if all the registers in its left *or* right input register set are active at the current test time. The useless modules are considered for detecting the propagation of spurious transitions to useless registers. Once AMS is updated with UM, useless registers UR are computed using the updated AMS. A register is assigned to UR if all the modules in its input module set are active at the current test time. All the useless registers detected in the current iteration are used to update the set of active registers ARS in the next iteration. Once ARS is updated, new useless modules are detected and this recursive propagation of spurious transitions continues until no new useless registers are detected. At the end of the recursive propagation of spurious transitions (lines 5 to 10 in Fig. 7) AMS and ARS contain *not only* the tested modules and their test registers, but also *all* the active data path elements during the current test time. AMS and ARS are used to compute both necessary and useless power dissipation associated with selecting candidate module CM_i to be scheduled at the current test time. Finally, the candidate module CM_s which leads to minimum power dissipation is selected to be scheduled at the current test time.

It is interesting to note that both algorithms *ACCEPT-MOVE* and *SELECT-MODULE* introduced in this section guarantee that as far as there is a potential solution leading to lower test application time and BIST area overhead, and which eliminates useless power dissipation then it will be selected. This is unlike traditional testable design space exploration algorithms [26, 27] where the single aim is to reduce test application time and BIST area overhead without any consideration of power dissipation.

Table 1. Experimental results obtained using the proposed power conscious test synthesis and scheduling (PC-TSS) for BIST RTL data paths

Design	Synthesis type	Power Constraint $12\times P_u$				CPU time (s)	Power Constraint $15\times P_u$				CPU time (s)	Power Constraint $19\times P_u$				CPU time (s)
		TAP (P_u)	SP	TAT (T_u)	BAO (N_G)		TAP (P_u)	SP	TAT (T_u)	BAO (N_G)		TAP (P_u)	SP	TAT (T_u)	BAO (N_G)	
8DCT-10	TA-TSS	13.8	9	10	104	10	16.88	6	9	113	9	17.87	9	8	104	12
	PC-TSS	11.8	8	10	122		12.5	6	10	113		16.5	9	8	106	
	red (%)	**14**	**11**	**0**	**-17**		**25**	**0**	**-11**	**0**		**7**	**0**	**0**	**-1**	
8DCT-14	TA-TSS	12.12	5	8	83	6	12.97	4	8	78	4	21.8	6	5	87	4
	PC-TSS	11	5	8	91		11.12	4	8	82		18	5	5	91	
	red (%)	**9**	**0**	**0**	**-9**		**14**	**0**	**0**	**-5**		**17**	**16**	**0**	**-4**	
EWF-17	TA-TSS	14	8	5	91	2	14	10	5	91	1	15.4	8	5	83	2
	PC-TSS	10.33	8	5	67		10	9	5	102		14.6	8	5	92	
	red (%)	**26**	**0**	**0**	**26**		**28**	**10**	**0**	**-12**		**5**	**0**	**0**	**-10**	
32DCT-33	TA-TSS	12.45	17	20	338	344	13.15	16	19	321	444	17.85	16	14	321	418
	PC-TSS	11.27	12	22	251		12.35	12	20	286		17.17	14	14	321	
	red (%)	**5**	**29**	**-10**	**25**		**6**	**25**	**-5**	**10**		**3**	**12**	**0**	**0**	
32DCT-38	TA-TSS	12.68	14	22	298	322	13.42	15	21	289	495	17.4	13	15	298	453
	PC-TSS	11.3	11	23	251		12.47	13	21	303		16.5	13	16	303	
	red (%)	**10**	**21**	**-4**	**15**		**7**	**13**	**0**	**-1**		**5**	**0**	**-8**	**-1**	

5. Experimental Results

Power conscious test synthesis and test scheduling has been implemented on SUN SPARC 20 workstation within a BIST synthesis environment [27] targeting low test application time. To give insight into the savings achieved using the presented approach, Table 1 shows a comparison of test application power (TAP) and shifting power (SP) when using the proposed Power Conscious Test Synthesis and Scheduling (PC-TSS) and Time and Area Test Synthesis and Scheduling (TA-TSS). PC-TSS employs the newly proposed algorithms in section 4. TA-TSS [27] explores the testable design in order to minimize test application time and BIST area overhead under power constraints without considering the effect of useless power dissipation. The comparison is carried out for a number of benchmark examples including elliptic wave digital filter (EWF) and 8 and 32 point discrete cosine transform (DCT). The benchmarks were synthesized using the ARGEN high-level synthesis system [38] for different execution time constraints ranging from 10 to 38. The test application times (TAT) of adders and multipliers are assumed to be $T_+ = T_u$, and respectively $T_* = 4\times T_u$, where T_u is a reasonably large integer and can be estimated for the required fault coverage using the techniques from [39]. The power dissipation for registers, adders and multipliers are assumed to be $P_R = P_u$, $P_+ = P_u$ and $P_* = 4\times P_u$ where P_u can be derived using the techniques from [40] or it can be computed for each module using given pseudorandom sequences that achieve the required fault coverage. To assess the effectiveness of the proposed techniques, PC-TSS and TA-TSS have been compared for different power constraints ranging from $12\times P_u$ to $19\times P_u$. The generic high level model of power dissipation provides the flexibility of applying the proposed algorithms to various library modules with different power characterization. Finally, BIST area overhead (BAO) is evaluated in terms of equivalent number of two input gates (N_G).

For all the evaluated circuits the proposed PC-TSS produces significantly less TAP and less or comparable SP than the TA-TSS. For example in the case of *EWF-17* with power constraint $15\times P_u$ reductions of 28% in TAP and of 5% in SP are achieved. The reduction caused by power conscious test synthesis and scheduling is also dependent on the structural interconnect of the data path elements which is exploited by the algorithms presented in section 4. This means that savings vary from one example to another as in the case of *32DCT-33* with power constraint $12\times P_u$ where savings of 5% in TAP and of 29% in SP are achieved. It should be noted that for some examples TAT and/or BAO are lower in the case of TA-TSS. The savings in power dissipation in the case of PC-TSS are due to the fact that the power constraint which is met by TA-TSS ignores useless power dissipation. This is not the case for PC-TSS where useless power dissipation is accounted for both TAP and SP. Furthermore, for all the evaluated circuits due to the underestimation of power dissipation (ignorance of useless power dissipation) the TA-TSS *does not meet the power constraints* which can decrease the reliability of the circuit and lead to yield loss [3, 4]. The proposed PC-TSS algorithm allows the large testable design space to be explored power-consciously and efficiently leading to high quality solutions in low computational time. For example it takes less than 500s to efficiently explore the testable design space of complex circuits as 32 point DCTs.

6. Conclusions

This paper has shown for the first time how power dissipation is minimized in BIST RTL data paths by using power conscious test synthesis and test scheduling. The power dissipation was classified into necessary and useless power during test application. The effect of test synthesis and scheduling on power dissipation was analyzed and power minimization has been achieved in two steps. Firstly, during the testable design space exploration only power conscious test synthesis moves are accepted leading to minimization of useless power during test application and while shifting out test responses. Secondly, the module selection during power conscious test scheduling leads to minimum increase in power dissipation while reducing both test application and test area overhead. Experimental results using generic power models have shown savings in power dissipation up to 28% during test application and up to 29% while shifting out test responses. The proposed power conscious test synthesis and scheduling is of prime importance for satisfying power constraints in BIST RTL data paths leading to higher yield and reliability.

Future work will investigate the impact of power conscious test synthesis and scheduling algorithms on the performance, cost and power dissipation of test controller. Further, a full comparison of actual power dissipation of two circuits using built-in self-test that are synthesized using traditional and power conscious algorithms will be undertaken.

References

[1] M. Pedram, "Power minimization in IC design: Principles and applications," *ACM Transactions on Design Automation of Electronic Systems*, vol. 1, pp. 3–56, Jan 1996.

[2] P. Girard, "Low power testing of VLSI circuits: Problems and solutions," in *First International Symposium on Quality of Electronic Design (ISQED)*, pp. 173–180, 2000.

[3] Y. Zorian, "A distributed BIST control scheme for complex VLSI devices," in *Proc. 11th IEEE VLSI Test Symposium*, pp. 4–9, 1993.

[4] S. Wang and S. Gupta, "ATPG for heat dissipation minimization during test application," *IEEE Transactions on Computers*, vol. 47, pp. 256–262, Feb 1998.

[5] P. Bardell, W. McAnney, and J. Savir, *Built-In Self Test - Pseudorandom Techniques*. John Wiley & Sons, 1986.

[6] S. Wang and S. Gupta, "ATPG for heat dissipation minimization during scan testing," in *Proc. 34th Design Automation Conference*, pp. 614–619, 1997.

[7] X. Zhang, K. Roy, and S. Bhawmik, "POWERTEST: A tool for energy conscious weighted random pattern testing," in *12th International Conference on VLSI Design*, pp. 416–422, 1999.

[8] X. Zhang and K. Roy, "Design and synthesis of low power weighted random pattern generator considering peak power reduction," in *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 148–156, 1999.

[9] M. Brazzarola and F. Fummi, "Power characterization of LFSRs," in *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 138–146, 1999.

[10] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector inhibiting technique for low energy BIST design," in *Proc. 17th IEEE VLSI Test Symposium*, pp. 407–412, 1999.

[11] S. Manich, A. Gabarro, M. Lopez, J. Figueras, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, P. Teixeira, and M. Santos, "Low power BIST by filtering non-detecting vectors," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 16, pp. 193–202, June 2000.

[12] F. Corno, M. Rebaudengo, M. S. Reorda, and M. Violante, "A new BIST architecture for low power circuits," in *IEEE European Test Workshop (ETW99)*, pp. 160–164, 1999.

[13] F. Corno, M. Rebaudengo, M. S. Reorda, G. Squillero, and M. Violante, "Low power BIST via hybrid cellular automata," in *18th IEEE VLSI Test Symposium*, 2000.

[14] C. Ravikumar and N. Prasad, "Evaluating BIST architectures for low power," in *7th Asian Test Symposium*, pp. 430–434, 1998.

[15] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "Circuit partitioning for low power BIST design with minimized peak power consumption," in *8th Asian Test Symposium (ATS99)*, pp. 89–94, 1999.

[16] D. Gizopoulos, N. Kranitis, A. Paschalis, M. Psarakis, and Y. Zorian., "Effective low power bist for datapaths," in *Proc. of the Design, Automation and Test in Europe Conference (DATE)*, p. 757, 2000.

[17] F. Corno, M. Rebaudengo, M. S. Reorda, and M. Violante, "Optimal vector selection for low power BIST," in *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 219–226, 1999.

[18] S. Gerstendorfer and H. Wunderlich, "Minimized power consumption for scan-based BIST," *Journal of*

Electronic Testing: Theory and Applications (JETTA), vol. 16, pp. 203–212, June 2000.

[19] S. Wang and S. Gupta, "LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation," in *Proc. IEEE International Test Conference*, pp. 85–94, 1999.

[20] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static compaction techniques to control scan vector power dissipation," in *Proc. of the 18th IEEE VLSI Test Symposium*, 2000.

[21] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test application," *IEEE Transactions on CAD*, vol. 17, pp. 1325–1333, Dec 1998.

[22] S. Wang and S. Gupta, "DS-LFSR: A new BIST TPG for low heat dissipation," in *Proc. IEEE International Test Conference*, pp. 848–857, 1997.

[23] N. Nicolici, B. Al-Hashimi, and A. Williams, "Minimisation of power dissipation during test application in full scan sequential circuits using primary input freezing," *IEE Proceedings - Computers and Digital Techniques*, vol. 147, July 2000.

[24] N. Nicolici and B. Al-Hashimi, "Scan latch partitioning into multiple scan chains for power minimization in full scan sequential circuits," in *Proc. IEEE/ACM Design Automation and Test in Europe (DATE 2000)*, (Paris, France), pp. 715–722, March 2000.

[25] T.-C. Huang and K.-J. Lee, "An input control technique for power reduction in scan circuits during test application," in *Proc. 8th Asian Test Symposium*, pp. 315–320, 1999.

[26] S. Lin, C. Njinda, and M. Breuer, "Generating a family of testable designs using the BILBO methodology," *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol. 4, no. 2, pp. 71–89, 1994.

[27] N. Nicolici and B. Al-Hashimi, "Efficient BIST hardware insertion with low test application time for synthesized data paths," in *Proc. IEEE/ACM Design Automation and Test in Europe (DATE 99)*, (Munich, Germany), pp. 289–295, March 1999.

[28] G. Lakshminarayana, A. Raghunathan, N. Jha, and S. Dey, "Power management in high level synthesis," *IEEE Transactions on VLSI*, vol. 7, pp. 7–15, Mar 1999.

[29] R. Chou, K. Saluja, and V. Agrawal, "Scheduling tests for VLSI systems under power constraints," *IEEE Transactions on VLSI*, vol. 5, pp. 175–184, Jun 1997.

[30] V. Muresan, V. Muresan, X. Wang, and M. Vladutiu, "The left edge algorithm and the tree growing technique in block-test scheduling under power constraints," in *Proc. of the 18th IEEE VLSI Test Symposium*, 2000.

[31] E. Larsson and Z. Peng, "An estimation-based technique for test scheduling," in *Electronic Circuits and Systems Conference*, pp. 25–28, 1999.

[32] C. Ravikumar, A. Verma, and G. Chandra, "A polynomial-time algorithm for power constrained testing of core based systems," in *8th Asian Test Symposium*, pp. 107–112, 1999.

[33] K. Chakrabarty, "Design of system-on-a-chip test access architectures under place-and-route and power constraints," in *Proc. IEEE/ACM Design Automation Conference (DAC)*, pp. 432–437, 2000.

[34] A. Raghunathan, N. Jha, and S. Dey, "Register transfer level power optimization with emphasis on glitch analysis and reduction," *IEEE Transactions on CAD*, vol. 18, pp. 1114–1131, Aug 1999.

[35] N. Nicolici, "Power minimisation techniques for testing low power VLSI circuits," Tech. Rep. Upgrade from MPhil to PhD, University of Southampton, UK, December 1999.

[36] F. Glover and M. Laguna, "Tabu search," in *Modern Heuristic Techniques for Combinatorial Problems* (C. Reeves, ed.), pp. 70–150, McGraw-Hill Book Company, 1995.

[37] G. Craig, C. Kime, and K. Saluja, "Test scheduling and control for VLSI built-in self-test," *IEEE Transactions on Computers*, vol. 37, pp. 1099–1109, Sep 1988.

[38] P. Kollig and B. Al-Hashimi, "A new approach to simultaneous scheduling, allocation and binding in high level synthesis," *IEE Electronics Letters*, vol. 33, pp. 1516–1518, Aug 1997.

[39] A. Majumdar and S. Vrudhula, "Fault coverage and test length estimation for random pattern testing," *IEEE Transactions on Computers*, vol. 44, pp. 234–247, Feb 1995.

[40] E. Macii, M. Pedram, and F. Somenzi, "High level power modeling, estimation, and optimization," *IEEE Transactions on CAD*, vol. 17, pp. 1061–1079, Nov 1998.